

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA  
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



CHADLI BENDJEDID UNIVERSITY  
FACULTY OF SCIENCES & TECHNOLOGIES  
COMPUTER SCIENCE DEPARTMENT

# THESIS

---

## Towards a Deep Learning-Based E-learning Platform

---

Submitted in partial fulfillment of the requirements for a  
Master's degree in Computer Science  
Option : Intelligent Computer Systems

Presented By :  
Mostepha Amine BDEIRA

On June 23, 2024, In front of the Jury composed of :

|            |                |     |       |
|------------|----------------|-----|-------|
| President  | Mr. BOUTOUIL A | MCB | UCBET |
| Examiner   | Mr. BENTRAD S  | MCA | UCBET |
| Supervisor | Mr. CHEMAM C   | MCB | UCBET |

2023/2024

---

## ACKNOWLEDGEMENT

*In the name of Allah, the Most Gracious, the Most Merciful,*

*I would like to take this opportunity to express a deep sense of gratitude and appreciation to Allah, the source of all knowledge and wisdom, who gave me the perseverance and strength to complete this work.*

*I am highly indebted to all the teachers of the Department of Computer Science, who have been truly helpful and supportive throughout our academic journey. Their love for teaching and passion for excellence motivated and inspired us a lot.*

*I am indeed highly grateful to **Mr. CHEMMAM**, whose continuous support and valuable guidelines from time to time enriched this study project as my supervisor. His expertise, patience, and encouragement have gone a long way in shaping this work.*

*I would like to express my gratitude to all those who have directly or indirectly helped me with this project. Your time, feedback, and insight are much appreciated. Notably, I want to thank **Salah, Abdelhadi, Djaber, Anis**.*

*I am deeply indebted to my parents and family, who have been a constant source of support, love, and encouragement all the way. It is their faith in me that has given me so much strength.*

*I now turn my mind to express my sincere appreciation to all colleagues and friends who have assisted or supported me in this process. Your encouragement and friendship in traversing this pathway of advanced education have made this journey really so memorable.*

*Thank you one and all for being part of this important milestone in my academic and personal growth.*

---

## DEDICATION

*This work is dedicated to :*

- *My father, whose unwavering support, wisdom, and love have been my guiding light. Your belief in me has given me the strength to pursue my dreams.*
- *My mother, for her endless patience, love, and sacrifices. Your unwavering faith in me has been my greatest motivation.*
- *To my sisters , though kilometers may separate us, your love and encouragement have always been close to my heart. Your belief in me has been a source of strength.*
- *To the I-tech club, for providing a platform where knowledge is shared, and ideas flourish. Your commitment to innovation and learning has inspired me to strive for excellence.*

---

# TABLE OF CONTENTS

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>Table of Contents</b>                                            | <b>2</b>  |
| <b>List of Figures</b>                                              | <b>4</b>  |
| <b>List Of Tables</b>                                               | <b>5</b>  |
| <b>Introduction</b>                                                 | <b>1</b>  |
| <b>1. Background &amp; Motivation</b>                               | <b>1</b>  |
| 1.1 Background :-                                                   | 1         |
| 1.2 Motivation :-                                                   | 1         |
| 2. Objectives                                                       | 2         |
| 3. Research Questions                                               | 3         |
| 4. Structure Of The Thesis                                          | 4         |
| <b>Chapter 01 : State of The Art</b>                                | <b>6</b>  |
| 1.1 E-Learning Platforms : An Overview                              | 6         |
| 1.1.1 Types of E-Learning Platforms:                                | 6         |
| 1.1.2 Advantages of E-Learning Platforms:                           | 8         |
| 1.1.3 Challenges and limitations of E-Learning Platforms:           | 9         |
| 1.1.4 Current Trends and Future Directions of E-learning Platforms: | 10        |
| 1.2 Deep Learning in Education                                      | 11        |
| 1.2.1 Personalized Learning                                         | 12        |
| 1.2.2 Predictive Analytics                                          | 12        |
| 1.2.3 Content Creation & Enhancement                                | 13        |
| 1.2.4 Automated Administrative Tasks                                | 13        |
| 1.2.5 Challenges and Considerations                                 | 13        |
| 1.2.6 Case Studies and Real-World Applications                      | 14        |
| 1.2.7 Future Directions                                             | 15        |
| 1.3 Video Annotation & Linked Data                                  | 15        |
| 1.3.1 Video Annotation                                              | 15        |
| 1.3.2 Linked Data                                                   | 21        |
| 1.4 Existing Solutions & Gaps                                       | 28        |
| 1.4.1 Existing Solutions                                            | 28        |
| 1.4.2 Gaps in Existing Solutions                                    | 29        |
| <b>Chapter 02 : Conceptual Study</b>                                | <b>30</b> |
| 2.1 Introduction                                                    | 30        |
| 2.2 UML Diagrams                                                    | 30        |
| 2.2.1 Use Case Diagrams                                             | 31        |
| 2.2.2 Class Diagram                                                 | 35        |
| 2.2.3 Sequence Diagrams                                             | 39        |
| 2.3 System Main Functionalities                                     | 45        |
| 2.3.1 Personalized Course Recommendations                           | 45        |
| 2.3.2 AI-Driven Assignment Generation                               | 62        |
| <b>Chapter 03 : Implementation and Results</b>                      | <b>66</b> |

---

|                                           |           |
|-------------------------------------------|-----------|
| 3.1 Implementation-----                   | 66        |
| 3.1.1 Development Environment-----        | 66        |
| 3.1.2 Overview of Codebase Structure----- | 68        |
| 3.1.3 Keycode Snippets-----               | 74        |
| 3.2 Results and Analysis-----             | 78        |
| 3.2.1 Deep Learning Model :-----          | 78        |
| 3.3 Platform Representation-----          | 83        |
| <b>Conclusion and Future Work-----</b>    | <b>97</b> |
| 1. Summary-----                           | 97        |
| 2. Key Findings and Implications-----     | 97        |
| 3. Limitations and Challenges-----        | 98        |
| 4. Recommendations for Future Work-----   | 98        |
| 5. Conclusion-----                        | 99        |
| <b>Bibliography-----</b>                  | <b>99</b> |

---

## LIST OF FIGURES

**[Fig 01 :](#)** Explains how does the Video Annotation Framework work

**[Fig 02.](#)** A Simple Explanation of how does the Manual Video Annotation Works

**[Fig 03.](#)** System Architecture of semi-automatic video annotation tool

**[Fig 04.](#)** Visualization-Based Active Learning Pipeline

**[Fig 05.](#)** The workflow of bounding box annotation for object tracking method

**[Fig 06.](#)** Course Recommendation UseCase Diagram

**[Fig 07.](#)** Assignment Generation Process Use Case Diagram

**[Fig 08.](#)** Video Transcription Storing Process Use Case Diagram

**[Fig 09.](#)** Video Annotation Use Case Diagram

**[Fig 10.](#)** Course Content Management Use Case Diagram

**[Fig 11.](#)** Class Diagram of the System

**[Fig 12.](#)** Recommendation Process Sequence Diagram

**[Fig 13.](#)** Authentication Process Sequence Diagram

**[Fig 14.](#)** Assignment Generation Sequence Diagram

**[Fig 15.](#)** Transcription Storing Sequence Diagram

**[Fig 16.](#)** Techniques used by Collaborative Filtering

**[Fig 17.](#)** Personalized Course Recommendation System Design Diagram

**[Fig 18.](#)** Windows Logo

**[Fig 19.](#)** Vscode Logo

**[Fig 19.](#)** Xampp Logo

**[Fig 20.](#)** npm Logo

**[Fig 21.](#)** Spark Logo

**[Fig 22.](#)** CodeIgniter Logo

**[Fig 23.](#)** a Diagram explain the MVC Architecture

**[Fig 24.](#)** Python Logo

**[Fig 25.](#)** PHP Logo

**[Fig 26.](#)** HTML5 Logo

**[Fig 27.](#)** CSS3 Logo

**[Fig 28.](#)** Javascript Logo

**[Fig 29.](#)** TailwindCss Logo

**[Fig 30.](#)** MySQL Logo

**[Fig 31.](#)** Anthropic Claude Logo

**[Fig 32.](#)** VideoJs Logo

---

**[Fig 33. Flask Logo](#)**

**[Fig 34. Visualization of Loss history of model 1](#)**

**[Fig 35. Evaluation Results of Model 1](#)**

**[Fig 36. Visualization of Loss history of model 2](#)**

**[Fig 37. Evaluation Results of Model 2](#)**

**[Fig 38. Courses dataframe representation](#)**

**[Fig 39. Courses word cloud](#)**

**[Fig 40. Total number of ratings probability](#)**

**[Fig 41. Mean Ratings Probability](#)**

**[Fig 42. Mean Ratings vs Total number of ratings](#)**

**[Fig 43. How does the embedding layers work](#)**

**[Fig 44. Visualizing model 1 architecture](#)**

**[Fig 45. Model 2 architecture](#)**

**[Fig 46. HomePage Desktop View](#)**

**[Fig 47. HomePage Mobile View](#)**

**[Fig 48. About Page](#)**

**[Fig 49. Events Page](#)**

**[Fig 50. Blog Page](#)**

**[Fig 51. Blog Post Page](#)**

**[Fig 52. Login Page](#)**

**[Fig 53. RegisterPage](#)**

**[Fig 54. Dashboard Interface 1](#)**

**[Fig 55. Dashboard Interface 2](#)**

**[Fig 56. Users List Interface](#)**

**[Fig 57. Profile view Interface](#)**

**[Fig 58. Profile edit Interface](#)**

**[Fig 59. Messaging Interface](#)**

**[Fig 60. Instructors List](#)**

**[Fig 61. Blog Post Creation List](#)**

**[Fig 62. Assignment generation interface](#)**

**[Fig 63. Courses Catalog](#)**

**[Fig 64. Course Creation interface](#)**

**[Fig 65. Video database index + fetching transcriptions](#)**

---

**[Fig 66.Adding videos metadata](#)**

**[Fig 67.New lesson creation for instructors](#)**

**[Fig 68.Course Preview](#)**

**[Fig 69. .Lesson Preview](#)**

**[Fig 70. .Search Engine Results](#)**

**[Fig 71. .Adding Annotations](#)**

---

## LIST OF TABLES

[1. Stats of a study : Student Engagement in Personalized Learning](#)

[2. Stats of a study : Effectiveness of Predictive Analytics](#)

[3. Merged Courses with Ratings](#)

[4. Mean Ratings of courses](#)

[5. Model 1 Config](#)

[6. Model 2 Config](#)

---

## ABSTRACT

This dissertation explores the platform's concept, architecture, and implementation, emphasizing features such as individualized learning paths, easy content creation tools for instructors, and enhanced video learning experiences for students. The implementation phase includes details on the development environment, code structure, and the use of artificial intelligence.

The platform is designed to make learning more personal and to streamline administrative tasks, which should help students stay engaged and instructors be more efficient. The dissertation also reflects on what these findings mean, what was learned during the development, and how things could be improved in the future. These insights are important for understanding how AI can continue to shape education.

Looking ahead, AI in education could make for an increasingly tailored and interactive learning process that may through time revolutionize the approach toward online learning.

**Keywords :** AI-enhanced e-learning, deep learning, NLP, personalized course recommendations, Surprise model, automated assignment generation, Claude model, video transcription, interactive video annotations, linked data, usability testing, performance testing, security testing, functional testing, deployment, student engagement, instructor efficiency.

---

## الملخص

تستكشف هذه الأطروحة مفهوم المنصة وبنيتها وتنفيذها، مع التركيز على ميزات مثل مسارات التعلم الفردية وأدوات إنشاء المحتوى السهلة للمدرسين وتجارب التعلم بالفيديو المحسنة للطلاب. تتضمن مرحلة التنفيذ تفاصيل حول بيئة التطوير وبنية الكود واستخدام الذكاء الاصطناعي.

تم تصميم النظام الأساسي لجعل التعلم أكثر خصوصية و تبسيط المهام الإدارية، الأمر الذي من شأنه أن يساعد الطلاب على الاستمرار في المشاركة ويجعل المعلمين أكثر كفاءة. وتعكس الأطروحة أيضًا ما تعنيه هذه النتائج، وما تم تعلمه أثناء التطوير، وكيف يمكن تحسين الأمور في المستقبل. تعتبر هذه الأفكار مهمة لفهم كيف يمكن للذكاء الاصطناعي أن يستمر في تشكيل التعليم.

وبالنظر إلى المستقبل، يمكن للذكاء الاصطناعي في التعليم أن يؤدي إلى عملية تعليمية مصممة وتفاعلية بشكل متزايد والتي قد تحدث ثورة مع مرور الوقت في النهج تجاه التعلم عبر الإنترنت.

**الكلمات المفتاحية :** التعلم الإلكتروني المعزز بالذكاء الاصطناعي، التعلم العميق، البرمجة اللغوية العصبية، توصيات الدورات التدريبية المخصصة، نموذج المفاجأة، إنشاء المهام الآلية، نموذج كلود، نسخ الفيديو، تعليقات الفيديو التفاعلية، البيانات المرتبطة، اختبار قابلية الاستخدام، اختبار الأداء، اختبار الأمان، الاختبار الوظيفي، النشر، مشاركة الطلاب، كفاءة المعلم.

# INTRODUCTION

## 1. BACKGROUND & MOTIVATION

### 1.1 BACKGROUND :

Basic education is no exception as this high-paced tech progress is already helping reshape the whole world. For years a foundation of in-class lessons, physical textbooks, and in-person interaction has facilitated the foray into education. Except there was a bit of a problem: the landscape of education was shaken up a bit with the changes in the internet and digital technologies. E-learning platforms have emerged as powerful ways of giving flexible, accessible, and personalized learning to students across the globe.

E-learning has revolutionized the delivery and consumption of educational content through the platform portfolios of Moodle, Blackboard, and Coursera. The learners on these platforms have a broad portfolio of courses and resources that, in turn, help them get quality education at their convenience. It is enhanced by the integration of multimedia content, discussion forums, and automated assessments, hence making the whole experience more educative to the learners.

However, e-learning platforms usually present several challenges. In most cases, this creates one major challenge: maintaining student engagement with so many distractions in the online environment. The lack of physical presence sometimes contributes to lower motivation. Further, the 'one size fits all' mode of education traditionally is mostly not capable of serving the different needs of the students. This is perhaps where the need for more personalized learning experiences lies.

---

### 1.2 MOTIVATION :

To build an AI-powered e-learning platform effectively, it is crucial to address several key issues that currently hinder the efficiency, effectiveness, and accessibility of online learning. This motivation stems from the need to create a transformative educational experience that leverages the capabilities of artificial intelligence. The primary factors driving this motivation are detailed below:

**Personalized Learning :** Traditional educational methods often do not help in satisfying individual students' needs and their learning pace. Personalized learning aims through deep

---

learning algorithms to customize educational content that complements student capabilities. Such individualization ensures student concentration to the maximum and, consequently, increment of learning.

**Accessibility :** E-learning platforms can level-set masses' education by making it accessible to the global population. It is essential that they be inclusive to take care of students with diverse needs, abilities, and capabilities. AI technologies, for example, automated transcription and video annotation, enable the captioning of videos and linking of content to more extensive resources, thereby making it more accessible.

**Efficiency for Instructors :** They make it possible for the instructors to complete the process of creating assignments, grading, and transcribing lessons in less time. Automated essay scoring with AI-enabled assignment generation will be much less manual work for an educator and certainly more effective in time.

**Enhanced Understanding :** Video annotation, integrated with linked data, helps deliver a much richer and interactive learning experience. With video content related to external resources on which the content is based, students gain an in-depth insight into the topics they are exposed to and thus better understand them. It not only enhances understanding but also leads to independent learning and critical thinking.

**Scalability :** One of the key advantages that the e-learning platforms offer is the possibility to scale and reach out to many learners at a go. AI and deep learning technologies will take the scalability of these platforms to the next level by providing personalized learning processes through automation for such a big audience. That will remain an important attribute in meeting the ever-increasing demand for online education to assure quality education for all.

---

## 2. OBJECTIVES

This dissertation focuses on developing an advanced e-learning platform by infusing deep learning and artificial intelligence to render the learning experience better for both students and faculty. The research will cover the following specific objectives:

- **To Develop a Personalized Course Recommendation System Using Deep Learning.**
- **To Create an AI-Driven Assignment Generation Tool.**
- **To Develop a Video Annotation System Enhanced by Linked Data.**

- 
- **To Evaluate the Impact of the AI-Enhanced E-Learning Platform.**
  - **To Ensure Scalability and User-Friendliness of the Platform.**

### **3. RESEARCH QUESTIONS**

This dissertation follows the mentioned research questions to guide the process of developing and evaluating the AI-eLearning platform:

- 1. How could DeepLearning algorithms be used to effectively provide the recommendations of courses on an e-learning platform?**
  - The study will address which deep learning methods can be integrated to analyze student data and learning patterns. The goal is to identify the best algorithms for course recommendation that will match individual student needs, as a way to improve better student engagement and learning outcomes.
- 2. What approaches can be used to build an AI-based system for automatically generating assignments that align with the objectives of the course and the performance of the students?**
  - This task will be looking into how AI can support an instructor to create an assignment. The goal is to identify the best ways and algorithms for creating relevant and challenging items that are customized by course content and individual student performance.
- 3. What are the best practices for implementing video annotation systems that use linked data to improve better comprehension and engagement with educational content?**
  - The question focuses on the use of video annotation and linked data to help interact and enhance learning. It proposes the identification of effective strategies to link video content with some external resources, making such annotations part of the e-learning platform integration.
- 4. How would AI-based transcription services be applied on an e-learning platform in order to provide accurate and accessible transcriptions of video lectures?**

- 
- The present question addresses the use of AI in transcription tools to make learning more accessible. This will focus on the identification of reliable AI models for the transformation of speech into text and the integration of such models within the platform to support students with different learning needs.

**5. How should the platform be designed to make sure it is scalable, user-friendly, and high-performing when catering to a large volume of users?**

- This question will address matters of technology in developing the platform. It has to determine the best practices in architecture that will guarantee the platform is scalable without causing performance issues and is simple to use for both students and instructors.

**6. What are the consequences for student engagement, learning outcomes, and instructor efficiency as deep learning technologies with AI get integrated into an e-learning environment?**

- The latter would assess how effective an AI-rich e-learning platform really will be. It will test whether personalized recommendations, auto-assignments, video transcriptions, and annotations impact student performance and instructor productivity.

---

## **4. STRUCTURE OF THE THESIS**

The thesis is organized into six main chapters, each addressing specific details of the development and evaluation of the AI-enhanced e-learning platform:

- **Introduction**

This chapter provides a brief overview of the research project, covering the background, motivation, aims, research questions, study scope, and limitations; it tries to put the research into context. This sets the stage for the e-learning platform development and evaluation.

- **Literature Review**

---

Literature review, on the other hand, will identify existing e-learning platforms, how deep learning is applied in education, AI-driven tools in assignment generation and transcription, in video annotation, and linked data, and other identified gaps by the previous researches and how these findings will be addressed by this project.

- **Conceptual Study**

In this chapter, a full technical system development using CodeIgniter 4 and Python, which explains the functions, is elaborated on. It presents the front-end and back-end development, integration of AI components for personalized recommendations, assignment generation transcription, video annotation, and linked data implementation.

- **Implementation and Results**

This chapter outlines the implementation process and, along with it, the analysis of findings from the performance assessment of the e-learning platform. It presents both data types: quantitative (such as the performance metrics) along with a qualitative report. Such analysis of results will thus inform how the platform has influenced student engagement, learning outcomes, and even the instructor's efficiency.

- **Conclusion and Future Work**

This chapter outlines the results of the dissertation, which add contributions and help bring understanding to the domain of e-learning and educational technology. It discusses limitations found during research and suggestions for future work to enhance the platform further and guard against existing challenges.

---

---

# CHAPTER 1

---

## STATE OF THE ART

### 1.1 E-LEARNING PLATFORMS : AN OVERVIEW

What e-learning platforms offer these days is considered a revolution and a qualitative leap made by humanity with regard to education in general, especially after the Covid pandemic that struck the world and paralyzed its movement for several months. These platforms, with the help of technology, provide educational content directed not only to academic students, but even The general public. These platforms may also contribute to facilitating communication and administrative tasks. The development of e-learning has been closely linked to technology. Despite the presence of many effective platforms today, such as Coursera, Udemy, and Moodle, these platforms still face many challenges in terms of participation, real evaluation, and customization.

#### 1.1.1 TYPES OF E-LEARNING PLATFORMS:

E-learning platforms can be broadly categorized into several types based on their primary function and target audience :

- **Learning Management Systems (LMS) :**

A learning management system is a software application or web-based technology used to plan, implement and assess a specific learning process. It's used for e-learning practices and, in its most common form, consists of two elements: a server that performs the base functionality and a user interface (UI) that is operated by instructors, students and administrators.

Typically, an LMS provides an instructor with a way to create and deliver content, monitor student participation, and assess student performance. It might also provide students with interactive features, such as threaded discussions, video conferencing and discussion forums.

---

Businesses, government agencies, and traditional and online educational institutions often use these systems. They can improve traditional educational methods, while also saving organizations time and money. An effective system lets instructors and administrators efficiently manage elements such as user registration and access, content, calendars, communication, quizzes, certifications and notifications.

The Advanced Distributed Learning group, sponsored by the U.S. Department of Defense, has created a set of specifications called the Sharable Content Object Reference Model (SCORM) to encourage the standardization of LMSes. [\[1\]](#)

- **Massive Open Online Courses (MOOCs) :**

A massive open online course (MOOC /mu:k/) or an open online course is an online course aimed at unlimited participation and open access via the Web. In addition to traditional course materials, such as filmed lectures, readings, and problem sets, many MOOCs provide interactive courses with user forums or social media discussions to support community interactions among students, professors, and teaching assistants (TAs), as well as immediate feedback to quick quizzes and assignments. MOOCs are a widely researched development in distance education, first introduced in 2008, that emerged as a popular mode of learning in 2012, a year called the "Year of the MOOC".

Early MOOCs (cMOOCs: Connectivist MOOCs) often emphasized open-access features, such as open licensing of content, structure and learning goals, to promote the reuse and remixing of resources. Some later MOOCs (xMOOCs: extended MOOCs) use closed licenses for their course materials while maintaining free access for students. [\[2\]](#)

- **Virtual Learning Environments (VLE) :**

A Virtual Learning Environment, or VLE, refers to a digital space where learners and educators conduct online courses. Educators may deliver course materials through video presentations, audio recordings, virtual classes or other digital means. VLE gives students access to education no matter where they are.

VLEs are part of the broader concept of e-Learning as students and teachers use various multimedia resources and communication tools to access and deliver educational materials. Virtual learning environments are designed to support teaching and learning through optimized digital classrooms that may include learning management systems (LMS) and other educational technology to encourage learning. [\[3\]](#)

- **Adaptive Learning Platforms (ALP) :**

---

Adaptive Learning Platforms, like a learning management system, are used for content delivery. All course content is loaded into the system where the platform uses algorithms that rely on a learner's interaction with the system to create responsive individual learning paths. As a learner completes their course work, the system adjusts the learning path in real-time by changing variables like the pace of instruction, frequency of assessments and content. In this way, each learner receives a unique learning experience grounded in their needs and skill level.

The system evaluates what the learner knows based on frequent formative assessments. Where learning gaps are identified, the system re-routes the learner back through previous content until the required degree of mastery is achieved. The learner only advances through course content as they demonstrate their mastery of each learning outcome. [\[4\]](#)

---

### 1.1.2 ADVANTAGES OF E-LEARNING PLATFORMS:

E-learning platforms, like any other technological advancements, offer distinct advantages and come with certain limitations. Below is a detailed list highlighting the key advantages of e-learning platforms :

- **Interactive Presentation:** E-learning delivers scientific data in an engaging way.
- **Improved Communication:** There is more communication and interaction taking place between students and teachers.
- **Flexibility:** Learning can happen without traveling anywhere.
- **Self- Paced Learning:** Digital resources, CDs, and the internet.
- **Enhanced Onboarding and Feedback:** Increases retention time reduces user dropoff and provides consistent feedback.
- **Improved Motivation:** Increased motivation of the students of learning.
- **Active Participation:** That promotes active not just passive listening by students.
- **Individual Differences:** Takes into account individual differences among students.
- **Advancing Achievement:** Increases academic achievement among students.
- **Quicker Learning Capacity:** Enhances learning abilities of students maximum as a whole.
- **Enriched Curriculum:** Makes the educational curriculum more enriching.
- **Always online:** All learning materials and courses 24/7. [\[5\]](#) [\[6\]](#)

---

### **1.1.3 CHALLENGES AND LIMITATIONS OF E-LEARNING PLATFORMS:**

As for challenges and limitations that face the E-learning platforms in these days are :

#### **1. Teaching Style and Musings on the Cultural Divide:**

- a. The challenge of comprehending, and of dealing with, the very different ways in which students learn and in which they have been educated, within all of their various cultural contexts.
- b. Teaching methods used will not be appropriate to use in an E-learning.
- c. Meeting diverse learning preferences requires academics to modify teaching materials, which is labor-intensive and cumbersome.

#### **2. Pedagogical E-learning challenges:**

- a. There are different pedagogical approaches for this E -learning like effective online interaction and assessment methods.
- b. Academics will have to adopt new ICT skills and learn how to provide good online courseware and content.
- c. The absence of well-implemented, well-designed courses impacts effectiveness in learning.

#### **3. Technological Challenges:**

- a. Problems in quality of E-learning systems in terms of usability issues, performance problems, and no capacity to correspond to needs.
- b. Academic technical supports are under-resourced and end up frustrating and decreasing end user productivity.
- c. It would be more complicated and problematic to involve various applications and systems in the E-learning environment.

#### **4. Main problem of Technical Training:**

- a. Lack of proper training for teachers in order to make them aware of how to use E learning systems efficiently.
- b. Training is frequently not something academics want to do or gives them enough tried and tested advice.
- c. The training for Academics is to Learn How to Create and Make Pedagogically Sound Materials on the Online Medium as well as the Technical Aspect.

#### **5. Time Management Challenges:**

- a. It may take educators more time to incorporate e-learning than traditional methods of instruction.
- b. Academics are expected to be always online which is hard to maintain with other duties.

- 
- c. Low morale is likely to reduce student learning when teachers are overworked, the article says. [\[7\]](#)
- 

#### 1.1.4 CURRENT TRENDS AND FUTURE DIRECTIONS OF E-LEARNING PLATFORMS:

After the tremendous spread of e-learning platforms currently, we can extract from them the following trends :

- **Multimedia Learning** : Utilizes a combination of media such as audio, video, images, and music to create interactive and rich media learning experiences.
- **Technology-Enhanced Learning** : Learning aided by or delivered through technology like the internet, computers, and video conferencing. Offers cost savings, personalized feedback, and a flexible learning environment.
- **Computer-Based Instruction** : Uses computers to enhance learning, providing an engaging and motivating environment through multimedia tools.
- **Computer-Managed Instruction** : Involves using computers for testing, analysis, record-keeping, and guiding students' progress.
- **Computer-Based Training** : Focused on both educational and industrial training, though expensive due to the required programming skills.
- **Computer-Assisted Instruction** : Automated learning system using simulations, providing realistic and varied learning experiences at lower costs.
- **Web-Based Learning** : Online learning that initially faced challenges but has become an effective and less expensive method for training.
- **Virtual Learning Environments** : Web-based learning that simulates a classroom environment, supporting both synchronous and asynchronous learning.
- **M-Learning (Mobile Learning)** : Learning through portable devices like MP3 players and handheld computers, offering convenience and mobility.
- **Selective Open Online Courses (SOOCs)** : Similar to MOOCs but with selective enrollment, ensuring higher active participation and retention rates.

Now for the future direction of e-learning platforms, there is lot of promises such :

- **Sensitive Open Online Courses (SEOOCs)** : Courses with screening tests to identify students' attitudes and perceptions towards online learning, focusing on psychological and cognitive factors to improve retention and effectiveness.

- 
- **Integration of Artificial Intelligence** : Emphasis on incorporating the latest technologies to enhance the learning experience, including artificial intelligence and machine learning for personalized learning paths.
  - **The Focus on Learner Perception** : Greater attention to understanding and catering to the psychological and behavioral aspects of learners to reduce dropout rates and improve outcomes.
  - **Blended Learning Models** : Combining traditional face-to-face learning with online components to create a more flexible and effective educational experience.
  - **Increase of Usage for Mobile Technologies Learning** : Continued growth in the use of mobile devices for learning, providing access to educational content anytime and anywhere.
  - **Enhanced Interactivity and Engagement** : Development of more interactive and engaging content to keep learners motivated and involved in the learning process.
  - **Global Reach and Accessibility** : Expanding the reach of E-learning to provide education to a global audience, breaking down geographical barriers.
  - **Data-Driven Insights** : Utilizing data analytics to gain insights into learner behavior and outcomes, allowing for continuous improvement of E-learning programs. [\[8\]](#)
- 

## 1.2 DEEP LEARNING IN EDUCATION

Deep learning is one of the emerging technologies within the broader context of machine learning, and its transformative power has come to offer the additional potential to change educational practices. Deep learning architectures conduct automatic feature extraction out of extensive data using artificial neural networks with a large number of layers to be able to understand complex designs and make predictions with high accuracy. The applications of deep learning in education are diverse, spanning from personalized learning to predictive analytics, content creation, and automation in the administration of educational institutions.

---

## 1.2.1 PERSONALIZED LEARNING

One of the most promising applications of deep learning, which is focused on personalization, is in the domain of education. A dynamical change in the educational content is done under the system of customized learning in response to the needs, likes of the student, and the pace of learning peculiar to each student. Deep learning models analyze interaction data, performance metrics, and even behavioral patterns among data from many sources to tune the teaching.

| Impact of Personalized Learning Systems | Increase in Student Engagement | Decrease in Course Dropout Rates |
|-----------------------------------------|--------------------------------|----------------------------------|
| Percentage                              | 47%                            | 32%                              |

### *1 . Stats of a study : Student Engagement in Personalized Learning*

Research indicates that personalized learning systems may increase student engagement and academic success by as high as 47%. Course dropout decreases by 32% when customized learning tools are applied. [\[9\]](#) [\[10\]](#)

---

## 1.2.2 PREDICTIVE ANALYTICS

The predictive analytics capability of education is also helpful, whereby institutions can identify and support students who are likely to fail or drop out of studies. Deep learning models are enabled to predict future performance by digging through historical data and trends, therefore serving actual notice to fellow educators on timely interventions and support that can be given to at-risk students.

| Impact of Predictive Analytics | Improvement in Academic Performance | Early Identification of At-Risk Students |
|--------------------------------|-------------------------------------|------------------------------------------|
| Percentage                     | 23%                                 | Significant                              |

### *2 . Stats of a study : Effectiveness of Predictive Analytics*

Research highlights the effectiveness of predictive analytics in improving retention rates and academic performance. A study showed that predictive analytics could improve academic performance by 23%. [\[9\]](#) [\[10\]](#)

---

---

### 1.2.3 CONTENT CREATION & ENHANCEMENT

Deep learning is equipped to develop and fine-tune educational content through content creation. Right from generating personal quizzes to different study materials, the models used in natural language processing create these. AI-assisted content creation tools include Quizlet and Duolingo, which use deep learning to create personalized content for education adapted to every learner's needs and the feedback about their learning process. Moreover, any existing composition can be developed further while the process is underway by using appropriate models available through the deep learning infrastructure, with real-time feedback for such improvement. This can even be taken further through automated grading systems where deep learning algorithms could score a written assignment and comment on it constructively. In this way, it also lessens the burden on educators. [\[9\]](#) [\[10\]](#)

---

### 1.2.4 AUTOMATED ADMINISTRATIVE TASKS

Besides improving student learning, deep learning can also automate many administrative tasks for an education system. These could be class scheduling, registering students, allocating resources, and many others. For instance, chatbots based on deep learning can aid in answering some basic queries of students or even parents instantly so that the administrative workforce can be indulged in other essential things. This increases the efficiency of doing the tasks and at the same time leads to better management of institutions concerned with education. [\[9\]](#) [\[10\]](#)

---

### 1.2.5 CHALLENGES AND CONSIDERATIONS

Despite its potential, the application of deep learning in education comes with several challenges:

1. **Data Privacy** : Deep learning models demand big data sets, which, most of the time are rich with sensitive student information. This is the reason that the privacy and security of data are vital in using these models. Impose stringent requirements for data handling by institutions operating under the regulations, such as the General Data Protection Regulation in the European Union.
2. **Model Interpretability** : Deep learning models are complex and not easily explained; hence, they are usually known to be "black boxes," a severe issue for

---

application in educational systems where accountability and transparency are crucial.

3. **Resource and Expertise Requirements** : Applying deep learning models in education has some resource and expertise requirements. Educational institutions might, therefore, face issues, including infrastructural support, to ensure that skilled personnel are available to develop, maintain, and interpret models. This is where collaborations with technology providers and continuous training for teachers and administrators are necessary in such situations. [\[10\]](#)
- 

## 1.2.6 CASE STUDIES AND REAL-WORLD APPLICATIONS

Here are two case studies and a real-world application showing how deep learning implementations in the education sector have succeeded in doing so:

1. **Deeper Learning Networks** : The Learning Policy Institute conducted an inter-case analysis that involved seven districts implementing strategies to support deeper learning. Initially, strategies focused on student-centered learning approaches and deep learning was brought into play to improve the educational outcome. Case studies further proved this, that schools using deep learning strategies had students who were performing better and taking more interest in their studies.
  2. **Statistical and Deep Learning in Precision Education** : An article from the Educational Technology Journal is on combining precision education in Taiwan by applying statistical learning and deep learning techniques. This uses material according to the needs and learning patterns of a student. From the case study, it is established that with deep learning in education, one can take on unique learning styles to assist students in bettering their outcomes.
- **Real-World Application: Duolingo** :

It is one of the most popular language-learning applications developed with deep learning algorithms to personalize learning. The app also automatically modifies the level of lesson difficulty according to the performance and progress of a learner. For its part, deep learning is employed to create numerous chatbots, which, in turn, help the learner to be able to practice conversations in different languages and eventually get instant corrections and feedback on the same. Such a service endows the Duolingo app with the practical applications of deep learning, making language learning accessible and effective for millions of users around the globe.

---

---

## 1.2.7 FUTURE DIRECTIONS

Now, for an overview of their future avenues in deep learning in education, the following areas of inquiry are highlighted and their sources used:

1. **Enhanced Personalization** : Deep learning algorithms will enable educators to parse massive student data to identify learning patterns and customize instructional strategies for learners. This individualized approach guarantees that every student receives the support they need, ultimately leading to better academic results.
2. **Multidisciplinary Integration** : Only when deep learning is taken in along with the lately developed technologies of augmented reality (AR) and virtual reality (VR) will the learning become immersive and interactive. Such integrations will be much more engaging and practical for the students.
3. **Ethical AI** : Ethical guidelines and the conceptual framework regarding the use of AI in education should be developed to ensure fairness, transparency, and accountability. Addressing Ethical Concerns: Addressing ethical issues in deploying AI-based technologies in educational setups.
4. **Creating Scalable Solutions** : Developing scalable deep learning solutions for easy deployment in academic institutions with different levels of resource constraints. This will include the development of affordable and accessible technologies that may be implemented across various educational contexts. [\[11\]](#) [\[12\]](#)

## 1.3 VIDEO ANNOTATION & LINKED DATA

The textual annotation of video content has emerged as a crucial tool for deepening educational experiences. Such a method makes the video content more searchable, navigable, and comprehensible. An increased educational value of video content is achieved when linked data bridges the annotation to the outside world. Such a combination kindles off a better, more interactive learning experience: a cartridge between concepts, offering a better understanding for students.

### 1.3.1 VIDEO ANNOTATION

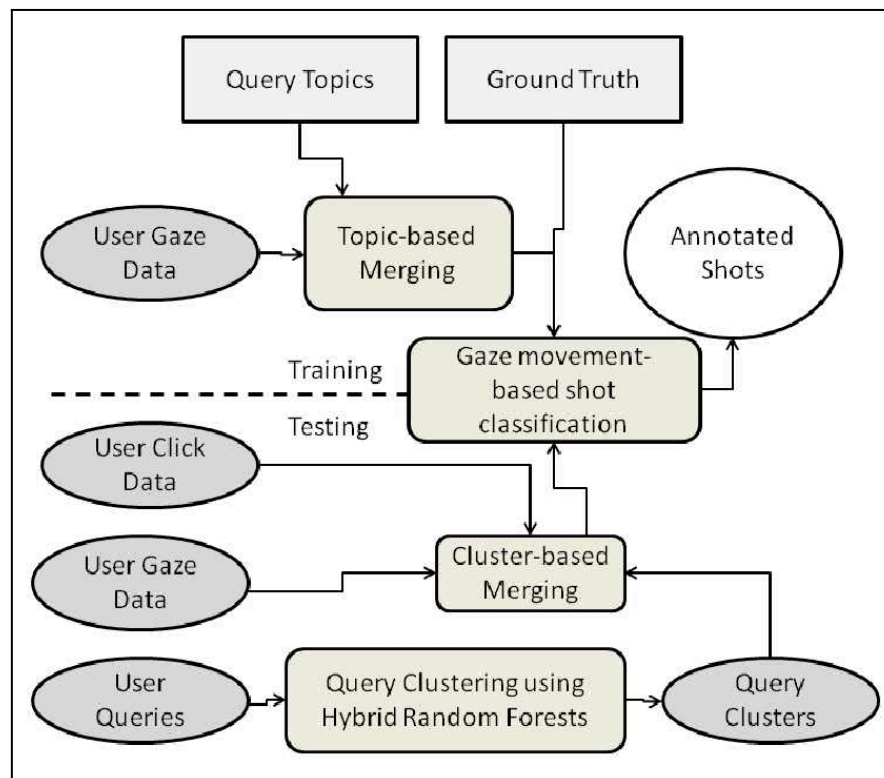
Video annotation is a process by which metadata can be added to video content to make it more easily accessible and richer in educational value. Video metadata can be in the form of

textual comments, labels, links, or even interactive elements such as quizzes or questions at a particular time for the video content. The video is annotated to make it more prosperous, compelling, and contemporary, and to provide additional context and information for learning. [\[13\]](#)

## 1. What is Video Annotation for ?

Video annotation adds labels, annotations, or metadata to video data, enhancing its understanding and analysis by machines. Video annotation involves tagging it with information on the scenes, characters, and events.

These annotations allow for accurate recognition, tracking, and interpretation of visual features by providing context information to algorithms and models. Video annotation aids in the transformation of raw video data into useful insights by labeling items or regions of interest within frames or sequences.



**Fig 01 :** Explains how does the Video Annotation Framework work [\[14\]](#)

## 2. Types of Video Annotation

Many types of annotations can be made on videos, and each of them serves different analysis and interpretation. These are some of the most common video annotations.

- 
- **Object Tracking** : Object tracking is monitoring the movements of a predefined moving object within a video. The annotation is essential for tasks including surveillance, driverless vehicles, and action recognition.
  - **Activity Recognition** : Here, one annotates videos based on the information for activity recognition, which implicates finding and tagging instances of human activity. Such annotations are helpful in areas as varied as video surveillance, sport analyses, and human-computer interaction.
  - **Scene Segmentation** : It is an annotation of a video whereby the video is separated into different scenes or segments through a difference in setting, movement, or subject matter. This annotation allows video summarization, video retrieval based on content, and video editing.
  - **Emotion Recognition** : This annotation is expected to recognize and label the emotional expressions or states of people in videos. The technology builds on a fusion of technologies in affective computing, psychological study, and sentiment analysis.
  - **Spoken Words or Dialogue Transcript and Labeling** : This annotation transcribes and labels spoken words or dialogue in a video. This annotation benefits video indexing, automatic subtitling, and transcription. [\[15\]](#)

### 3. Benefits of Video Annotation in Education

**Enhanced Engagement** : Video annotation will make the video content more interactive, where students can engage with the content in various ways. This can be in terms of quizzes, questions, and comments in the annotations that would have been made, which help the student ask critical questions.

**Improved Navigation** : Annotations offer easy navigation of video content for students. This can be achieved by adding labels as well as links. One can give a relevant link to a section in the video that suits the learning requirements of the students.

**Extra Context and Information** : A video annotator can take advantage of adding a link to another resource like an article, website, or other videos. That gives the students a broader view of what is being taught as it avails all these resources to them.

**Collaboration** : It becomes easy to collaborate with the students when the instructors have added annotations. Also, teachers can leave feedback directly on the video, through which students discuss topics, and teachers can send it back.

---

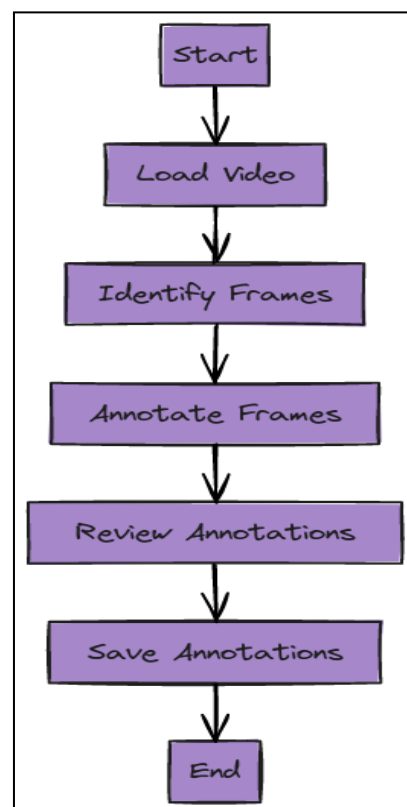
**Enhanced Retention** : Video annotations are interactive and help better retain what has been taught. When students practically engage with the content, they are bound to maintain and grasp more. [\[16\]](#)

#### 4. Techniques and Tools for Video Annotation

We have mainly 5 techniques for video annotation, following :

- **Manual Annotation** : Manual annotation is a traditional technique where human annotators watch videos and mark specific objects or events of interest. It is laborious and requires skilled human labor.

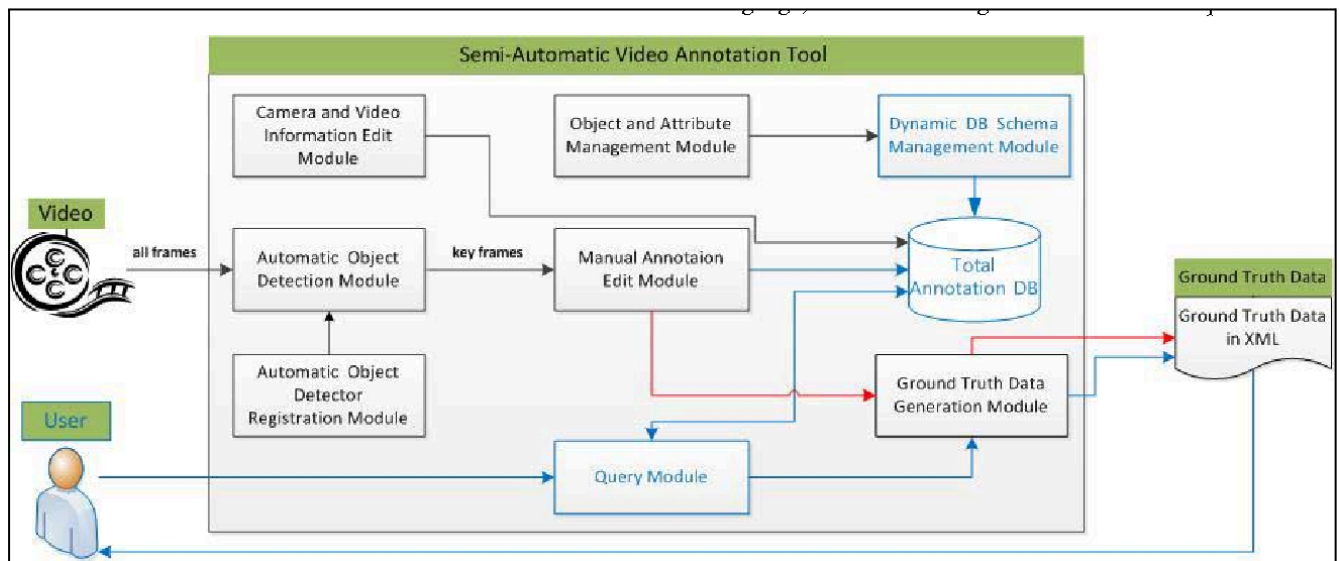
Manual annotation, on the other hand, provides highly accurate labeling and hence works well with high-quality training data. It's commonly employed when automation is difficult or when human judgment is required to complete the annotation process.



**Fig 02.** A Simple Explanation of how does the Manual Video Annotation Works

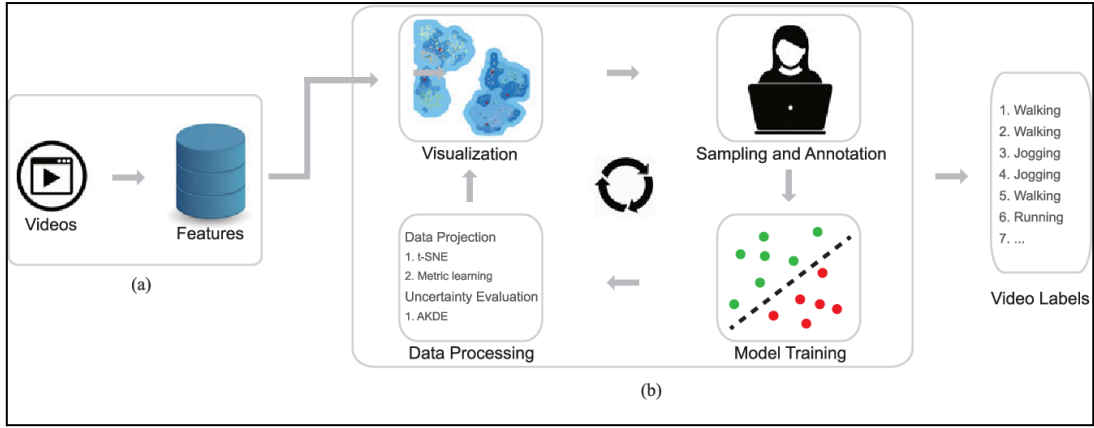
- **Semi-automatic annotation** : it is the concatenation of human knowledge and automated methods. For example, in annotation, computer vision techniques are applied, which make it easier for the annotators to make their annotations. These algorithms can then automatically detect the object or event and track it, thus reducing the amount of manual annotation.

The correctness of automated annotations is verified and corrected by a human annotator. In general, annotation with semi-automatic methods is done in a fraction of the time it would take for manual annotation, without losing accuracy.



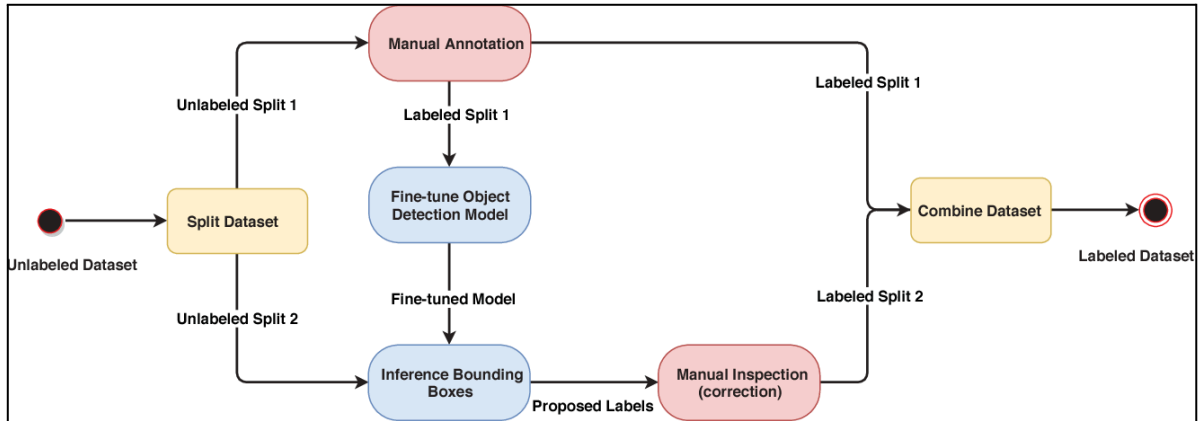
**Fig 03.** System Architecture of semi-automatic video annotation tool [18]

- **Active learning** : A technique used for active learning that selects only the most informative samples for annotation. This approach limits the time spent annotating to the most valuable samples. Compared to traditional methods requiring manual annotation, active learning strategies iteratively focus only on the samples of the video dataset that are particularly challenging to classify or exhibit ambiguity.



**Fig 04.** Visualization-Based Active Learning Pipeline [17]

- Bounding Boxes and Object Tracking :** Bounding boxes are the most common annotation method in use. They annotate a given video frame by drawing rectangles around the relevant elements. These boxes are a depiction of the content and limitations of an object.



**Fig 05.** The workflow of bounding box annotation for object tracking method [19]

- Semantic Segmentation :** This allows annotators to label each pixel in a video frame with the category an object belongs to. It is useful in the interpretation of scenes, translations from image to image, and segmentation of video, in that it provides precise information about the object boundaries. [20]

## 5. Challenges and Considerations for Video Annotation

Video annotation is crucial in various fields, such as computer vision, machine learning, and data analysis. Labeling and tagging specific objects, activities, or events inside a video requires

---

automated systems to interpret the visual data correctly. Annotating videos has many potential benefits, but there are also several obstacles and things to consider :

### **Subjectivity and inter-annotator variability :**

Human subjectivity and inter-annotator heterogeneity in video annotation are major issues. Inconsistent annotations are happening because different annotators describe the same video differently in light of variations in perception, biases, and individual understanding.

For dealing with the above problem, to define proper standards in annotation is mandatory. Such standards would very clearly define labeling criteria to bring a lot of uniformity to the annotations. Regular training sessions and discussions could bring the annotators' interpretations more in line and, therefore, could reduce subjectivity. Apart from using several annotators, it can identify and fix the differences by using the inter-annotator agreement metrics.

### **Balancing accuracy and efficiency in the annotation :**

There is a trade-off between getting a fast annotation and getting a fully precise one. Precise annotations can be very time-consuming, especially for large datasets. Pre-labeling with pre-trained models and active learning can help in this regard. These strategies reduce manual efforts, whereas maintaining accuracy through iterative feedback and quality control.

### **Dealing with complex video content and occlusions :**

However, video annotation is challenged by complex content, such as occlusions, motion blur, low resolution of frames, and crowded scenes. An occlusion is a major challenge, as the object may be hidden, and this complicates the annotation, especially in crowded and dynamic scenes.

Sophisticated object-tracking algorithms could handle occlusions. Multiple annotators and consensus-based approaches help boost accuracy. Specialized annotation tools with features that support magnification, image augmentation, and frame-by-frame analysis are also useful in annotating intricate video content. [\[20\]](#)

---

## **1.3.2 LINKED DATA**

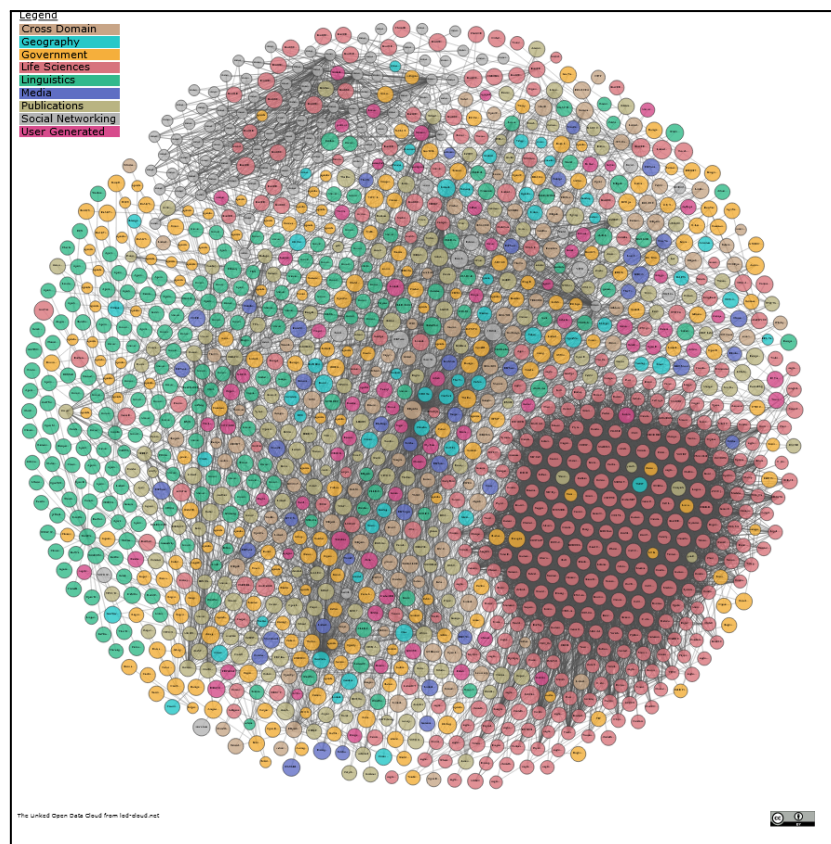
### **1. What is Linked Data ?**

---

The term Linked Data refers to a set of best practices for publishing structured data on the Web. These principles have been coined by Tim Berners-Lee in the design issue note Linked Data. The principles are:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information.
4. Include links to other URIs. so that they can discover more things.

The idea behind these principles is, on the one hand, to use standards for the representation and the access to data on the Web. On the other hand, the principles propagate to set hyperlinks between data from different sources. These hyperlinks connect all Linked Data into a single global data graph, similar as the hyperlinks on the classic Web connect all HTML documents into a single global information space. Thus, LinkedData is to spreadsheets and databases what the Web of hypertext documents is to word processor files. The Linked Open Data cloud diagrams give an overview of the linked data sets that are available on the Web. [\[37\]](#)



**Fig 17.** The linked open data cloud diagram

---

## 2. Benefits of Linked Data in Education :

- Linked data formats (like RDF) work well for mass editing.
- Multiple ways of describing items (data models, ontologies, hierarchies) can exist within the same system (even conflicting values!).
- Linked data can be edited, queried, and updated quickly.
- By virtue of linking, more relevant resources will be more visible/easily discoverable to users.
- The more information there is associated with an item, the easier it is for users to pivot to related concepts, associated images, or similar items in a click. [\[38\]](#)

## 3. Tools and Technologies of Linked Data :

Linked Data leverages a variety of tools and technologies to interconnect and utilize data across different domains and platforms. Here are some key tools and technologies commonly used in Linked Data :

- **Data Formats and Standards :**

**RDF (Resource Description Framework) :** RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.

RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple”). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications.

This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This *graph view* is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations. [\[39\]](#)

**RDFS :** RDFS is a general-purpose language for representing simple RDF vocabularies on the Web. Other vocabulary definition technologies, like OWL or SKOS, build on RDFS and provide language for defining structured, Web-based

---

ontologies which enable richer integration and interoperability of data among descriptive communities. [\[40\]](#)

**OWL (Web Ontology Language)** : The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies. OWL is part of the W3C's Semantic Web technology stack, which includes RDF, RDFS, SPARQL, etc.

The current version of OWL, also referred to as "OWL 2", was developed by the [W3C OWL Working Group] (now closed) and published in 2009, with a Second Edition published in 2012. OWL 2 is an extension and revision of the 2004 version of OWL developed by the [W3C Web Ontology Working Group] (now closed) and published in 2004. The deliverables that make up the OWL 2 specification include a Document Overview, which serves as an introduction to OWL 2. [\[41\]](#)

- **Query Languages :**

**SPARQL** : RDF is a directed, labeled graph data format for representing information on the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs. [\[42\]](#)

- **Data Serialization Format :**

**Turtle (Terse RDF Triple Language)** : A compact, human-readable format for RDF data.

**JSON-LD (JavaScript Object Notation for Linked Data)** : A JSON-based format to serialize Linked Data, designed to be simple for humans to read and write.

**RDF/XML** : An XML-based syntax for RDF.

---

- **Ontologies and Vocabularies :**

**FOAF :** FOAF is a project devoted to linking people and information using the Web. Regardless of whether information is in people's heads, in physical or digital documents, or in the form of factual data, it can be linked. FOAF integrates three kinds of network: *social networks* of human collaboration, friendship and association; *representational networks* that describe a simplified view of a cartoon universe in factual terms, and *information networks* that use Web-based linking to share independently published descriptions of this interconnected world. FOAF does not compete with socially-oriented Web sites; rather it provides an approach in which different sites can tell different parts of the larger story, and by which users can retain some control over their information in a non-proprietary format. [\[43\]](#)

**SKOS :** SKOS is a common data model for sharing and linking knowledge organization systems via the Web.

Many knowledge organization systems, such as thesauri, taxonomies, classification schemes and subject heading systems, share a similar structure, and are used in similar applications. SKOS captures much of this similarity and makes it explicit, to enable data and technology sharing across diverse applications.

The SKOS data model provides a standard, low-cost migration path for porting existing knowledge organization systems to the Semantic Web. SKOS also provides a lightweight, intuitive language for developing and sharing new knowledge organization systems. It may be used on its own, or in combination with formal knowledge representation languages such as the Web Ontology language (OWL). [\[44\]](#)

#### **4. Case Studies and Real World Applications :**

Linked Data is a well-applied technology, presenting solid evidence in a wide set of areas, such as the potential to link and exploit data in ways that were previously unforeseen. Some illustrative case studies and real-world applications are described below :

##### **a. BBC Sports and Music :**

**Description :** For the cases of the BBC Sports and Music websites, Linked Data enables connecting content from different sections of the site and other external resources.

---

**Impact :** Users get a more enriched, webbed experience. For example, sports results have players linked with their profiles, news articles related to the players, and all the historical data of a player.

**Technologies Used :** RDF, SPARQL, and ontology development.

**b. DBpedia :**

**Description :** DBpedia describes a derived dataset that is generated by extracting structured data from Wikipedia and transforming it into RDF. In essence, it results in a dataset capable of being queried and interlinked with other datasets.

**Impact :** DBpedia interlinks information in various domains—for instance, geography, media, and publications—and, therefore, forms the kernel of the Linked Open Data cloud that is live over the web.

**Technologies Used :** RDF, SPARQL, and extraction frameworks.

**c. Europeana :**

**Description :** Europeana is a European cultural heritage digital platform. It also exposes Linked Data that encourages totally different collections and repositories to be interlinked.

**Impact :** Millions of these cultural items through unified access by means of a single entry point for search fosters better discoverability with greater research capability.

**Technologies Used :** RDF, SPARQL, Linked Data principles.

**d. US Government Data (data.gov) :**

**Description :** Data.gov — the portal on which the US government releases the data that governs the different government agencies, because it widely uses and publicizes and interlink through Linked Data.

**Impact :** Transparency in Data, and Public Services innovation powered by enabling complex queries and integration of public data.

**Technologies Used :** RDF, JSON-LD, and SPARQL.

**e. The New York Times :**

**Description :** The New York Times provides linked data to maintain the comprehensive archive of news articles interrelated by content and metadata to enhance search and recommendation

---

**Impact :** It caters for enhanced user experience since users are able to access more relatable and highly related news content, and offer better content management from an internal perspective.

**Technologies Used :** RDF, SPARQL, and Ontology Creation.

**f. Linked Open Vocabularies (LOV) :**

**Description :** LOV is an information base for all the Linked Data vocabularies in use for the Semantic Web and a channel for the discovery and reuse of the Linked Open Vocabularies.

**Impact :** Reuse of standardized vocabularies is promoted to enhance the interoperability and integration of data, thereby improving the quality and connectivity of Linked Data.

**Technologies :** RDF, SPARQL and vocabulary management tools

**g. Library of Congress :**

**Description :** Linked Data is used to maintain the huge bibliographic information by the Library of Congress and to increase the approachability and usability of the information for researchers and the people.

**Impact :** Improves search and retrieval of bibliographic records, better interlinking of metadata, and facilitates more advanced scholarly research.

**Technologies Used :** RDF, SPARQL, and bibliographic ontologies.

**5. Future Directions :**

Linked Data is widely available today, in the LOD cloud and on Web pages. However, the development of applications using Linked Data has been hampered by a series of technical issues, from data quality to complex standards. In the following paragraphs, I give my own vision of the evolution of Linked Data.

1. **Agile standards :** RDF and its applications are governed by a monolithic and complex set of standards that are revamped every few years. In that context, agile and incremental efforts like schema.org will be increasingly popular and important as they correct, update or try out features on a continuous basis, akin to methodologies used for agile software.
2. **Smart clients :** using Linked Data productively is often more complex than it seems, as one typically has to spend considerable time selecting, aligning and cleaning up data (which is a common issue in Big Data and Data Science). Increasingly, Machine Learning methods will be able to automate such processes

---

to create smart Linked Data clients capable of ingesting, aligning and cleaning up raw Linked Data using sophisticated supervised models.

3. **Unification** : Linked Data is available today from several distinct and heterogeneous platforms (the LOD cloud, Wikidata, HTML pages, etc.) In the future, bridges will be built to integrate those platforms and create more extensive webs of data. Fribourg's VoldemortKG is a first effort in that direction, as it interlinks schema.org data to the LOD cloud. [\[45\]](#)
- 

## 1.4 EXISTING SOLUTIONS & GAPS

### 1.4.1 EXISTING SOLUTIONS

Over the last decade, many e-learning systems have been introduced, all involving several technological advances to promote online education. The more common e-learning systems include:

- **Moodle** : is one of the most popular open-source Learning Management Systems; it is used by many institutions. LMS provides very high flexibility, and multimedia, forums, quizzes, and assignments are supported, but it is demanding regarding manual activity, management of courses, and lacks AI-driven advanced personalization features. [\[21\]](#)
- **Blackboard** is a comprehensive LMS offering tools for content management, communication, and assessment. Features include video conferencing, discussion boards, and many more that support multimedia content. Although containing many features, it is quite complex to navigate and offers little in the seamless integration of AI technologies for personalized learning experience. [\[22\]](#)
- **Coursera** is a MOOC platform that partners with universities and organizations to offer online courses, specializations, and degrees. Coursera uses some degree of personalization through the use of adaptive learning paths, but primarily, it serves a large and diverse audience, making it difficult to go deeply personal in the learning experience of the individual user. [\[23\]](#)

### 1.4.2 GAPS IN EXISTING SOLUTIONS

Despite the many advances of e-learning platforms, there are still several gaps that make the delivery of a truly fully personalized and efficient online education difficult to achieve.

---

**Personalize Learning Experience :**

- **Gap :** Most of the existing platforms do not use deep learning algorithms to offer course recommendations in accordance with a user's learning trajectory and preferences. [\[24\]](#)
- **Need :** Implementing AI-driven personalization can enhance student engagement and learning outcomes by tailoring content to the specific needs of each learner .

**Automated Assignment Generation :**

- **Gap :** Nowadays, on most of the e-learning platforms, the teaching staff has to generate the assignments manually, which is a little bit time-consuming and inconsistent.
- **Need :** An AI-powered tool for automated assignment generation can save educators much of their time and effort, in order to ensure that all the assignments made are relevant and appropriately challenging for the students.

**Interactive Video Annotation:**

- **Gap :** Although video content is very common, the ability to place interactive annotations and linked data within video media is not strongly supported.
- **Need :** Providing the video content with annotations and linked data could improve the learning experience, enabling the students to study associated resources and get a deeper understanding of the material

---

---

# CHAPTER 2

---

## CONCEPTUAL STUDY

### 2.1 INTRODUCTION

This chapter conceptualizes the AI-enhanced e-learning platform along with a detailed UML study on system structure and interaction. The mentioned UML diagrams will help to present the design of the system. This includes architecture, components, and how they interact in enabling personalization through course recommendation, AI-driven assignment creation, automatic video transcription, and annotation of the video with linked data.

**The e-learning platform aims to address several key objectives :**

- To offer personalized course recommendations using deep learning models.
- To create AI-driven tools that aid in automated assignment generation and video transcription.
- To implement interactive video annotation enhanced with linked data.
- To evaluate the platform's impact on student engagement, learning outcomes, and instructor efficiency.

The following sections will cover various UML diagrams, including use case diagrams, class diagrams, sequence diagrams, activity diagrams, and component diagrams.

### 2.2 UML DIAGRAMS

UML, short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. [\[31\]](#)

---

UML defines nine diagrams, we will only cite those with which we worked :

## 2.2.1 USE CASE DIAGRAMS

UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system.

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process. [\[32\]](#)

### 1. Actors Defining :

- **Learner** : Interacts with the system to receive course recommendations, view assignments, access transcriptions, and utilize video annotations.
- **Instructor** : Uses the system to upload course content, generate assignments, and annotate videos.

### 2. Use Cases :

- **Personalized Course Recommendations** : Offer course recommendations based on student preferences and history of interactions.
- **Assignment Creation** : The instructor can create assignments directly from the lecture transcripts.
- **Video Transcription Storing** : Create video transcriptions for searchable and accessible lecture video content.
- **Video Annotation** : An instructor and learners provides interactive annotations for video content.
- **Course Content Management** : Allows an instructor to manage course materials by enabling uploading, updating, and deletion of content.

### 3. Use Case Diagram Explanation :

#### a. Course Recommendations :

- A student logs into the system.
- The system fetches the student's profile and interaction history.

- 
- Data is processed by the Recommendation Engine to gain personalized course recommendations.
  - The student's dashboard displays the recommendations.

**b. Assignment Generation :**

- An instructor uploads a lecture video.
- The video gets transcribed by YouTubeTranscriptApi.
- The resulting transcription is stored in the database.
- The Assignment Generator retrieves transcribed video from the database and generates the assignment.
- The instructor can view and refine the generated assignments.

**c. Video Transcription Storing :**

- An educator uploads a video.
- The video gets transcribed by YouTubeTranscriptApi.
- The resulting transcription is stored in the database.
- A video is transcribed by using the YouTubeTranscriptApi.
- The generated model is then stored in the database.

**d. Video Annotation :**

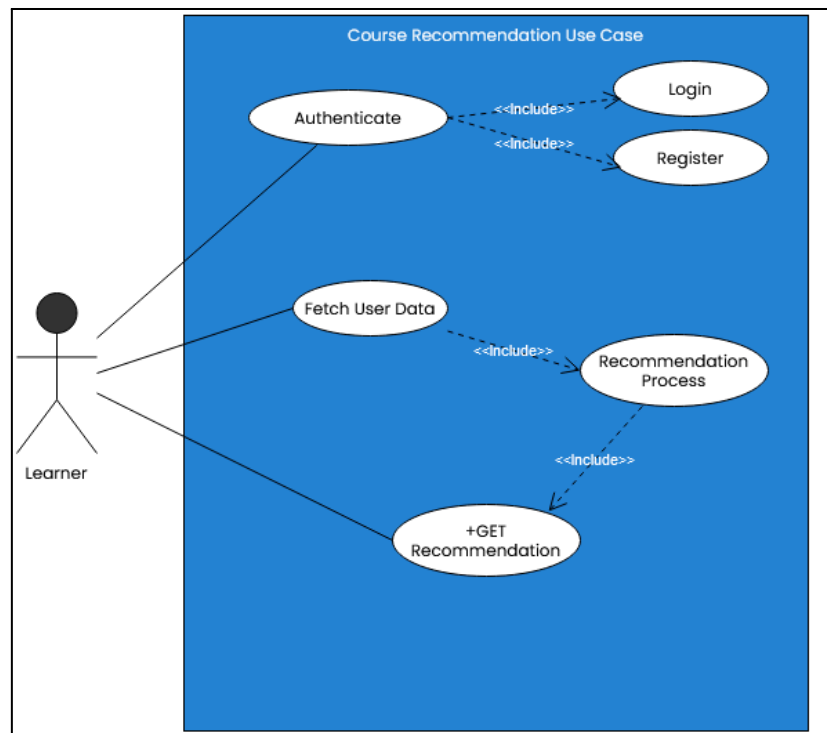
- An educator can add arbitrary text annotations to the educational video at user-defined time-stamp positions.
- An annotation may be a comment, question, or link to some other learning resource.

**e. Course Content Management**

- Instructor manages course materials through an intuitive interface.
- Instructors can upload new content, update existing materials, and delete outdated resources.

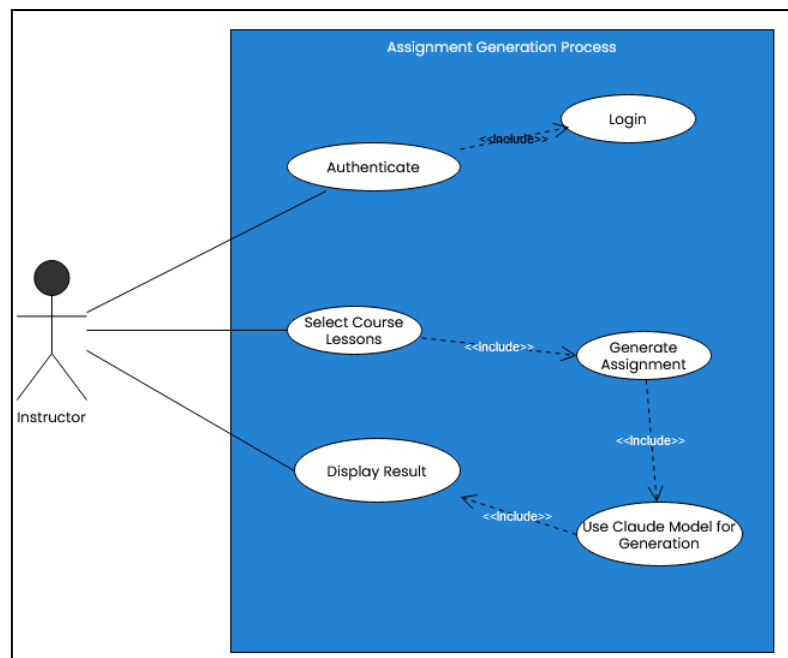
**4. Diagram Visual Representation :**

a. **Course Recommendations use case diagram :**



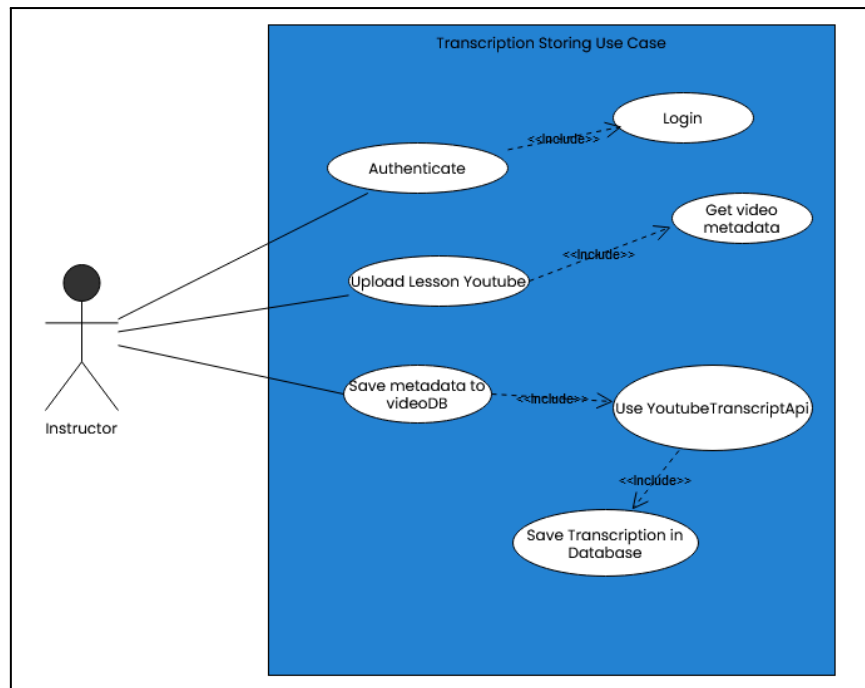
**Fig 06.** Course Recommendation UseCase Diagram

b. **Assignment Generation Use Case Diagram :**



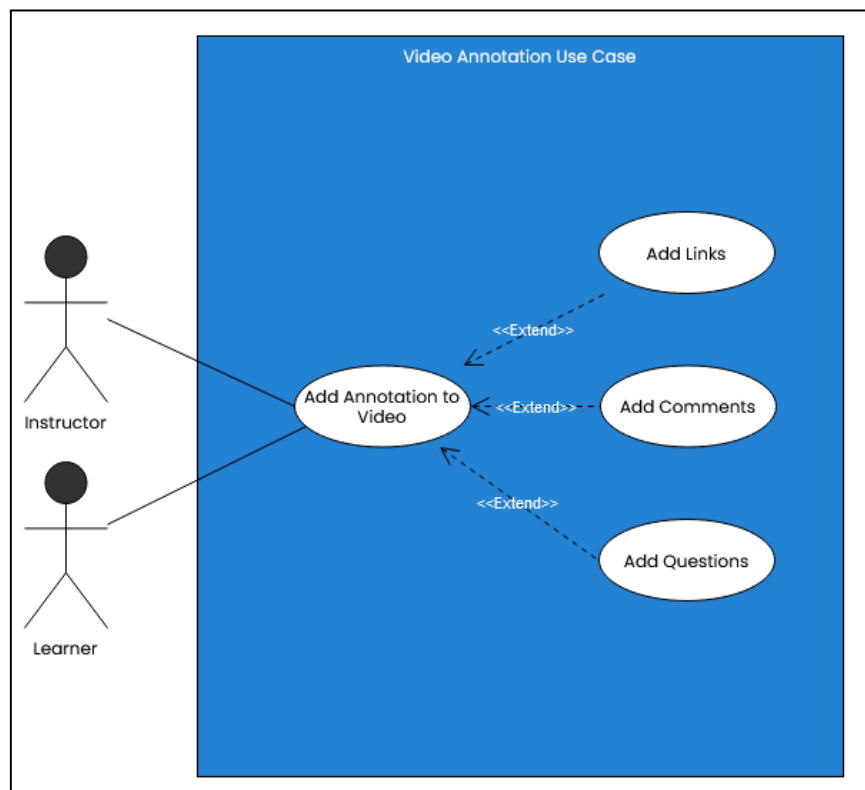
**Fig 07.** Assignment Generation Process Use Case Diagram

c. **Video Transcription Storing Use Case :**



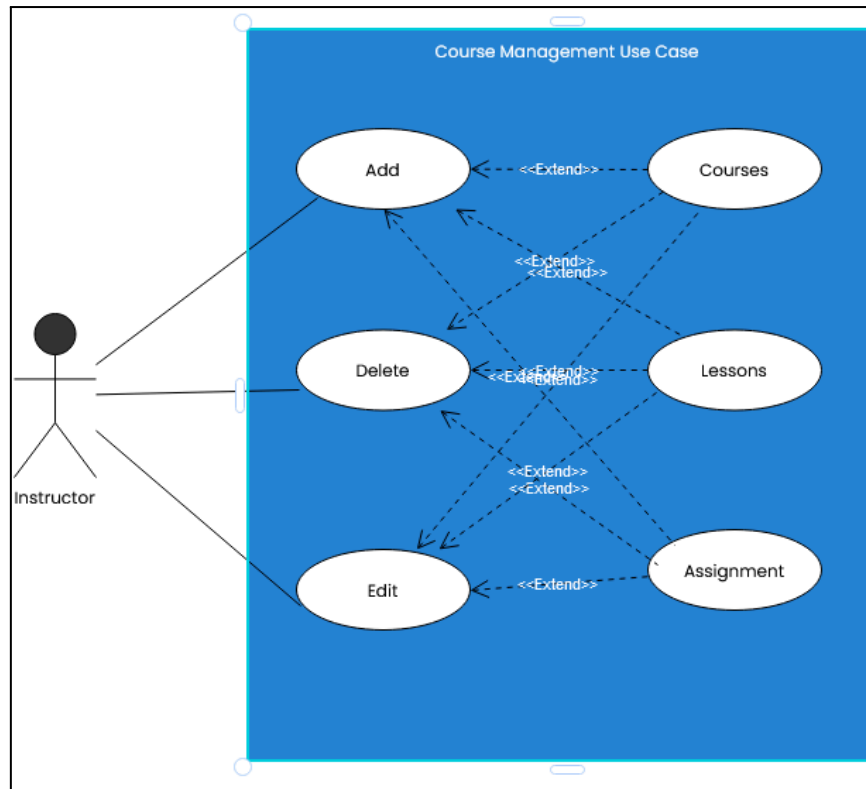
**Fig 08.** Video Transcription Storing Process Use Case Diagram

**d. Video Annotation Use Case :**



**Fig 09.** Video Annotation Use Case Diagram

**e. Course Content Management Use Case :**



**Fig 10.** Course Content Management Use Case Diagram

## 2.2.2 CLASS DIAGRAM

Class diagrams are a type of UML (Unified Modeling Language) diagram used in software engineering to visually represent the structure and relationships of classes in a system. UML is a standardized modeling language that helps in designing and documenting software systems. They are an integral part of the software development process, helping in both the design and documentation phases. [33]

Below is a detailed class diagram for the AI-enhanced e-learning platform :

### 1. Classes and their Description :

#### a. User

- **Attributes :**

id : Integer, first\_name : String, last\_name : String, username : String, email : String, dob(date of birth) : date, agree\_terms : Boolean, phone : String, desc : String, country : String, role : string (Admin/Learner/Instructor).

- **Methods :**

---

login(), register(), logout(), savePreferences(), enroll(),

**b. Course**

- **Attributes :**

id : Integer, title : String, description : String, prerequisites : String, image : String,  
tags : String, difficulty\_level : String, estimated\_duration : String

- **Methods :**

addCourse(), updateCourse(), deleteCourse()

**c. Lessons**

- **Attributes :**

id : Integer, course\_id : Integer, title : String, content : List<Content>, videoID :  
String

- **Methods :**

addLesson(), updateLesson(), deleteLesson()

**d. RecommendationEngine**

- **Attributes :**

trainingData : List<interactions>

- **Methods :**

fetchUserData(), getRecommendation(userId : Integer) : List <Course>

**e. AssignmentsGenerator**

- **Attributes :**

transcriptData : String

- **Methods :**

generateAssignment(transcript : String) : List<Assignment>

**f. Video Transcriber**

- **Attributes :**

---

videoID: String, transcript : String

- **Methods :**

fetchTranscript(videoID : String) : String

**g. Annotation Tool**

- **Attributes :**

annotationID : Integer, content : String, timestamp : Float

- **Methods :**

addAnnotation(), updateAnnotation(annotationID : Integer, content : String),  
deleteAnnotation(annotationID : Integer)

**h. Interactions**

- **Attributes :**

id : Integer, userID: Integer, courseId : Integer, comment : String, rating : Integer

- **Methods :**

addInteraction(rating), updateInteraction(interactionID : Integer, rating :  
Integer)

**i. Assignments**

- **Attributes :**

GoogleForms Reserved#

- **Methods :**

addAssignment(), editAssignment(), deleteAssignment()

**2. Relationships :**

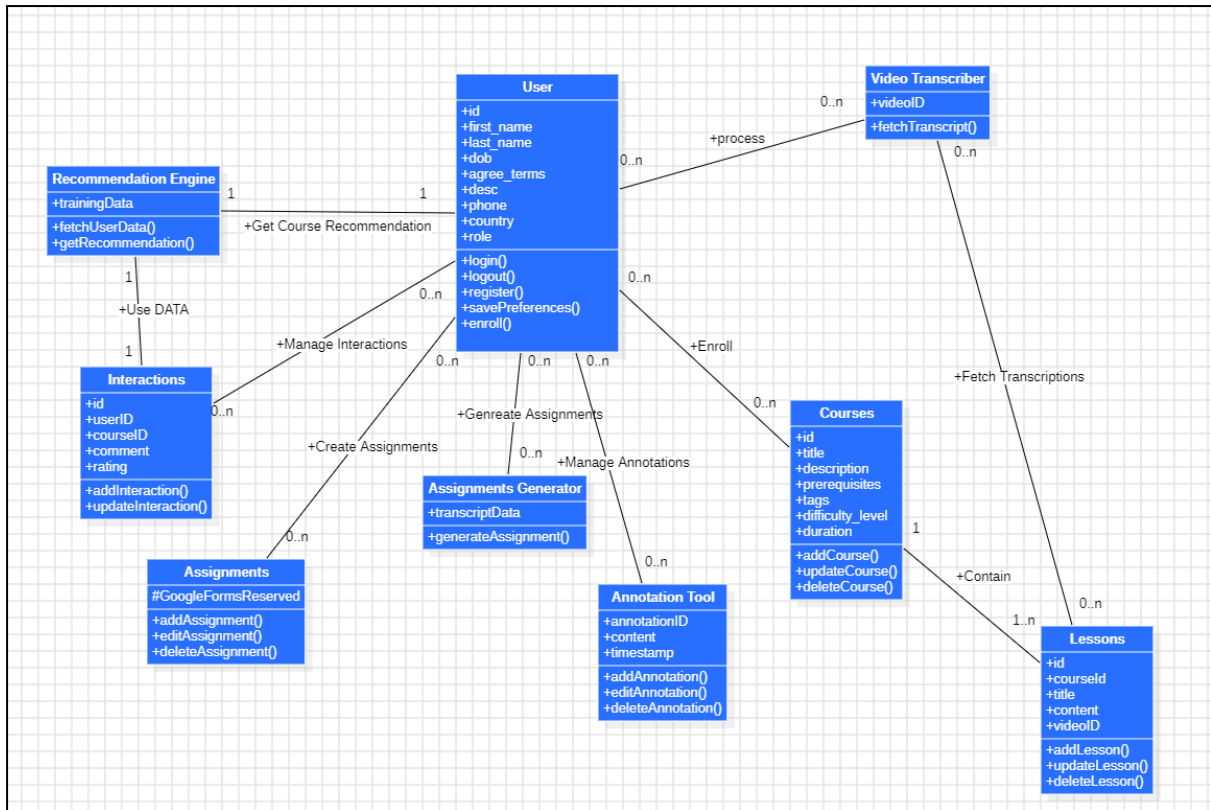
- User to Course:** A user can enroll in multiple courses (many-to-many relationship).
- User to Video Transcriber :** users can start the process of transcribing multiple times for many videos (many to many relationships).
- User to Recommendation Engine :** a user can get recommendation once (one to one relationships).

- 
- d. **User to Interactions** : a user can manage multiple interactions ( many to many relationships).
  - e. **User to Assignments** : a user can create multiple assignments (many to many relationships).
  - f. **User to Assignments Generator** : a user can generate multiple assignments (many to many relationships).
  - g. **User to Annotation Tool** : a user can manage multiple annotations. (many to many relationships).
  - h. **Course to Lessons**: A course contains multiple pieces of content (one-to-many relationship).
  - i. **Video Transcriber to Lessons** : a video transcriber can fetch transcriptions for multiple videos (many to many relationships).
  - j. **RecommendationEngine to Interaction**: The RecommendationEngine uses user interactions as training data. (one to one relationship).

### 3. Explaining Of Key Components :

- a. **User Class** : deals with many classes for many operations, such Video Transcribing, Adding Interactions, Generating Assignments, Course Enrollments, Manage Annotations ... etc.
- b. **Recommendation Engine** : Implements the Surprise model to provide personalized course recommendations based on user interactions.
- c. **Courses Class** : Handles Course information and content management.
- d. **Lessons Class** : Represents various types of educational resources, including videos, assignments, and text information.
- e. **Assignment Generator Class** : Uses the Claude model to generate assignments from video transcriptions.
- f. **Video Transcriber Class** : Uses the YoutubeTranscriptApi to transcribe video lessons.
- g. **Annotation Tool Class** : Allows instructors and learners to annotate video content, which will enhance the learning experience.
- h. **Assignments Class** : Allows the instructor to create assignments in Google Forms platform.
- i. **Interactions Class** : will be managed by the user, and at the same time provides the Recommendation Engine with users data.

### 4. Diagram Visual Representation :



**Fig 11.** Class Diagram of the System

## 2.2.3 SEQUENCE DIAGRAMS

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when. [34]

### Sequence Diagram for Personalized Course Recommendations :

#### 1. Actors :

- Learner
- System (Website)
- Recommendation Engine

#### 2. Sequence of Interactions :

##### Learner Authenticate :

- The student initiates a login request to access the platform.

- 
- The system verifies the student's credentials using the Database.

#### **Request for Course Recommendations :**

- Once logged in, the student requests personalized course recommendations.
- The system retrieves the student's profile and interaction history from the Database.

#### **Data Retrieval :**

- The RecommendationEngine sends a query to the Database to fetch the required data (e.g., past course interactions, ratings).
- The Database responds with the relevant data.

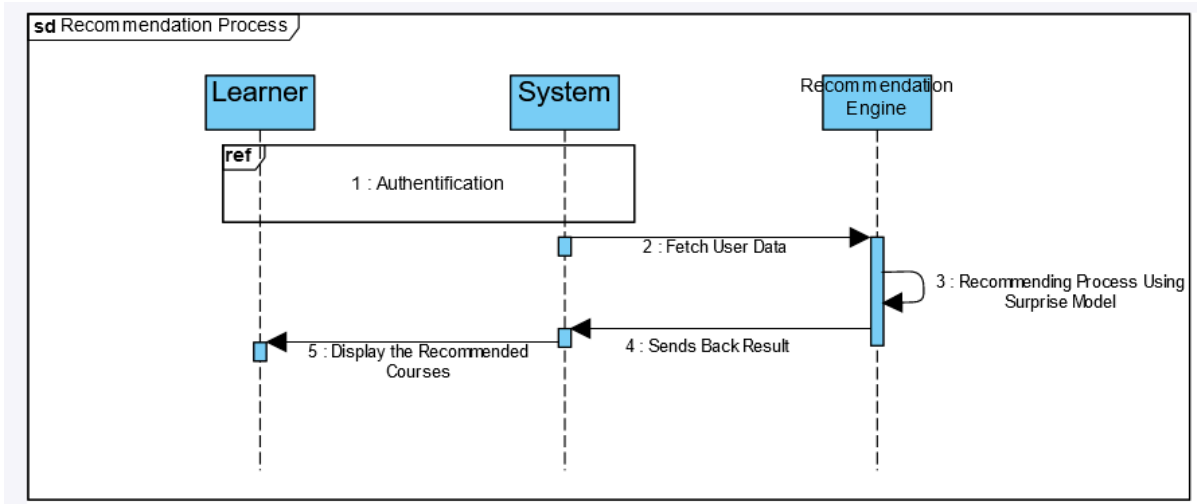
#### **Generate Recommendations :**

- The RecommendationEngine processes the retrieved data using the trained Surprise model.
- The model generates a list of recommended courses for the student.

#### **Display Recommendations :**

- The generated course recommendations are sent back to the student's interface.
- The student views the personalized course recommendations on their dashboard.

### **3. Visual Representation of the Diagram :**



**Fig 12.** Recommendation Process Sequence Diagram

### Sequence Diagram for Authentication :

#### 1. Actors :

- User(Learner/Instructor/Admin)
- System
- Database

#### 2. Sequence of Interactions :

##### User Initiates Login :

- The user (student or educator) enters their login credentials (username and password) and submits the login form.
- The system receives the login request.

##### System Validates Credentials :

- The system extracts the entered credentials and prepares to validate them against the stored data.
- The system sends a query to the Database to retrieve the stored credentials for the given username.

##### Database Responds :

- The Database fetches the stored credentials for the given username.
- The Database returns the retrieved credentials to the system.

##### System Verifies Credentials :

- The system compares the entered password with the stored password (hash) retrieved from the database.
- If the credentials match, the system proceeds to create a session for the user.

##### Session Creation :

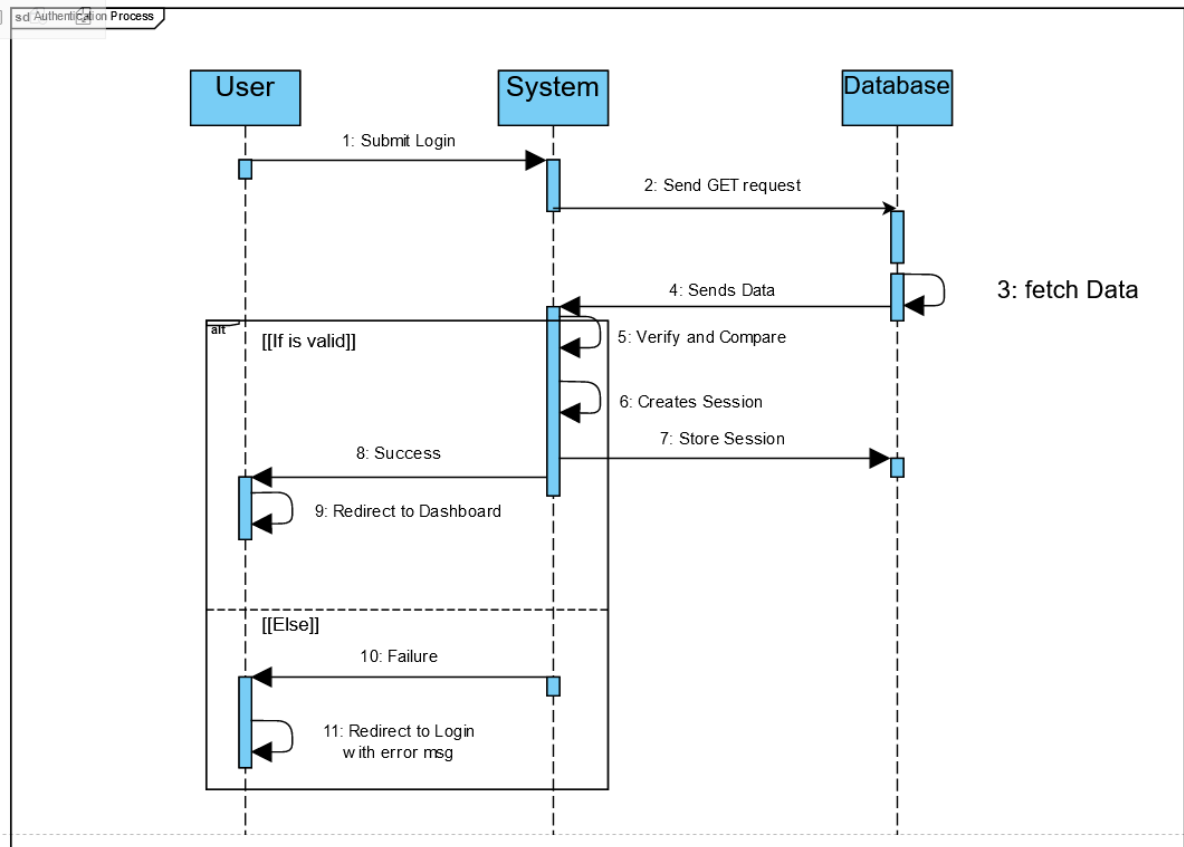
- The system generates a session token and stores it in the database.

- The session token is sent back to the user as part of the response.

#### Login Success :

- The system sends a response to the user indicating a successful login.
- The user is redirected to the dashboard or homepage.

### 3. Visual Representation of the Diagram :



**Fig 13.** Authentication Process Sequence Diagram

### Sequence Diagram for Assignment Generation :

#### 1. Actors :

- Instructor
- System
- Claude
- Database

#### 2. Sequence of Interactions :

##### Transcription List Page :

- The instructor sends a request to visit the page to the system.

- The System sends a GET request to DATABASE to fetch data.
- The Database sends back data to the system.
- The system displays the page.

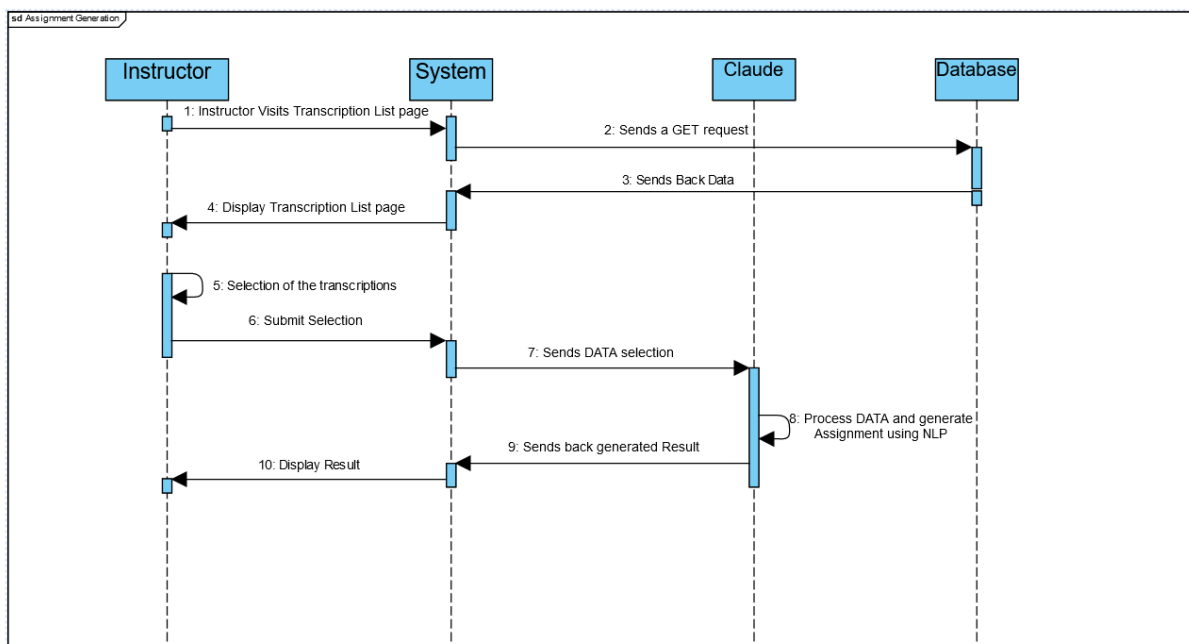
#### Transcriptions Selection :

- The instructor selects the transcriptions from the list that he wishes to generate assignments from.
- The instructor submits his selection to the system.

#### Assignment Generation Process :

- The system sends DATA selection to the Claude Model.
- Claude model uses the data selection to generate assignment text using NLP techniques.
- Claude model sends the result to the system.
- The system displays the result to the instructor.

### 3. Visual Representation of the Diagram :



**Fig 14.** Assignment Generation Sequence Diagram

#### Sequence Diagram for Transcription Storing :

##### 1. Actors :

- Instructor
- System
- YoutubeTranscriptApi
- Database

## 2. Sequence of Interactions :

### Add Video Page :

- The instructor visits the add video page.
- The system sends a GET request to the database.
- The database sends back data to the system.
- The System displays the “add video” page.

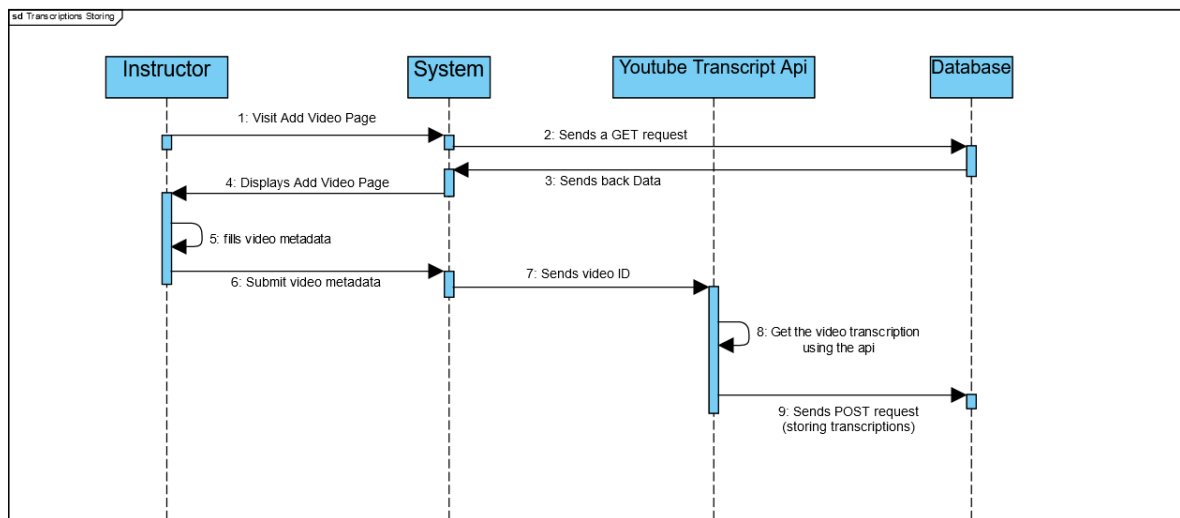
### Video Metadata submit :

- The instructor fills the video metadata on the form.
- The instructor submits the form.

### Video Transcription Process :

- The System sends the metadata to the YoutubeTranscriptApi.
- YoutubeTranscriptApi gets the videoID from the data and uses it to get the transcriptions of the video.
- YoutubeTranscriptApi sends a POST request to the database (Saving Video Transcription).

## 3. Visual Representation of the Diagram :



**Fig 15.** Transcription Storing Sequence Diagram

---

## 2.3 SYSTEM MAIN FUNCTIONALITIES

### 2.3.1 PERSONALIZED COURSE RECOMMENDATIONS

The system brings in a deep learning model from Scikit-learn, called Surprise, for personal course recommendations based on historical student data, such as course enrollments, ratings, and course completion status. The model predicts the most beneficial courses for a student based on their individual learning patterns and preferences.

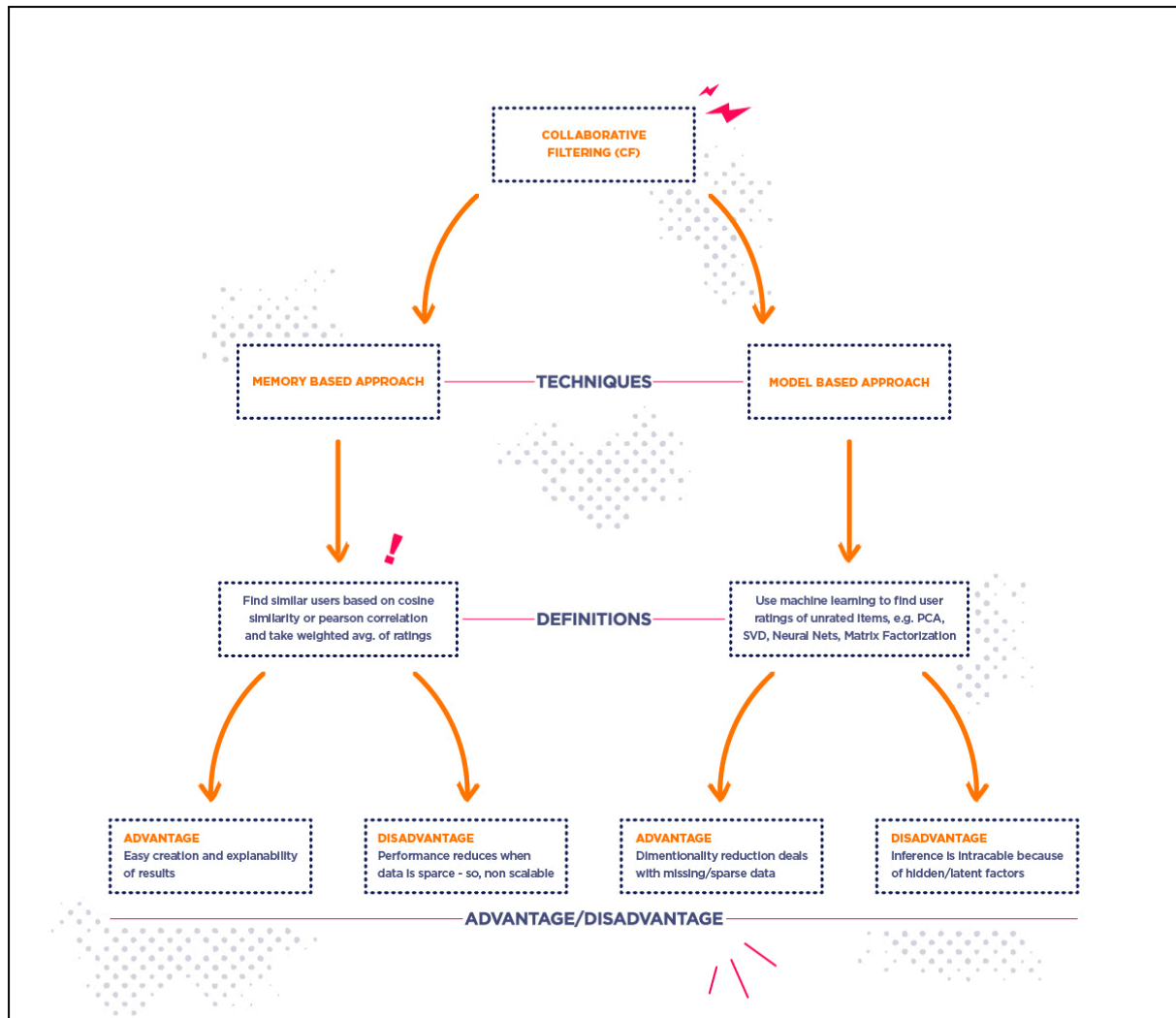
#### 1. Introduction to the Surprise Library:

**Surprise** is one of the “scikit-based Python” libraries to develop recommender systems using explicit rating data. Since Surprise is one of scikit-based libraries, it is relatively easy to use for developers that are quite familiar using scikit-based libraries, such as scikit-learn. Conveniently, Surprise provides several algorithms to develop **collaborative filtering** recommender systems, starting from memory-based method (i.e. K-Nearest Neighbors or KNN) to model-based method (i.e. Singular Value Decomposition or SVD). [\[25\]](#)

#### 2. What is Collaborative Filtering ?

**Collaborative filtering** is a technique used in recommendation systems to suggest items (such as products, movies, or music) to users based on their historical choices and behaviors. This method relies on the patterns and preferences of multiple users rather than requiring extensive details about the items themselves. [\[26\]](#)

Collaborative filtering relies on the relationships between users and items, using ratings and behaviors to determine item similarity. It leverages the collective preferences of a user community to provide personalized recommendations. [\[26\]](#)



**Fig 16.** Techniques used by Collaborative Filtering

### 3. SVD (Singular Value Decomposition) Algorithm

Singular value decomposition (SVD) is a matrix factorization method that generalizes the eigendecomposition of a square matrix ( $n \times n$ ) to any matrix ( $n \times m$ ) [27]

SVD is similar to Principal Component Analysis (PCA), but more general. PCA assumes that input square matrix, SVD doesn't have this assumption. General formula of SVD is:

$M = U \Sigma V^t$ , where:

- $M$ -is original matrix we want to decompose
- $U$ -is left singular matrix (columns are left singular vectors).  $U$  columns contain eigenvectors of matrix  $MM^t$
- $\Sigma$ -is a diagonal matrix containing singular (eigen)values

- 
- V-is right singular matrix (columns are right singular vectors). V **columns contain eigenvectors of matrix  $M^T M$**  [\[28\]](#)

a. **How to Calculate it ?**

i. **Power Iteration**

Power iteration starts with  $b_0$  which might be a random vector. At every iteration this vector is updated using following rule:

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|}$$

First we multiply  $b_0$  with original matrix A ( $Ab_k$ ) and divide result with the norm ( $\|Ab_k\|$ ). We'll continue until the result has converged (updates are less than threshold). [\[29\]](#)

Power method has few assumptions:

- $b_0$  has a nonzero component in the direction of an eigenvector associated with the dominant eigenvalue. Initializing  $b_0$  randomly minimizes possibility that this assumption is not fulfilled
- matrix A has a dominant eigenvalue which has strictly greater magnitude than other eigenvalues.

These assumptions guarantee that the algorithm converges to a reasonable result. The smaller the difference between the dominant eigenvalue and second eigenvalue, the longer it might take to converge. [\[29\]](#)

ii. **Power Iteration for n Singular values**

To get more than just the most dominant singular value from the matrix, we could still use power iteration. We'll implement a new function which uses our previous `svd_power_iteration` function. Finding the first dominant singular value is easy. We could use the previously mentioned function. But how to find the second singular value?

We should remove the dominant direction from the matrix and repeat finding the most dominant singular value [\[30\]](#). To do that we could subtract previous eigenvector(s)

---

component(s) from the original matrix (using singular values and left and right singular vectors we have already calculated):

$$A_{\text{next}} = A - (\text{singular\_value}_i)(u_i)(v_i)^t \text{ [28]}$$

#### 4. **working of the Surprise model :**

##### a. **Data Preparation :**

- i. The data used includes user IDs, course IDs, and ratings given by users to courses.
- ii. This data is typically in the form of a dataframe with columns for user ID, item ID (course ID), and rating.

##### b. **Loading Data :**

The data is loaded into the Surprise framework using the **Dataset** module, which helps in organizing the data for training and testing.

##### c. **Data Training :**

The SVD is used to decompose the user-item interaction matrix into the product of two lower-dimensional matrices. This helps in capturing the latent factors that influence user preferences.

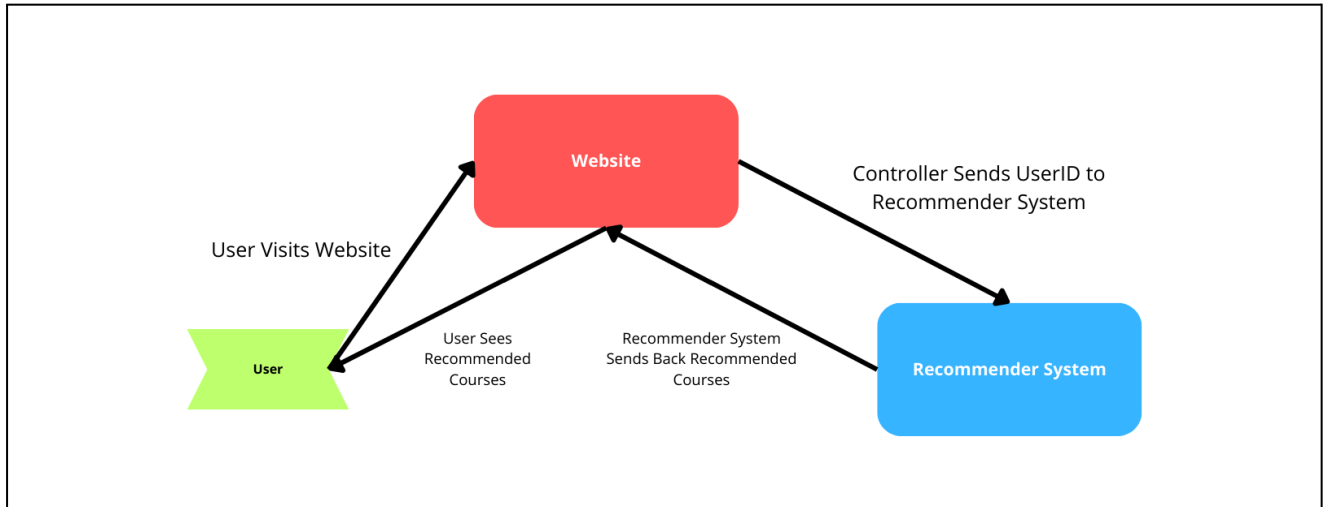
##### d. **Making Predictions :**

Once the model is trained, it can predict ratings for courses that users have not yet rated. These predictions are used to generate personalized course recommendations .

##### e. **Evaluation :**

The performance of the recommendation system is evaluated using metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

## 5. System Design



**Fig 17.** Personalized Course Recommendation System Design Diagram

## 6. Dataset Loading and Cleaning :

Eventually, we've used two datasets in this phase, one for courses, and the other for courses ratings by users.

### a. Courses Dataset :

The dataset used in this study is from the famous elearning website Coursera, consists of information on various courses available on the Coursera platform. {Source}. The dataset includes details about the course names, the institutions offering them, their difficulty levels, course ratings, URLs, descriptions, and the skills they aim to impart. Here is a detailed description of each column in the dataset:

**Course Name:** The name of the course as listed on Coursera.

- Example: "Write A Feature Length Screenplay For Film Or Television"

**University:** The institution offering the course.

- Example: "Michigan State University"

**Difficulty Level:** The intended difficulty level of the course, which can be one of "Beginner", "Intermediate", or "Advanced".

- Example: "Beginner"

---

**Course Rating:** The average rating of the course as provided by users, typically on a scale from 0 to 5.

- Example: 4.8

**Course URL:** The web link to the course page on Coursera.

- Example: "<https://www.coursera.org/learn/write-a-feature-length-screenplay>"

**Course Description:** A brief overview of what the course covers, its objectives, and the topics included.

- Example: "Write a Full-Length Feature Film Script. In this course, you will learn...";

**Skills:** The specific skills that the course aims to teach, often listed as keywords.

- Example: "Drama, Comedy, Screenwriting, Film, Directing"

This dataset provides a comprehensive view of the educational offerings on Coursera, allowing for the analysis of course popularity, difficulty levels, and the specific skills taught. This information is essential for developing and evaluating recommendation systems aimed at suggesting courses to users based on their preferences and past interactions.

#### b. Course Ratings Dataset :

The dataset used in this study, referred to as **ratings.csv**, is derived from the MovieLens dataset, which contains user ratings for various courses. This dataset includes details about the users, the courses they rated, their rating scores, and the timestamps of these ratings. Here is a detailed description of each column in the dataset:

**user\_id:** A unique identifier for each user.

- Example: 1

**course\_id:** A unique identifier for each course.

- Example: 1

**rating:** The rating given by the user to the movie, typically on a scale from 0.5 to 5.0 in increments of 0.5.

- Example: 4.0

---

**timestamp:** The time when the rating was given, represented as a Unix timestamp.

- Example: 964982703

This dataset provides a comprehensive view of user interactions with courses, enabling the analysis of user preferences and rating patterns. Such information is crucial for developing and evaluating recommendation systems aimed at suggesting courses to users based on their preferences and past interactions.

**c. Preprocessing :**

We've used nltk library to preprocess data, where we removed stopwords and unnecessary columns in both datasets such as "timestamp" in rating dataset, and such rating and course url and university from courses dataset we also added a column for courses called "course\_id" were it gives an id for each course.

```
RangeIndex: 3522 entries, 0 to 3521
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   title           3522 non-null   object
1   difficulty_level 3522 non-null   object
2   description      3522 non-null   object
3   skills          3522 non-null   object
4   course_id       3522 non-null   int64
dtypes: int64(1), object(4)
memory usage: 137.7+ KB
```

*Fig 38.* Courses dataframe representation

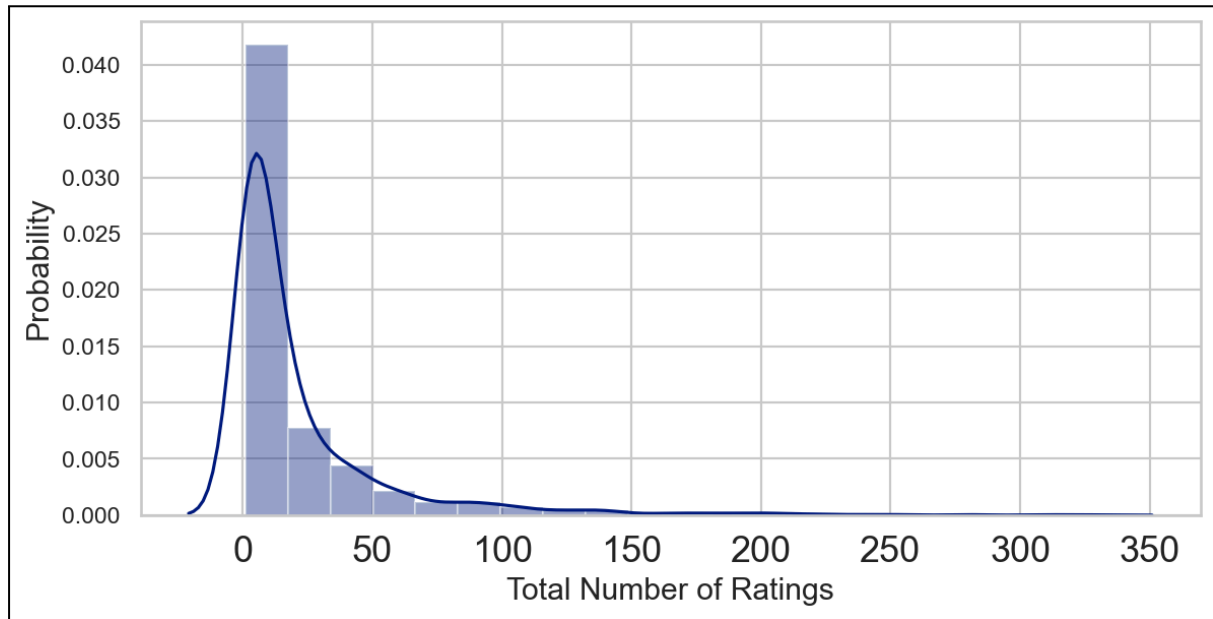
**d. Dataset Merging:**

we merged the two datasets with user\_id on course\_id means that now it shows all the ratings of users for all courses that were rated.



#### f. Dataset visualisation :

Most of these courses stand among the Top 50 courses in the Coursera ratings even today.



**Fig 40.** Total number of ratings probability

We can see that -

- Majority of the courses have less than 50 ratings.
- The number of courses having more than 100 ratings is very low.

Refer one cell above, we can see that there are only 3 courses with 300+ ratings.

#### Highly rated courses :

|                                                              | mean ratings | total ratings |
|--------------------------------------------------------------|--------------|---------------|
| title                                                        |              |               |
| Bugs 101: Insect-Human Interactions                          | 5.0          | 1             |
| Ruby on Rails: An Introduction                               | 5.0          | 1             |
| Introduction to Forensic Science                             | 5.0          | 1             |
| Business English: Networking                                 | 5.0          | 1             |
| Product Design: Designing a modern table lamp using SketchUp | 5.0          | 2             |
| First Steps in Linear Algebra for Machine Learning           | 5.0          | 2             |
| Advanced Grammar & Punctuation Project                       | 5.0          | 1             |
| Research Design: Inquiry and Discovery                       | 5.0          | 1             |
| International Humanitarian Law in Theory and Practice        | 5.0          | 2             |
| Exploratory Data Analysis with MATLAB                        | 5.0          | 1             |

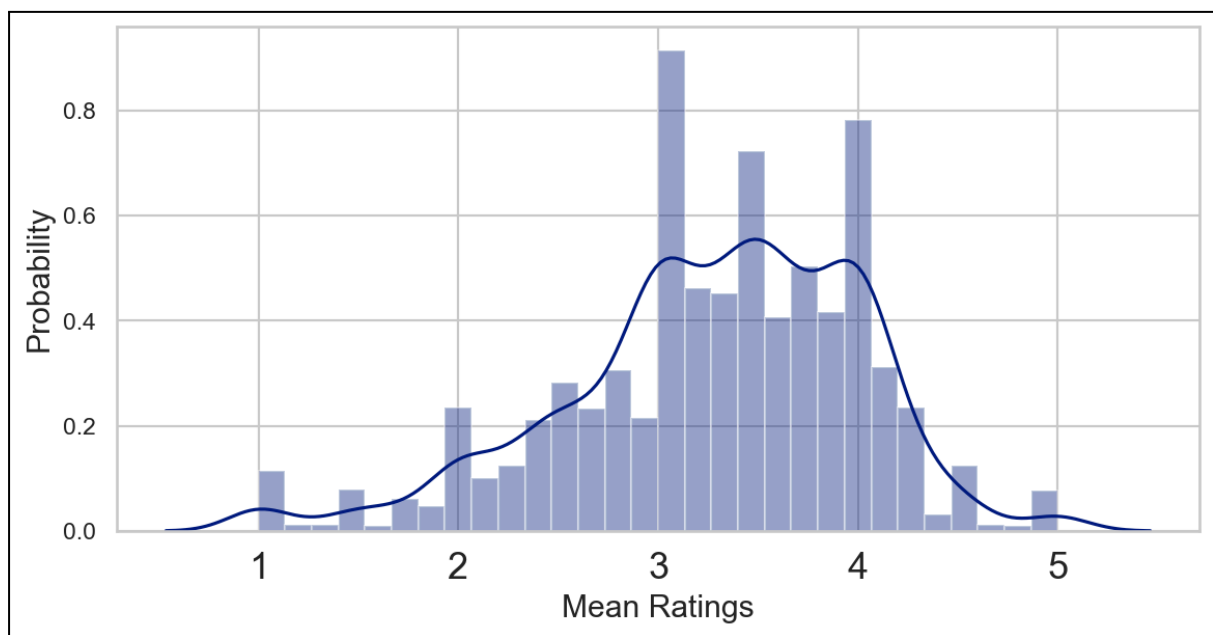
**Table 04.** Mean Ratings of courses

---

Total no of users that gave rating of 5.0 : 26

Total no of Individual users that gave rating of 5.0 : 22

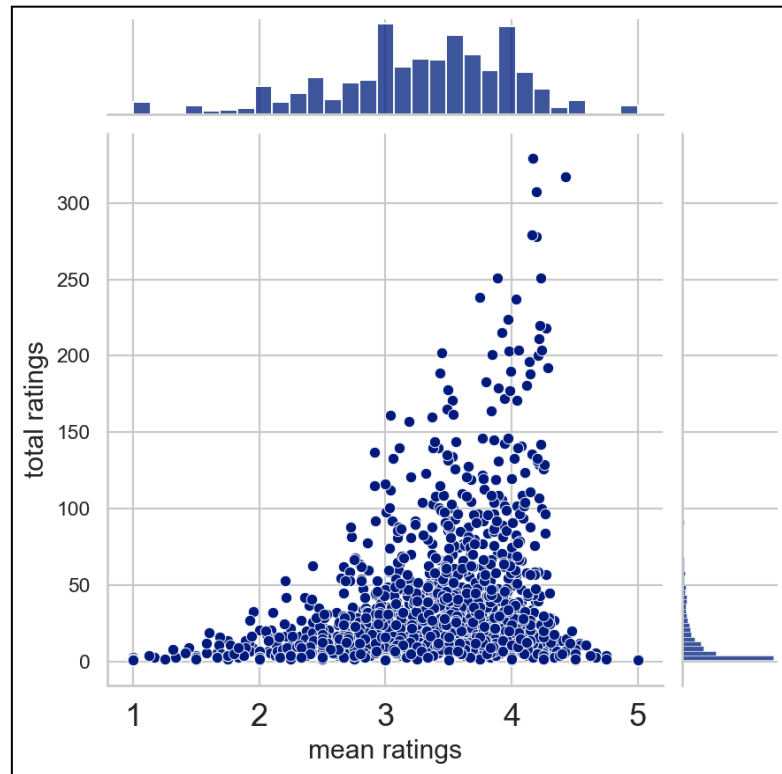
1. As you can see there are over 22 users that have rated 5 stars, among which there are 26 individual raters ( only users rated the course 5 star ).
2. So, this cannot be the lone factor that should be considered while recommending courses. As this factor only shows the preferences of a particular user.
3. It would make a good recommendation system if we can use both the factors ( -> Highly Rated Courses and Heavily Rated Courses <- ) together.



**Fig 41.** Mean Ratings Probability

---

### Mean Ratings vs Total number of ratings :



**Fig 42.** Mean Ratings vs Total number of ratings

Here, as you can see every Data Point represents a distinct Course, with y-coordinate representing the total number of users which have rated that course and x-coordinate representing the mean of all the ratings of the corresponding users.

Also you can see that there is a huge Density in the region corresponding to 0-50 no of users and between mean rating 3-4 .

Many more Visualizations can be drawn and many different conclusions can be inferred, But here, I'm going to focus on Collaborative Filtering.

## 7. Matrix factorization using Deep Learning (Keras)

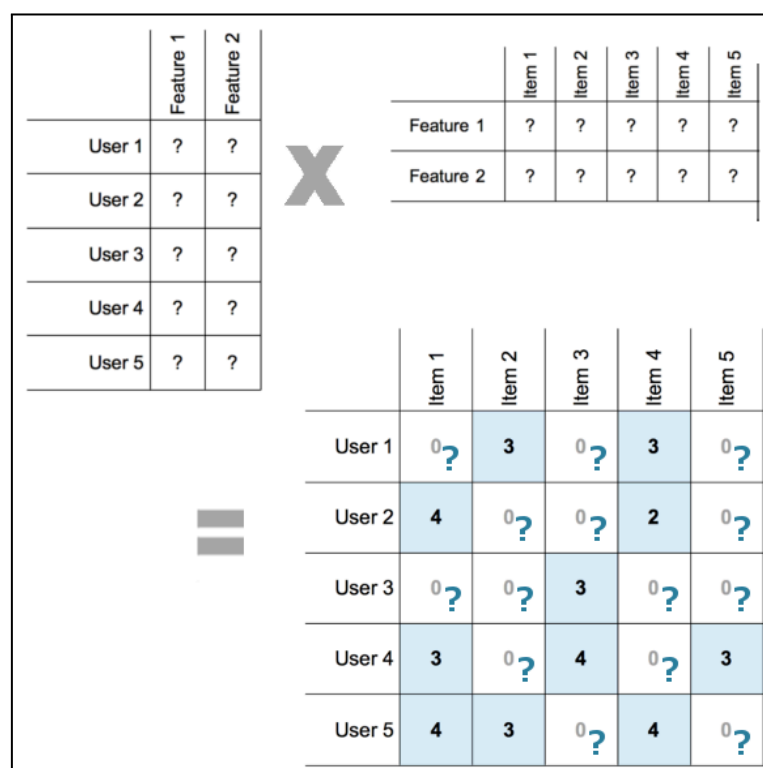
### a. Keras :

Keras is an open-source library that provides a Python interface for artificial neural networks. Keras was first independent software, then integrated into the TensorFlow library, and later supported more. [\[63\]](#)

### b. Building the model using Embedding layers :

An embedding is a mapping from discrete objects, such as words or courses in our case, to a vector of continuous values. These are used to find similarities between discrete objects.

The concept behind matrix factorization models is that the preferences of a user can be determined by a small number of hidden factors. And these are called Embeddings.



**Fig 43.** How does the embedding layers work

As you can see in the image, there are 2 features both for the user and the items. These are the latent factors or the hidden factors. These factors have a value for the corresponding user and determine to what extent that user likes the feature.

For Eg. the features for the user could be ->

- How much does he like python courses?

- whether he likes python courses?

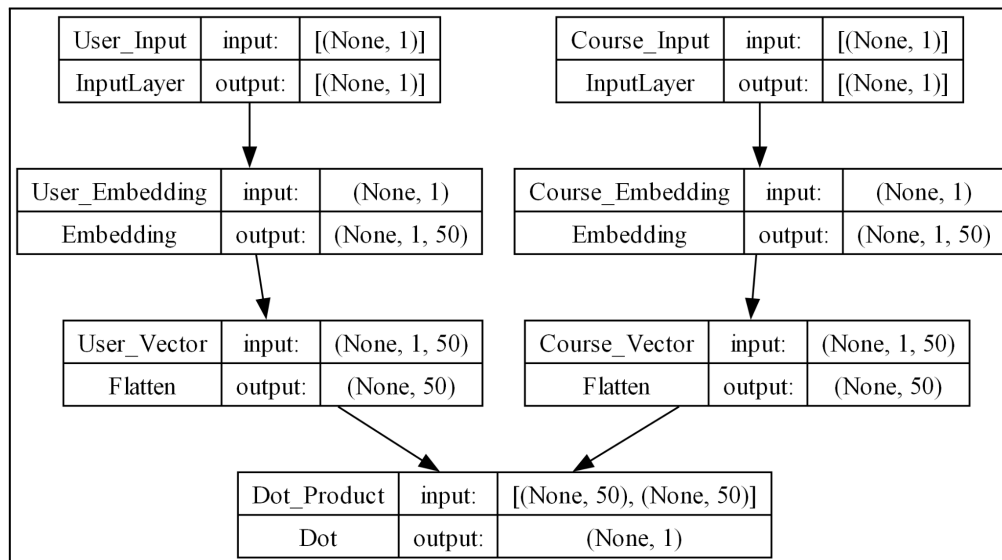
And the features for the courses could be ->

- To what scale is it a programming course?
- Whether the course was recently released?

And finally we take dot product which gives us the user's rating for the course,

**Estimated Rating** = [ (How much he likes python courses ?) x (To what scale is it a python course) ] + [ (Whether he likes python courses?) x (Whether the course is recently released) ]

### c. Visualizing Model Architecture 1 :



**Fig 44.** Visualizing model 1 architecture

#### d. Model Config :

| Layer (type)                 | Output Shape  | Param # | Connected to                                    |
|------------------------------|---------------|---------|-------------------------------------------------|
| User_Input (InputLayer)      | [(None, 1)]   | 0       | []                                              |
| Course_Input (InputLayer)    | [(None, 1)]   | 0       | []                                              |
| User_Embedding (Embedding)   | (None, 1, 50) | 30500   | ['User_Input[0][0]']                            |
| Course_Embedding (Embedding) | (None, 1, 50) | 486200  | ['Course_Input[0][0]']                          |
| User_Vector (Flatten)        | (None, 50)    | 0       | ['User_Embedding[0][0]']                        |
| Course_Vector (Flatten)      | (None, 50)    | 0       | ['Course_Embedding[0][0]']                      |
| Dot_Product (Dot)            | (None, 1)     | 0       | ['User_Vector[0][0]',<br>'Course_Vector[0][0]'] |
| =====                        |               |         |                                                 |
| Total params: 516,700        |               |         |                                                 |
| Trainable params: 516,700    |               |         |                                                 |
| Non-trainable params: 0      |               |         |                                                 |

**Table 05.** Model 1 Config

#### e. Optimizer :

**Adam :** Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments.

Arguments

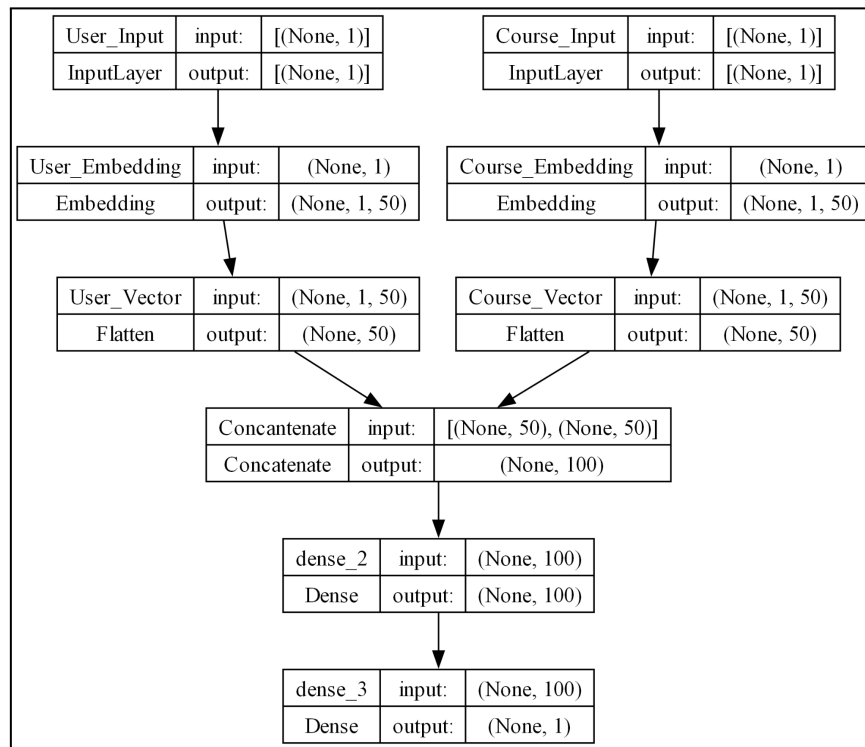
- **learning\_rate:** A float, a `keras.optimizers.schedules.LearningRateSchedule` instance, or a callable that takes no arguments and returns the actual value to use. The learning rate. Defaults to `0.001`.
- **beta\_1:** A float value or a constant float tensor, or a callable that takes no arguments and returns the actual value to use. The exponential decay rate for the 1st moment estimates. Defaults to `0.9`.
- **beta\_2:** A float value or a constant float tensor, or a callable that takes no arguments and returns the actual value to use. The exponential decay rate for the 2nd moment estimates. Defaults to `0.999`.

- 
- **epsilon:** A small constant for numerical stability. This epsilon is "epsilon hat" in the Kingma and Ba paper (in the formula just before Section 2.1), not the epsilon in Algorithm 1 of the paper. Defaults to `1e-7`.
  - **amsgrad:** Boolean. Whether to apply the AMSGrad variant of this algorithm from the paper "On the Convergence of Adam and beyond". Defaults to `False`.
  - **name:** String. The name to use for momentum accumulator weights created by the optimizer.
  - **weight\_decay:** Float. If set, weight decay is applied.
  - **clipnorm:** Float. If set, the gradient of each weight is individually clipped so that its norm is no higher than this value.
  - **clipvalue:** Float. If set, the gradient of each weight is clipped to be no higher than this value.
  - **global\_clipnorm:** Float. If set, the gradient of all weights is clipped so that their global norm is no higher than this value.
  - **use\_ema:** Boolean, defaults to `False`. If `True`, exponential moving average (EMA) is applied. EMA consists of computing an exponential moving average of the weights of the model (as the weight values change after each training batch), and periodically overwriting the weights with their moving average.
  - **ema\_momentum:** Float, defaults to 0.99. Only used if `use_ema=True`. This is the momentum to use when computing the EMA of the model's weights: 
$$\text{new\_average} = \text{ema\_momentum} * \text{old\_average} + (1 - \text{ema\_momentum}) * \text{current\_variable\_value}.$$
  - **ema\_overwrite\_frequency:** Int or None, defaults to None. Only used if `use_ema=True`. Every `ema_overwrite_frequency` steps of iterations, we overwrite the model variable by its moving average. If None, the optimizer does not overwrite model variables in the middle of training, and you need to explicitly overwrite the variables at the end of training by calling `optimizer.finalize_variable_values()` (which updates the model variables in-place). When using the built-in `fit()` training loop, this happens automatically after the last epoch, and you don't need to do anything.
  - **loss\_scale\_factor:** Float or `None`. If a float, the scale factor will be multiplied by the loss before computing gradients, and the inverse of the scale factor will be multiplied by the gradients before updating variables. Useful for preventing underflow during mixed precision training. Alternately, `keras.optimizers.LossScaleOptimizer` will automatically set a loss scale factor.

- **gradient\_accumulation\_steps:** Int or **None**. If an int, model & optimizer variables will not be updated at every step; instead they will be updated every **gradient\_accumulation\_steps** steps, using the average value of the gradients since the last update. This is known as "gradient accumulation". This can be useful when your batch size is very small, in order to reduce gradient noise at each update step. [\[64\]](#)

#### f. Model Architecture 2 :

In this architecture we will be concatenating the latent factors and use Dense layers to calculate the predicted ratings.



**Fig 45.** Model 2 architecture

#### g. Model Config 2 :

| Layer (type)                 | Output Shape  | Param # | Connected to                                    |
|------------------------------|---------------|---------|-------------------------------------------------|
| User_Input (InputLayer)      | [(None, 1)]   | 0       | []                                              |
| Course_Input (InputLayer)    | [(None, 1)]   | 0       | []                                              |
| User_Embedding (Embedding)   | (None, 1, 50) | 30500   | ['User_Input[0][0]']                            |
| Course_Embedding (Embedding) | (None, 1, 50) | 486200  | ['Course_Input[0][0]']                          |
| User_Vector (Flatten)        | (None, 50)    | 0       | ['User_Embedding[0][0]']                        |
| Course_Vector (Flatten)      | (None, 50)    | 0       | ['Course_Embedding[0][0]']                      |
| Concatenate (Concatenate)    | (None, 100)   | 0       | ['User_Vector[0][0]',<br>'Course_Vector[0][0]'] |
| dense_2 (Dense)              | (None, 100)   | 10100   | ['Concatenate[0][0]']                           |
| dense_3 (Dense)              | (None, 1)     | 101     | ['dense_2[0][0]']                               |
| Total params: 526,901        |               |         |                                                 |
| Trainable params: 526,901    |               |         |                                                 |
| Non-trainable params: 0      |               |         |                                                 |

**Table 06.** Model 2 Config

#### **h. Loss Function used :**

mean squared error (MSE), the average squared difference between the value observed in a statistical study and the values predicted from a model. When comparing observations with predicted values, it is necessary to square the differences as some data values will be greater than the prediction (and so their differences will be positive) and others will be less (and so their differences will be negative). Given that observations are as likely to be greater than the predicted values as they are to be less, the differences would add to zero. Squaring these differences eliminates this situation.

The formula for the mean squared error is  $MSE = \sum (y_i - p_i)^2 / n$ , where  $y_i$  is the  $i$ th observed value,  $p_i$  is the corresponding predicted value for  $y_i$ , and  $n$  is the number of observations. The  $\sum$  indicates that a summation is performed over all values of  $i$ . [\[65\]](#)

#### **i. Model Evaluation :**

We used the RMSE, MSE, Precision@k, Recall@k, f1@k, as metrics for the evaluation phase of our model.

**RMSE :** The Root Mean Squared Error (RMSE) is one of the two main performance indicators for a regression model. It measures the average difference between values predicted by a model and

---

the actual values. It provides an estimation of how well the model is able to predict the target value (accuracy). [\[66\]](#)

$$\text{RMSE} = \sqrt{[(\sum (P_i - O_i)^2) / n]}$$

**Precision@K** : Precision at K is the ratio of correctly identified relevant items within the total recommended items inside the K-long list. Simply put, it shows how many recommended or retrieved items are genuinely relevant.

Here is a formula :

$$\text{Precision@k} = \text{number of relevant items in k} / \text{total number items in k} \quad \text{[68]}$$

**Recall@k** : Recall at K measures the proportion of correctly identified relevant items in the top K recommendations out of the total number of relevant items in the dataset. In simpler terms, it indicates how many of the relevant items you could successfully find.

Here is a formula:

$$\text{Recall@k} = \text{Number of relevant items in k} / \text{total number of relevant items} \quad \text{[68]}$$

**f1@k** : is a metric used to evaluate the performance of recommendation systems, combining both precision and recall into a single measure. It is particularly useful in scenarios where both false positives (incorrect recommendations) and false negatives (missed relevant items) are important to consider.

Here is a formula :

$$f1@k = 2 * [(Precision@k * Recall@k) / (Precision@k + Recall@k)]$$

---

## 2.3.2 AI-DRIVEN ASSIGNMENT GENERATION

An AI-driven assignment generation system for use in an AI-enhanced e-learning platform will deploy an advanced NLP model for the auto-generation of educational assignments from lecture transcriptions. The system is based on the fact that it will save instructors time to generate relevant and contextually correct questions and assignments based on the content of the lecture. The implementation details about the workflow, component, and benefits of this feature are available in this section.

### 1. Workflow of AI-Driven Assignment Generation :

---

**a. Data Collection :**

**Transcribe video lectures :** this is done using the YouTubeTranscriptApi to transcribe spoken content from the lecture videos to text. This platform stores the transcriptions in its database.

**b. Input Processing :**

**Preprocessing :** The transcripts are cleaned up and formatted to eliminate any irrelevant content or noise. This step is needed to make sure that the text input is fit for the AI model to give expected outputs of quality.

**Segmentation :** Finally, the cleaned transcription is broken down into units of meaningful content, in this case paragraphs or sections, to form the basis of auto-generated assignments.

**c. Assignment Generation :**

**Model Input :** The Segmented transcription text is input to the Claude-3-opus-20240229, which is a sophisticated model in NLP with a human-like ability of producing texts.

**Text Generation :** the model answers questions, composes summaries, and produces other types of assignments on the basis of content from the transcription.

**d. Customization and Refinement :**

**Instructor Review :** All generated assignments are reviewed by educators against set learning objectives, as well as an assurance of quality.

**Adjustments :** Enabling instructors to modify the assignments to put in more context questions more applicable to their teaching aspects.

**Creation :** The assignments done are created on Google Forms at the end.

**2. Claude NLP models :**

Claude is a family of large language models developed by Anthropic and designed to revolutionize the way you interact with AI. Claude excels at a wide variety of tasks involving language, reasoning, analysis, coding, and more. Our models are highly capable, easy to use, and can be customized to suit your needs. [\[35\]](#)

- **Used NLP Model :**

In March 2024, Anthropic announced the Claude 3 model family, setting new industry benchmarks across a wide range of cognitive tasks. The family

---

includes three state-of-the-art models in ascending order of capability: Claude 3 Haiku, Claude 3 Sonnet, and Claude 3 Opus. [\[36\]](#)

All Claude 3 models demonstrate increased capabilities in analysis, forecasting, nuanced content creation, code generation, and conversing in non-English languages like Spanish, Japanese, and French. They also possess sophisticated vision capabilities, allowing them to process and analyze visual formats such as photos, charts, graphs, and technical diagrams. [\[36\]](#)

- **What Training Methods did Anthropic use to train Claude ?**

Anthropic employs a unique approach called Constitutional AI to train its AI systems, particularly language models like Claude, to be helpful, honest, and harmless without relying extensively on human feedback. The method involves two phases: supervised learning and reinforcement learning.

In the supervised learning phase, the model generates responses to prompts, self-critiques these responses based on a set of guiding principles (a “constitution”), and revises the responses. The model is then fine-tuned on these revised responses.

During the reinforcement learning from AI feedback (RLAIF) phase, responses are generated and compared according to their compliance with the constitution. This dataset of AI feedback is used to train a preference model that evaluates responses based on how well they satisfy the constitution. Claude is then fine-tuned to align with this preference model. [\[36\]](#)

### **3. Benefits of Ai-Driven Assignment Generator :**

**Efficiency** : It saves educators time and effort by automating the creation of assignments.

**Consistency** : Ensures that the assignments are consistently aligned with the lecture content.

**Scalability** : Generates a high-volume of assignments in very little time while meeting all the diversified course requirements.

**Adaptability** : It is possible to create assignments for any subject and level of complacency with the help of such a system.

---

#### **4. Summary of Ai-Driven Assignment Generator :**

The integration of the Claude-3-opus-20240229 model into the transcriptions from the YouTubeTranscriptApi makes this an immense tool to automate the development of educational assignments. Raising the efficiency and uniformity level in the production process of assignments, flexibility, and guarantee of adjustability to different educational environments on a massive scale are the advantages of an implemented system. Leveraging such cutting-edge AI technologies, the platform is designed to facilitate high-quality, contextually relevant assignments that closely map to the instructional content, leading to better educational outcomes and lessening the administrative burden on educators.

---

---

## **CHAPTER 3**

---

# **IMPLEMENTATION & RESULTS**

---

This chapter primarily talks about the technical implementation of an AI-based eLearning platform. This chapter will be presenting results from various testing phases, which are intended to demonstrate how the objectives of the project have been met and the implications of the findings

The Implementation phase will contain : The environment for development is dealt with in, implementation of codes, deployment, the challenges faced, results, the results of usability testing, performance testing security testing, functional testing.

## 3.1 IMPLEMENTATION

### 3.1.1 DEVELOPMENT ENVIRONMENT

#### 1. Hardware :

**Processor :** Intel(R) Core(TM) i5-8300H CPU @ 2.30GHZ, 4.00GHZ

**RAM :** 16GB 2666MHZ DDR4

**Graphic Card :** Geforce Nvidia GTX 1050 Ti Mobile 4GB GDDR5, Intel® UHD Graphics 630.

#### 2. Software :

##### a. Windows 10 Professional 22H2

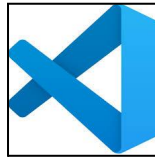


**Fig 18.** Windows Logo

Windows 10 is a major release of the Windows NT operating system developed by Microsoft. Microsoft described Windows 10 as an "operating system as a service" that would receive ongoing updates to its features and functionality, augmented with the ability for enterprise environments to receive non-critical updates at a slower pace or use long-term support milestones that will only receive critical updates, such as security patches, over their five-year lifespan of mainstream support. It was released in July 2015. [\[46\]](#)

---

b. **Visual Studio Code**



**Fig 19.** Vscode Logo

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor developed by Microsoft for Windows, Linux, macOS and web browsers. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded version control with Git. [\[47\]](#)

c. **Web Server : XAMPP**



**Fig 19.** Xampp Logo

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. [\[48\]](#)

d. **NPM**



**Fig 20.** npm Logo

npm is a package manager for the JavaScript programming language maintained by npm, Inc., a subsidiary of GitHub. npm is the default package manager for the JavaScript runtime environment Node.js and is included as a recommended feature in the Node.js installer. [\[49\]](#)

e. **Spark**



**Fig 21.** Spark Logo

Apache Spark is an open-source unified analytics engine for large-scale data processing. Spark provides an interface for programming clusters with implicit data parallelism and fault tolerance. [\[50\]](#)

---

### 3.1.2 OVERVIEW OF CODEBASE STRUCTURE

#### 1. Backend Development :

##### a. CodeIgniter 4 framework :

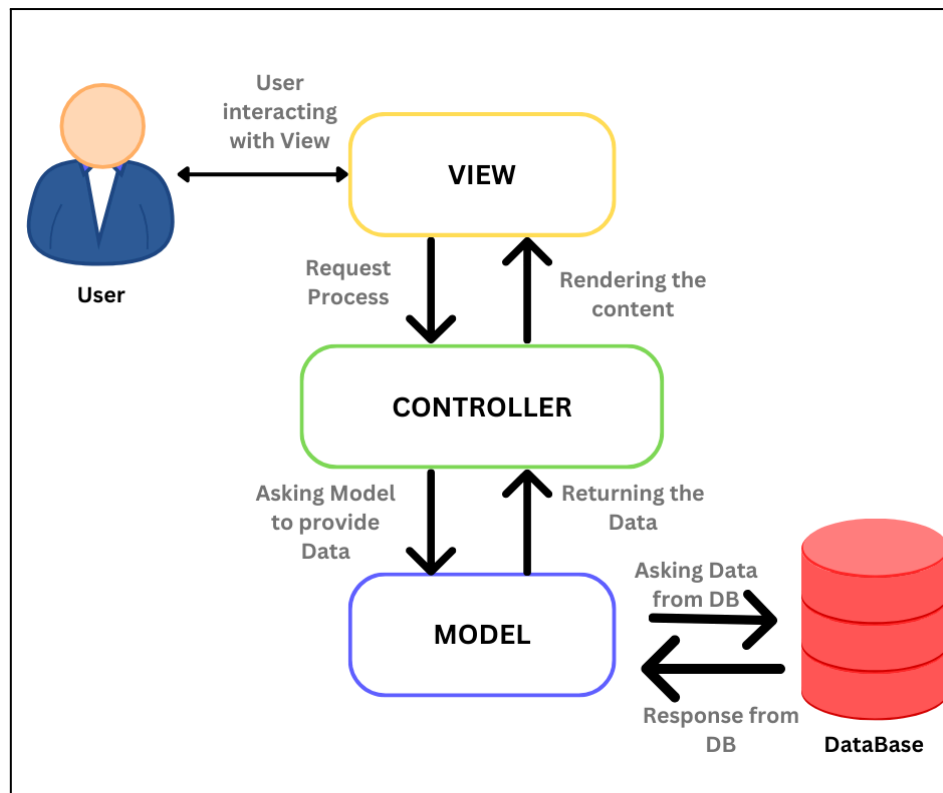


**Fig 22.** CodeIgniter Logo

CodeIgniter is an application development framework, which can be used to develop websites, using PHP and MVC architecture. It is an Open Source framework. It has a very rich set of functionality, which will increase the speed of website development work.

If you know PHP well, then CodeIgniter will make your task easier. It has a very rich set of libraries and helpers. By using CodeIgniter, you will save a lot of time, if you are developing a website from scratch. Not only that, a website built in CodeIgniter is secure too, as it has the ability to prevent various attacks that take place through websites. [\[51\]](#)

- **MVC architecture (Used by CodeIgniter) :**



**Fig 23.** a Diagram explain the MVC Architecture

The **Model-View-Controller (MVC)** framework is an architectural/design pattern that separates an application into three main logical components **Model**, **View**, and **Controller**. Each architectural component is built to handle specific development aspects of an application. It isolates the business logic and presentation layer from each other. It was traditionally used for desktop **graphical user interfaces (GUIs)**. Nowadays, MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects. It is also used for designing mobile apps.

MVC was created by **Trygve Reenskaug**. The main goal of this design pattern was to solve the problem of users controlling a large and complex data set by splitting a large application into specific sections that all have their own purpose. [\[52\]](#)

### **Controller :**

The controller is the component that enables the interconnection between the views and the model so it acts as an intermediary. The controller doesn't have to worry about handling

---

data logic, it just tells the model what to do. It processes all the business logic and incoming requests, manipulates data using the **Model** component, and interacts with the **View** to render the final output.

**View :**

The **View** component is used for all the UI logic of the application. It generates a user interface for the user. Views are created by the data which is collected by the model component but these data aren't taken directly but through the controller. It only interacts with the controller.

**Model :**

The **Model** component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. It can add or retrieve data from the database. It responds to the controller's request because the controller can't interact with the database by itself. The model interacts with the database and gives the required data back to the controller. [\[52\]](#)

**b. Python**



**Fig 24.** Python Logo

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming. [\[53\]](#)

**c. PHP**



*Fig 25.* PHP Logo

PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and released in 1995. The PHP reference implementation is now produced by the PHP Group. [\[54\]](#)

#### d. Flask



*Fig 33.* Flask Logo

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. [\[67\]](#)

## 2. Frontend Development :

#### a. HTML5



*Fig 26.* HTML5 Logo

HTML5 (Hypertext Markup Language 5) is **a markup language used for structuring and presenting hypertext documents on the World Wide Web**. It was the fifth and final major HTML version that is now a retired World Wide Web Consortium (W3C) recommendation. The current specification is known as the HTML Living Standard. [\[55\]](#)

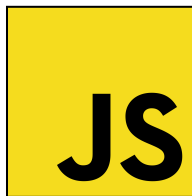
#### b. CSS3



**Fig 27.** CSS3 Logo

Cascading Style Sheets is a style sheet language used for specifying the presentation and styling of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. [\[56\]](#)

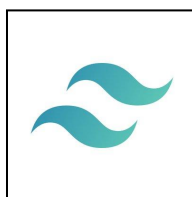
#### **c. JavaScript**



**Fig 28.** Javascript Logo

JavaScript, often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS. 99% of websites use JavaScript on the client side for webpage behavior. Web browsers have a dedicated JavaScript engine that executes the client code. [\[57\]](#)

#### **d. Tailwind Css**



**Fig 29.** TailwindCss Logo

Tailwind CSS is an open-source CSS framework. The main feature of this library is that, unlike other CSS frameworks like Bootstrap, it does not provide a series of predefined classes for elements such as buttons or tables. [\[58\]](#)

### **3. Database Integration :**

---

**a. MySQL :**



**Fig 30.** MySQL Logo

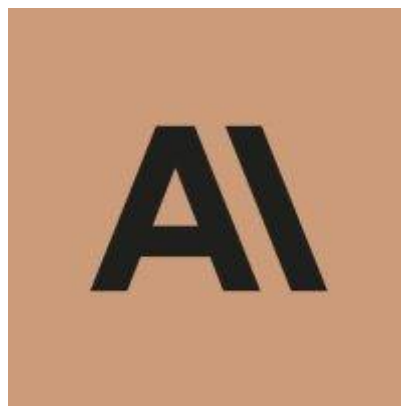
MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the acronym for Structured Query Language. [\[59\]](#)

**4. AI Api Modules :**

**a. YoutubeTranscriptAPI :**

This is a python API which allows you to get the transcript/subtitles for a given YouTube video. It also works for automatically generated subtitles, supports translating subtitles and it does not require a headless browser, like other selenium based solutions do! [\[60\]](#)

**b. Anthropic Claude Api :**



**Fig 31.** Anthropic Claude Logo

The Claude AI API represents a gateway to integrating Anthropic's advanced AI functionalities into your own applications or digital solutions. Whether you're looking to enhance a chatbot, develop intelligent data analysis tools, or create personalized content recommendations, the

---

Claude AI API provides the building blocks needed to bring these ideas to life with unprecedented ease and efficiency. [\[61\]](#)

**c. VideoJS Api for Annotation :**



**Fig 32.** VideoJs Logo

*Video.js* is a web video player built from the ground up for an HTML5 world. It supports HTML5 video and modern streaming formats, as well as YouTube and Vimeo. [\[62\]](#)

---

### 3.1.3 KEYCODE SNIPPETS

**1. Course Recommendation using Surprise :**

**Code Snippet :**

```
from surprise import SVD, Dataset, Reader

# Load the dataset from a DataFrame
data = Dataset.load_from_df(course_data[['user_id', 'course_id',
'rating']], Reader(rating_scale=(1, 5)))

# Initialize the SVD algorithm
algo = SVD()

# Train the model on the full dataset
trainset = data.build_full_trainset()
algo.fit(trainset)

# Predict the rating for a specific user and course
prediction = algo.predict(user_id, course_id)
print(f"Predicted rating: {prediction.est}")
```

**Explanation :**

- **Dataset Loading :** The `Dataset` class from Surprise is used to load the data from a DataFrame containing user IDs, course IDs, and ratings.
- **Algorithm Initialization :** The SVD (Singular Value Decomposition) algorithm is initialized.

- 
- **Model Training** : The model is trained on the full dataset using `fit()`.
  - **Prediction** : The model predicts the rating for a specific user and course using `predict()`, and the predicted rating is printed.

## 2. Assignment Generation Using Claude Api :

### Code Snippet :

```
def generate_assignments(transcriptions):
    client = anthropic.Anthropic(
        api_key="#",
    )

    message = client.messages.create(
        model="claude-3-opus-20240229",
        max_tokens=4000,
        temperature=0.7,
        system="You are an educational assistant that generates assignments in form of quizzes for an elearning website from given transcriptions.",
        messages=[
            {
                "role": "user",
                "content": [
                    {
                        "type": "text",
                        "text": f"Generate assignments based on the following transcriptions with a max of 10 questions:\n{transcriptions}"
                    }
                ]
            }
        ]
    )
    generated_text = message.content[0].text # Access the text attribute of the first TextBlock
    return generated_text
if __name__ == "__main__":
    # Ensure you have set your API key as an environment variable
    transcriptions = sys.argv[1]
    assignments = generate_assignments(transcriptions)
    print(assignments)
```

---

### Explanation :

- **Model Initialization** : The Claude model `Claude-3-opus-20240229` is initialized.
- **Input Transcription** : An example transcription text from a lecture is provided.
- **Assignment Generation** : The model generates assignments based on the provided transcription.
- **Output** : The generated assignments are printed.

### 3. Video Lesson Transcription Using YoutubeTranscriptApi :

#### Code Snippet :

```
from youtube_transcript_api import YouTubeTranscriptApi

# Get the transcript for a specific YouTube video ID
video_id = 'example_video_id'
transcript = YouTubeTranscriptApi.get_transcript(video_id)

# Print the transcript
for entry in transcript:
    print(f"{entry['start']}: {entry['text']}")
```

### Explanation :

**Fetch Transcript** : The `YouTubeTranscriptApi` fetches the transcript for a given YouTube video ID.

**Output** : The transcript entries are printed with their start times and text.

### 4. Video Annotation Storage and Retrieval :

#### Code Snippet :

```
def store_annotation(video_id, timestamp, content):
    annotation_data = {
        'video_id': video_id,
        'timestamp': timestamp,
        'content': content
    }
    db.annotations.insert_one(annotation_data)
    print(f"Annotation stored at {timestamp} seconds for video {video_id}")
    video_id = 'example_video_id'
```

```
timestamp = 120.5
content = 'This section explains the uncertainty principle.'
store_annotation(video_id, timestamp, content)
```

## 5. Integration Deep Learning Model with Codeigniter :

```
from flask import Flask, request, jsonify
import pickle
from surprise import Dataset, Reader

app = Flask(__name__)

# Load the trained model
with open('svd_model.pkl', 'rb') as model_file:
    algo = pickle.load(model_file)

# Function to get course recommendations for a specific user
def get_recommendations(user_id, num_recommendations=5):
    # Load the data for getting the list of all courses
    file_path = 'ratings.csv'
    reader = Reader(line_format='user item rating', sep=',',
rating_scale=(1, 5), skip_header=True)
    data = Dataset.load_from_file(file_path, reader=reader)

    # Get the list of all course IDs
    trainset = data.build_full_trainset()
    all_course_ids = trainset.all_items()
    all_course_ids = [trainset.to_raw_iid(course_id) for course_id in
all_course_ids]

    # Predict ratings for all courses for the given user
    predicted_ratings = []
    for course_id in all_course_ids:
        prediction = algo.predict(user_id, course_id)
        predicted_ratings.append((course_id, prediction.est))

    # Sort the courses by the predicted ratings in descending order
    predicted_ratings.sort(key=lambda x: x[1], reverse=True)

    # Return the top N recommended courses
    return predicted_ratings[:num_recommendations]

@app.route('/recommend', methods=['GET'])
def recommend():
    user_id = int(request.args.get('user_id'))
```

---

```
recommendations = get_recommendations(user_id)
return jsonify(recommendations)

if __name__ == '__main__':
    app.run(debug=True)
```

---

## 3.2 RESULTS AND ANALYSIS

### 3.2.1 DEEP LEARNING MODEL :

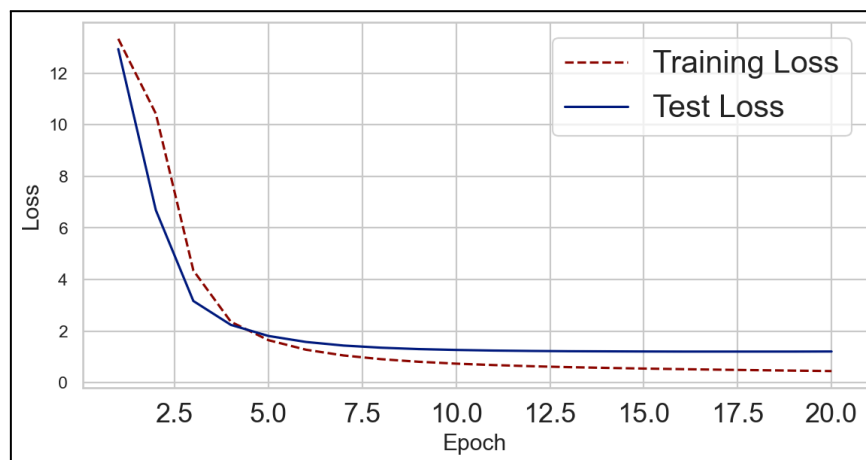
#### Results

In this section we will discuss the results of the two DLRM models developed for the recommendation phase of our project, we will give the metrics of both models, such RMSE and MAE and MSE for Precision, Precision@k, F1@k, Recall@k for ranking.

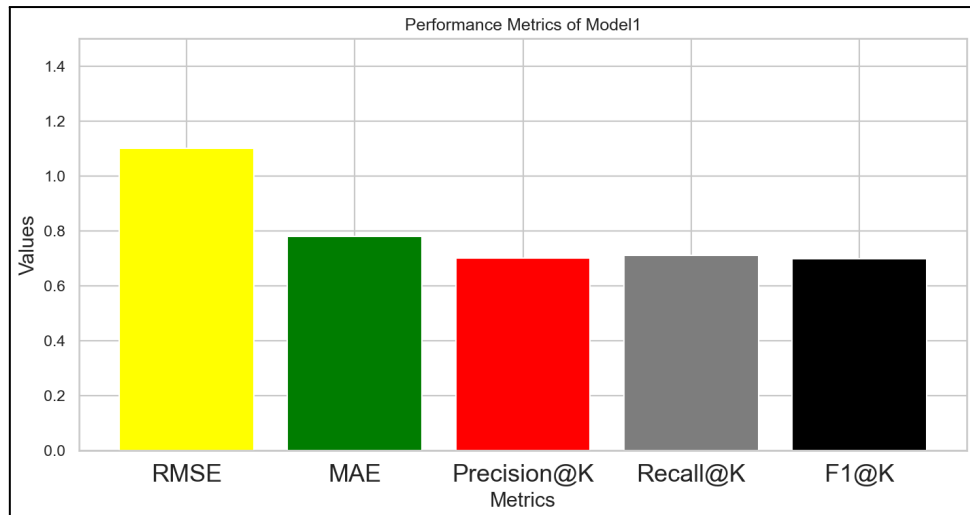
#### Model1:

##### Embedding-610-50-User\_Vector-Embedding-9724-50-Course\_Vector-Dot\_Product

The proposed recommendation model employs a Deep Learning Recommendation Model (DLRM) architecture to predict user-course interactions. The model consists of two parallel embedding layers: one for users and one for courses. Each embedding layer transforms the user and course IDs into dense vectors of 50 latent factors. These vectors are then flattened and combined using a dot product operation, capturing the interaction between the user and the course. This architecture effectively learns latent features from the user-course data, allowing it to generate personalized course recommendations.



**Fig 34.** Visualization of Loss history of model 1



**Fig 35.** Evaluation Results of Model 1

The recommendation model was evaluated using several key metrics, which provide insights into its performance. Here are the specific results:

- **RMSE (Root Mean Squared Error): 1.1004**
- **MAE (Mean Absolute Error): 0.7791**
- **Precision@K: 0.7014**
- **Recall@K: 0.7109**
- **F1@K: 0.6982**

## Analysis

### 1. RMSE and MAE:

- The RMSE of 1.1004 indicates that, on average, the predicted ratings deviate from the actual ratings by about 1.1 units. While this might seem substantial, it needs to be interpreted in the context of the rating scale used (typically 0.5 to 5.0).
- The MAE of 0.7791 shows that the average absolute error between the predicted and actual ratings is approximately 0.78. This metric is less sensitive to outliers than RMSE and provides a more straightforward interpretation of the average prediction error.

### 2. Precision@K:

- A Precision@K of 0.7014 means that 70.14% of the top-K recommended items are relevant. This indicates a good ability of the model to recommend relevant courses.

### 3. Recall@K:

- The Recall@K of 0.7109 signifies that the model successfully retrieves 71.09% of the relevant items in the top-K recommendations. This high recall value suggests that the model captures a significant portion of the relevant items.

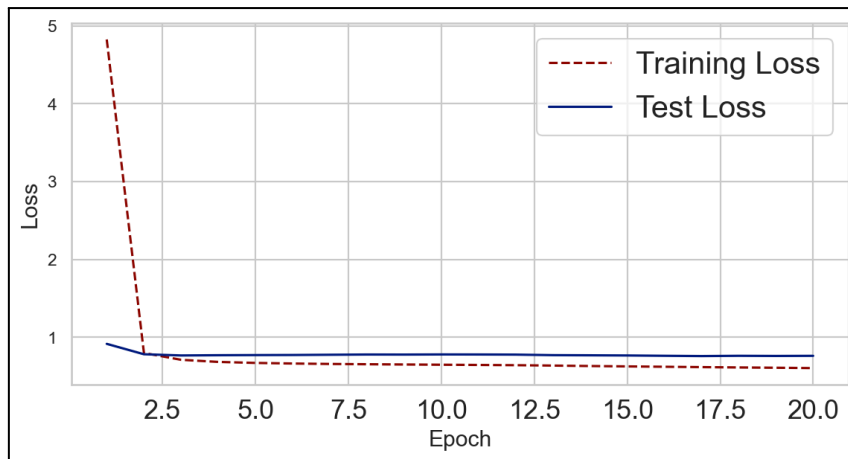
### 4. F1@K:

- The F1@K score of 0.6982 is the harmonic mean of Precision@K and Recall@K, providing a balance between the two. A value close to 0.70 indicates a good trade-off between precision and recall.

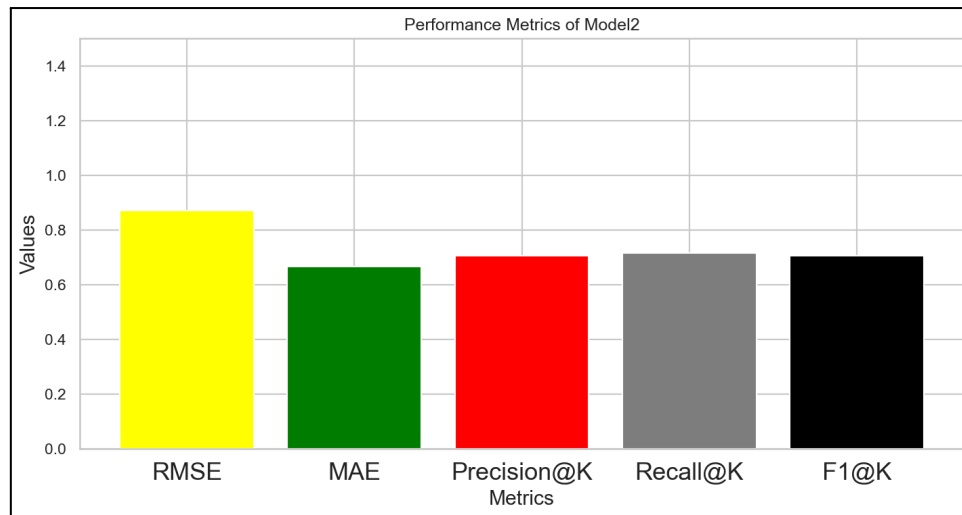
## Model 2:

### Embedding-610-50-User\_Vector-Embedding-9724-50-Course\_Vector-Concatenate-Dense-100-Relu-Dense-1

The second recommendation model employs a Deep Learning Recommendation Model (DLRM) architecture designed to predict user-course interactions with enhanced feature concatenation. It consists of embedding layers for users and courses, each producing 50 latent factors, which are then flattened into dense vectors. These vectors are concatenated to form a combined feature vector, capturing the interaction between users and courses. The concatenated vector is passed through a dense layer with 100 units and a ReLU activation function, followed by a final dense layer with 1 unit to produce the predicted rating. This architecture, improves prediction accuracy and recommendation relevance, as indicated by lower RMSE and MAE values and higher Precision@K, Recall@K, and F1@K scores compared to the previous model.



**Fig 36.** Visualization of Loss history of model 2



**Fig 37.** Evaluation Results of Model 2

The second recommendation model was evaluated using several key metrics to assess its performance. Here are the specific results:

- **RMSE (Root Mean Squared Error): 0.8718**
- **MAE (Mean Absolute Error): 0.6668**
- **Precision@K: 0.7053**
- **Recall@K: 0.7146**
- **F1@K: 0.7048**

### Analysis

#### 1. RMSE and MAE:

- The RMSE of 0.8718 is an improvement over the first model, indicating that the predicted ratings are, on average, closer to the actual ratings compared to the first model.
- The MAE of 0.6668 also shows an improvement, indicating more accurate predictions with a smaller average absolute error compared to the first model.

#### 2. Precision@K:

- A Precision@K of 0.7053 means that 70.53% of the top-K recommended items are relevant. This is a slight improvement over the first model, suggesting a better ability to recommend relevant courses.

#### 3. Recall@K:

- 
- The Recall@K of 0.7146 signifies that the model successfully retrieves 71.46% of the relevant items in the top-K recommendations. This value is slightly higher than the first model, indicating better coverage of relevant items.

4. **F1@K:**

- The F1@K score of 0.7048 is an improvement and indicates a good balance between precision and recall, suggesting the model effectively handles both aspects of recommendation relevance.

## **Comparative Analysis**

The performance of the two recommendation models was evaluated using several key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Precision@K, Recall@K, and F1@K. Below are the results for both models:

**Model 1:**

- **RMSE:** 1.1004
- **MAE:** 0.7791
- **Precision@K:** 0.7014
- **Recall@K:** 0.7109
- **F1@K:** 0.6982

**Model 2:**

- **RMSE:** 0.8718
- **MAE:** 0.6668
- **Precision@K:** 0.7053
- **Recall@K:** 0.7146
- **F1@K:** 0.7048

**Accuracy Metrics:**

- **RMSE:** Model 2 has a lower RMSE (0.8718) compared to Model 1 (1.1004), indicating that the predictions of Model 2 are closer to the actual ratings.
- **MAE:** Model 2 also has a lower MAE (0.6668) compared to Model 1 (0.7791), suggesting that the average magnitude of errors is smaller in Model 2.

**Precision@K:**

- 
- Model 2 has a slightly higher Precision@K (0.7053) than Model 1 (0.7014). This indicates that Model 2 is slightly better at recommending relevant items within the top K recommendations.

**Recall@K:**

- Model 2 has a higher Recall@K (0.7146) compared to Model 1 (0.7109), meaning Model 2 captures a greater proportion of relevant items in its top K recommendations.

**F1@K:**

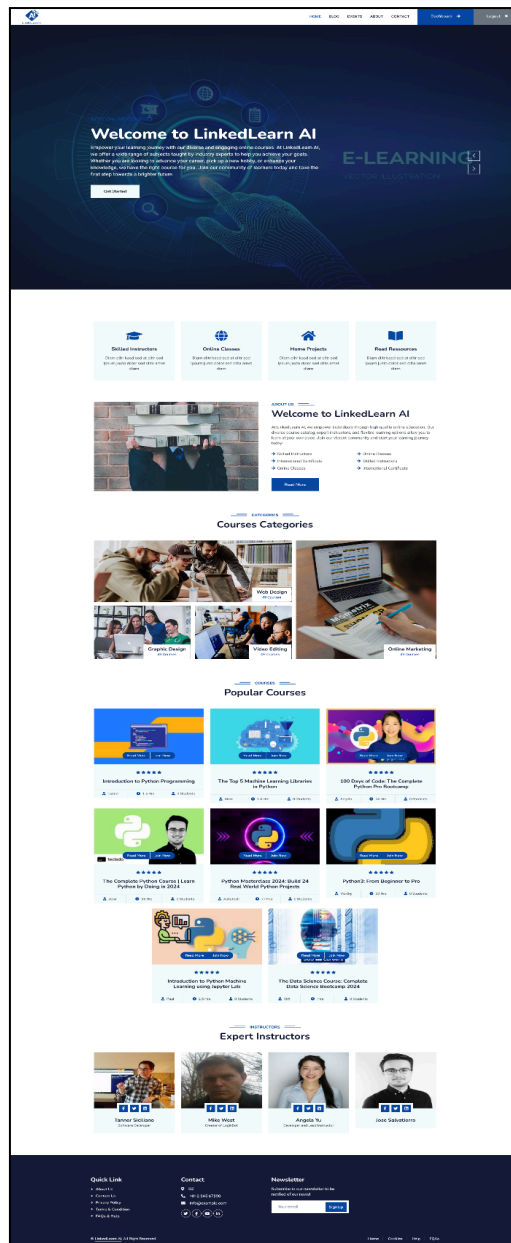
- The F1@K score of Model 2 (0.7048) is higher than that of Model 1 (0.6982), indicating a better balance between precision and recall in Model 2.

Based on the comparative analysis, Model 2 outperforms Model 1 across all evaluated metrics. The lower RMSE and MAE values indicate more accurate predictions, while the higher Precision@K, Recall@K, and F1@K scores suggest better relevance and coverage of the recommendations. Therefore, **Model 2 is the better model** for providing personalized course recommendations, as it delivers more accurate and relevant suggestions to users.

---

## 3.3 PLATFORM REPRESENTATION

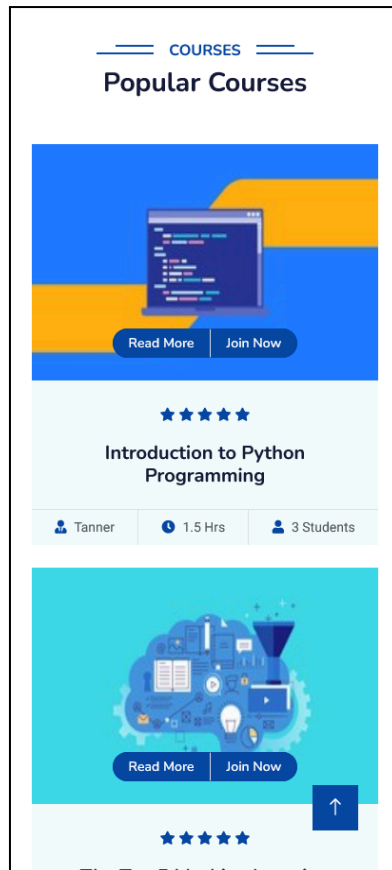
**Homepage :**



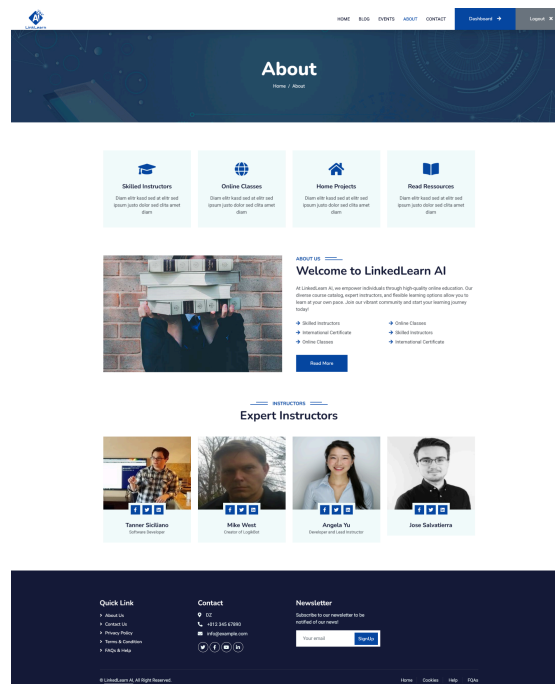
**Fig 46.** HomePage Desktop View

The homepage contains sections such as navbar, header, services, about, categories, popular courses, expert instructors, and footer.

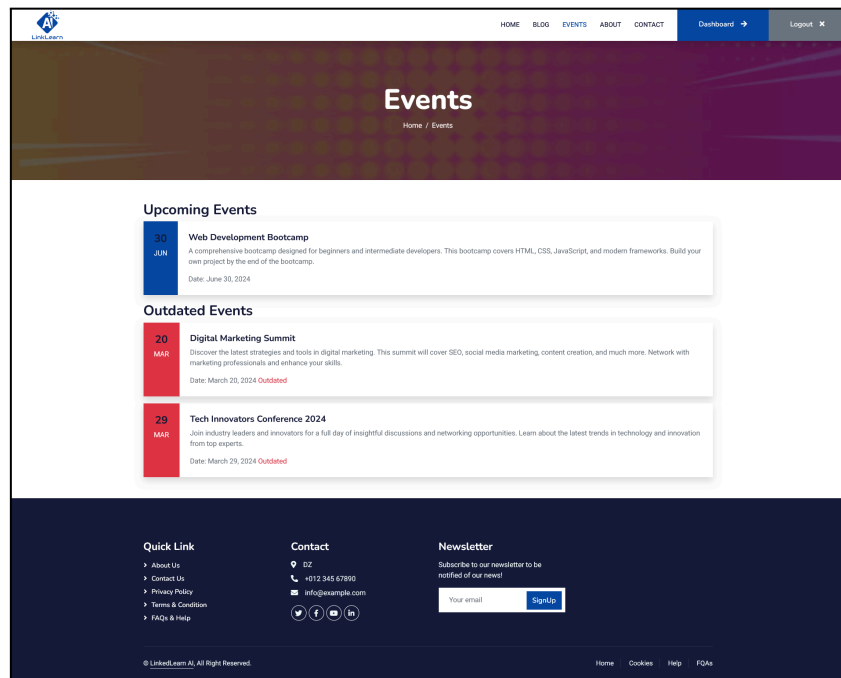
The screenshot down will represent a mobile view of the homepage :



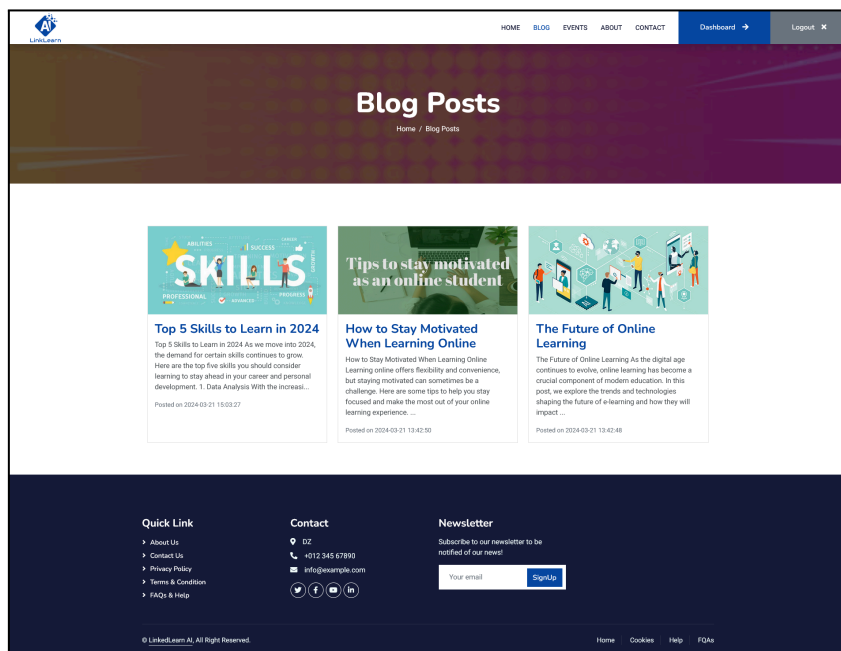
**Fig 47.** HomePage Mobile View



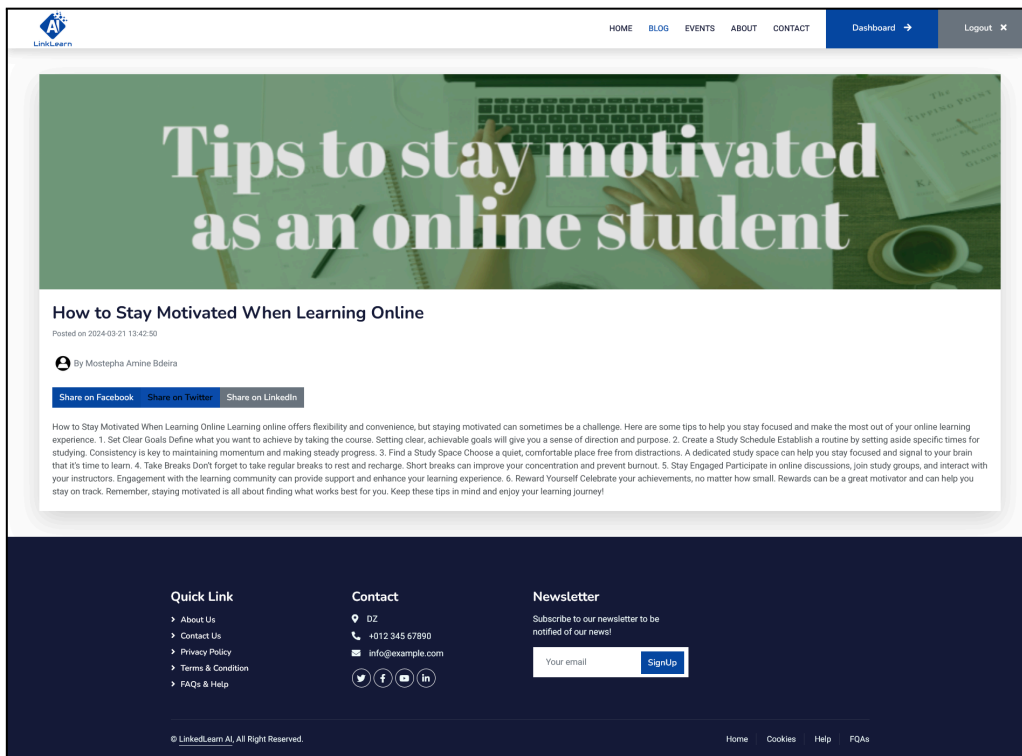
**Fig 48.** About Page



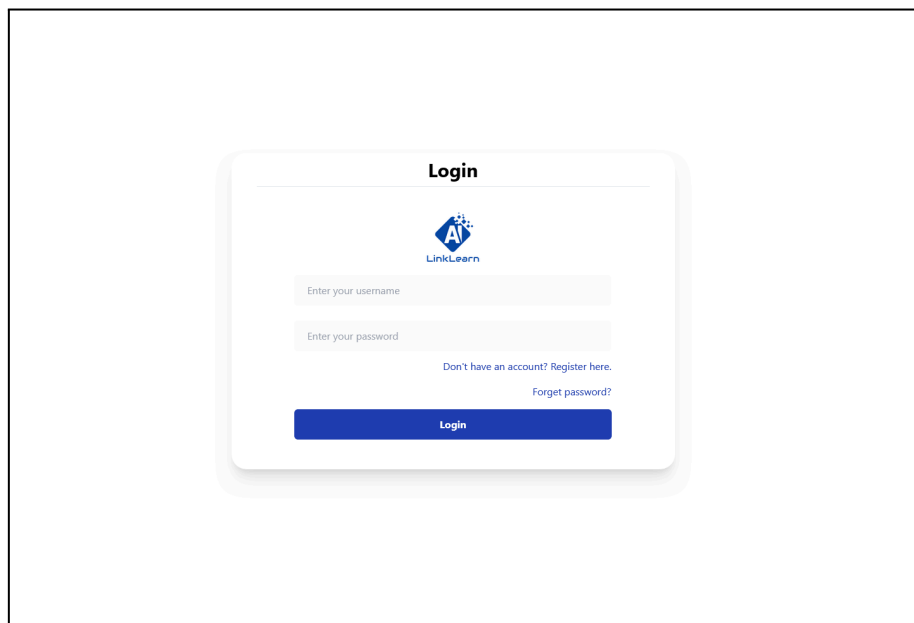
**Fig 49.** Events Page



**Fig 50.** Blog Page



**Fig 51.** Blog Post Page



**Fig 52.** Login Page

Register

Enter your first name

Enter your last name

Enter your username

Enter your email

Enter your password

jj / mm / aaaa

+123 4 56 78 90

Select a phone number that matches the format.

Do you agree terms of use ?

You have an account ? [Login here.](#)

Register

Fig 53. Register Page

AI

LinkLearn

Dashboard

Users

Messaging

Instructors

Courses

VideoDB

Create account

Search for projects

Home

Welcome, Mostepha Amine Bdeira

Welcome to your dashboard.

Latest Blog Posts

Top 5 Skills to Learn in 2024

Top 5 Skills to Lear...

Read more

How to Stay Motivated When Learning Online

How to Stay Motivate...

Read more

The Future of Online Learning

The Future of Online...

Read more

Quick Links

Dashboard

Create New Post

Upcoming Events

My Profile

Notifications

Learning Resources

Community Forums

Help & Support

Settings

Logout

Platform Stats

Users

50

Instructors

15

Courses

15

Fig 54. Dashboard Interface 1

88

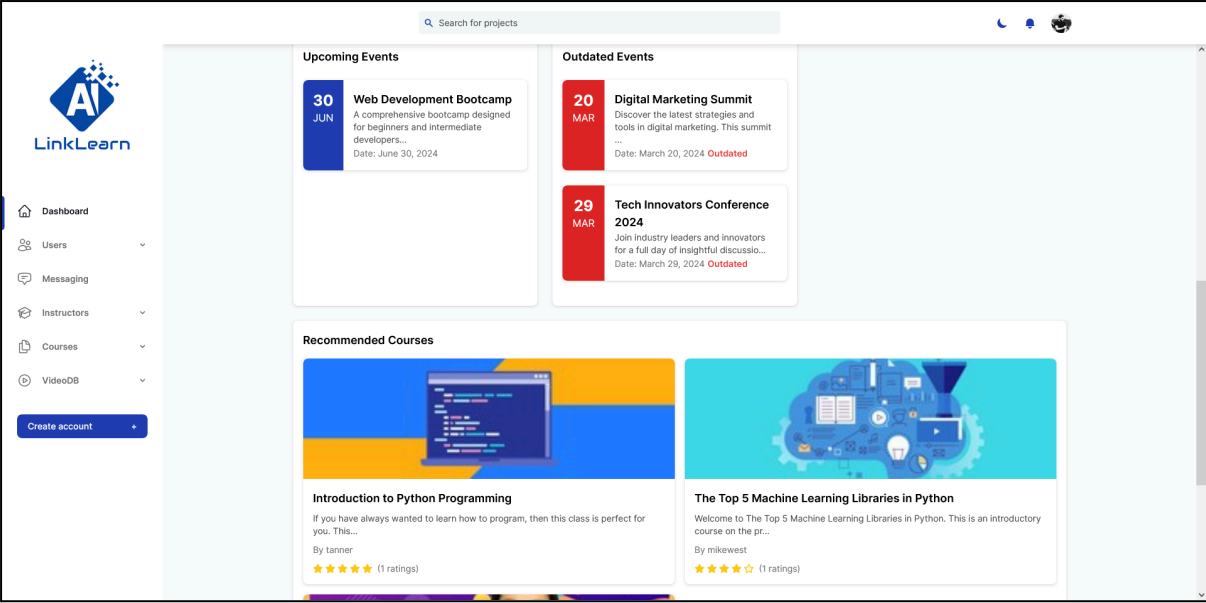


Fig 55. Dashboard Interface 2

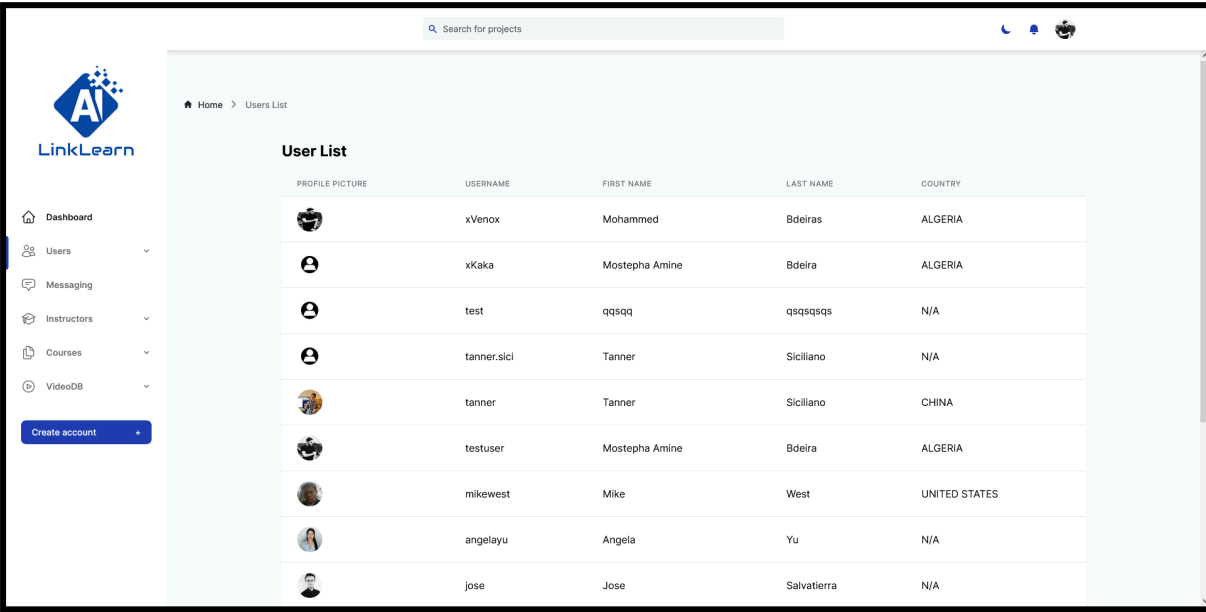
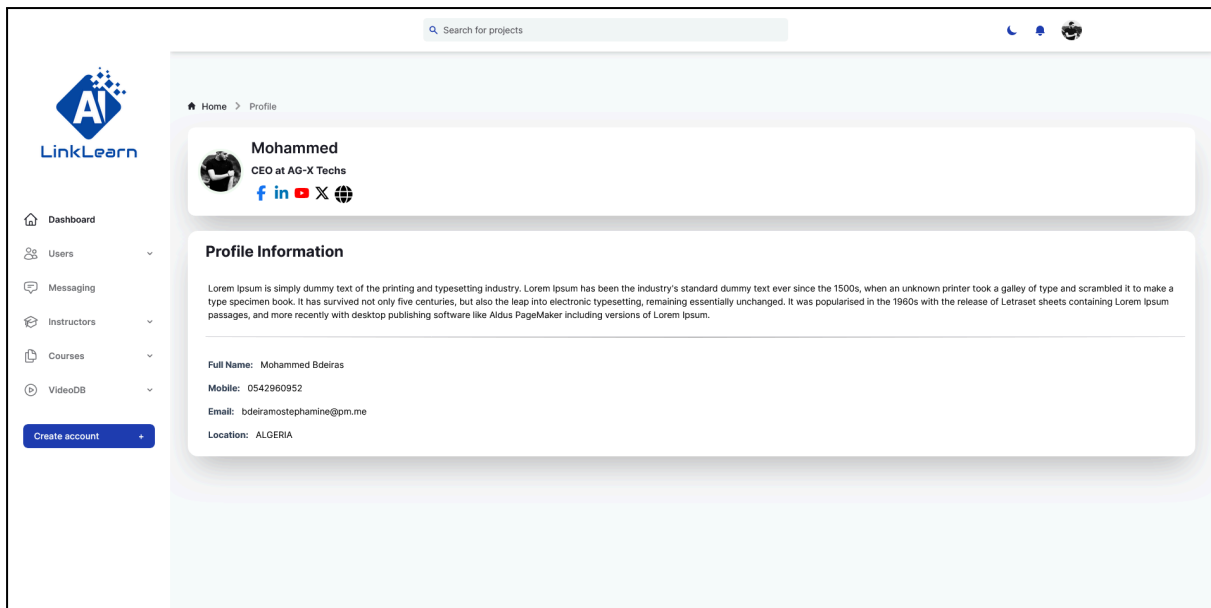
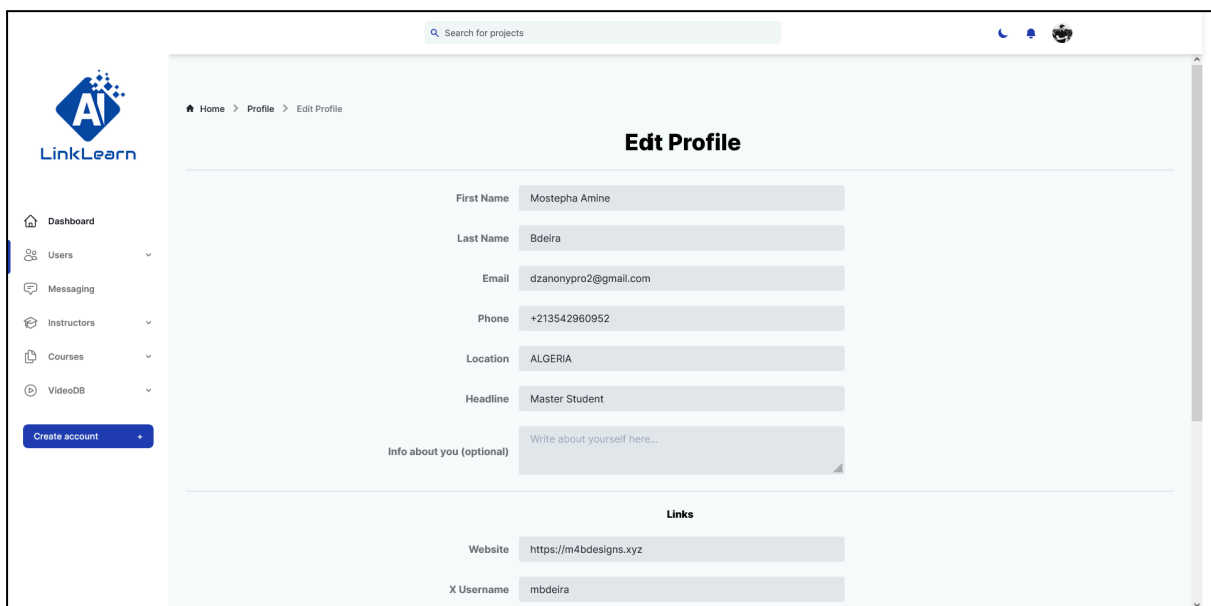


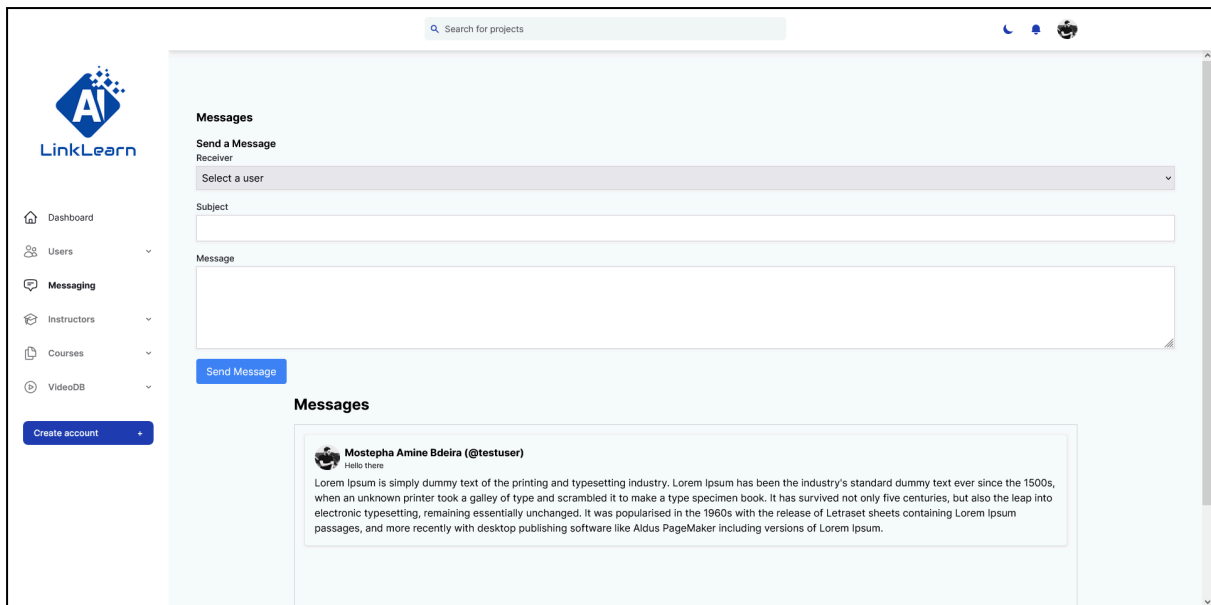
Fig 56. Users List Interface



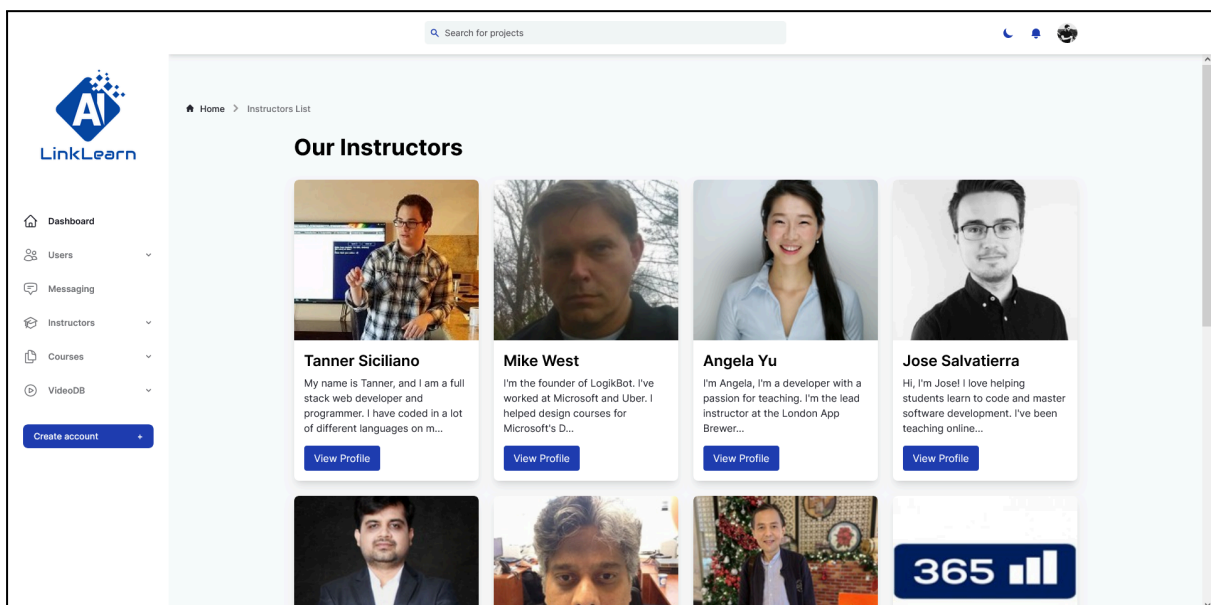
**Fig 57.** Profile view Interface



**Fig 58.** Profile edit Interface



**Fig 59.** Messaging Interface



**Fig 60.** Instructors List

AI

LinkLearn

Dashboard

Users

Blog Post

Messaging

Instructors

Courses

VideoDB

Create account

Search for projects

Home > Create Blog Post

Create a New Post

Title

Content

Image

Parcourir... Aucun fichier sélectionné.

Submit

Fig 61. Blog Post Creation List

AI

LinkLearn

Dashboard

Users

Blog Post

Messaging

Instructors

Courses

VideoDB

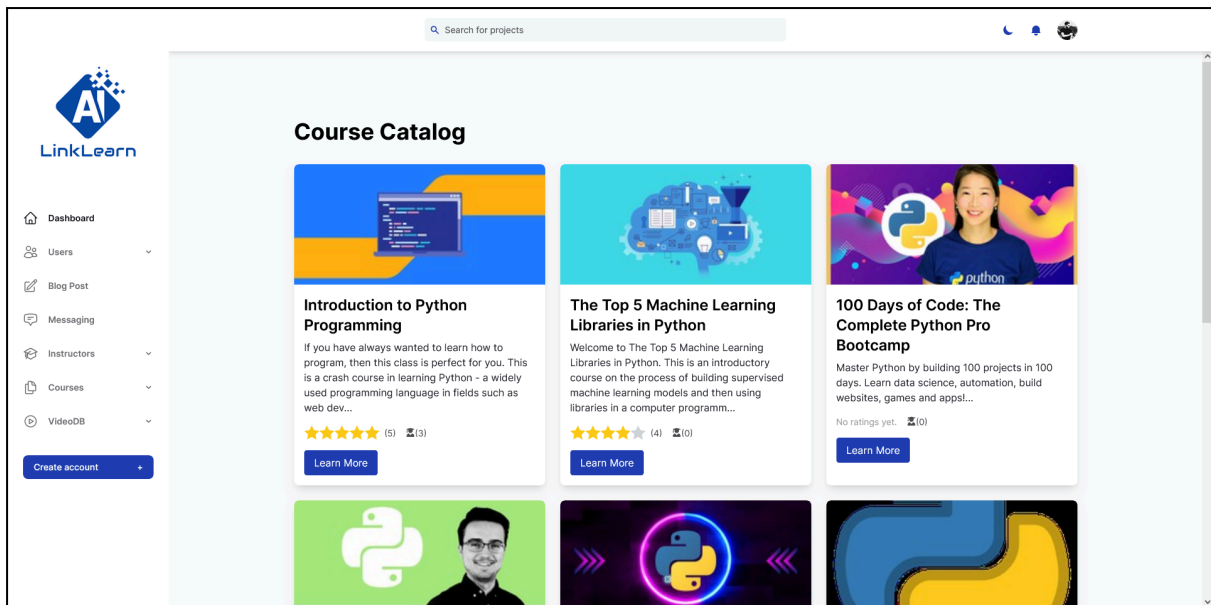
Create account

Search for projects

Select Videos for Assignment Generation

| SELECT                   | VIDEO TITLE                                          | TRANSCRIPT                                                                                              |
|--------------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| <input type="checkbox"/> | Introduction to Python - What is programming ?       | [Music] so we're going to look at what programming is okay CU it could be kind of confusing you hear... |
| <input type="checkbox"/> | Introduction to Python - Introduction                | hi my name's Tanner and thank you for taking the introduction to Python Programming course uh i'm ...   |
| <input type="checkbox"/> | Introduction to Python - Getting Started With Python | hi so we're going to jump right into python there's a lot of different tutorials on how to install...   |
| <input type="checkbox"/> | Introduction to Python - Hello World                 | so let's jump right into coding we're going to do our first program called hello world it's kind o...   |
| <input type="checkbox"/> | Introduction to Python - What is a Variable ?        | let's start variables right now a variable broadly speaking is simply a piece of information that ...   |
| <input type="checkbox"/> | Introduction to Python - Booleans                    | now that we've covered the basic idea of variables let's get into our first variable type which is...   |
| <input type="checkbox"/> | Introduction to Python - Ints and Floats             | we're going to start working with uh integers and floats so let's get right into it if we wanted t...   |
| <input type="checkbox"/> | Introduction to Python - Strings                     | now that we looked at basic variable types like integers and floats let's look at a string a strin...   |
| <input type="checkbox"/> | Introduction to Python - Tuples                      | now that we've really gone through the basic data types in Python we're going to go into some that...   |
| <input type="checkbox"/> | Introduction to Python - Lists                       | all right now that we covered tuples we're going to cover another piece of in uh data that's very ...   |
| <input type="checkbox"/> | Introduction to Python - Dictionaries                | so let's go into another very common python data type that you'll use a lot called a dictionary uh...   |
| <input type="checkbox"/> | Introduction to Python - Logic and Loops             | welcome to the logic and loop section of this course and this is what we're going to learn let's s...   |
| <input type="checkbox"/> | Introduction to Python - If, Elif and Else           | thank you for continuing to watch this course and we're going to get right into if statements if s...   |
| <input type="checkbox"/> | Introduction to Python - Example: FizzBuzz           | now that we've learned if statements we're going to look at a practical application of it it's a c...   |

Fig 62. Assignment generation interface



**Fig 63.** Courses Catalog

**Fig 64.** Course Creation interface

LinkLearn

- Dashboard
- Users
- Blog Post
- Messaging
- Instructors
- Courses
- VideoDB

Create account

Search for projects

| VIDEO TITLE                                          | URL                                         | ACTIONS          |
|------------------------------------------------------|---------------------------------------------|------------------|
| Introduction to Python - What is programming ?       | https://www.youtube.com/watch?v=IFkoU9fu6c  | Captions Fetched |
| Introduction to Python - Introduction                | https://www.youtube.com/watch?v=rByBWRhECw  | Captions Fetched |
| Introduction to Python - Getting Started With Python | https://www.youtube.com/watch?v=UrmMTgMtt_o | Captions Fetched |
| Introduction to Python - Hello World                 | https://www.youtube.com/watch?v=eIX-jfamtn8 | Captions Fetched |
| Introduction to Python - What is a Variable ?        | https://www.youtube.com/watch?v=g8W-MVYsAyY | Captions Fetched |
| Introduction to Python - Booleans                    | https://www.youtube.com/watch?v=1Kc80tpORvs | Captions Fetched |
| Introduction to Python - Ints and Floats             | https://www.youtube.com/watch?v=jcP5Jb7YacU | Captions Fetched |
| Introduction to Python - Strings                     | https://www.youtube.com/watch?v=1hH3Gv5WrxE | Captions Fetched |
| Introduction to Python - Tuples                      | https://www.youtube.com/watch?v=N6KPXhfp10w | Captions Fetched |
| Introduction to Python - Lists                       | https://www.youtube.com/watch?v=loCw-UJdiS4 | Captions Fetched |
| Introduction to Python - Dictionaries                | https://www.youtube.com/watch?v=tbSN_jL4uBk | Captions Fetched |
| Introduction to Python - Logic and Loops             | https://www.youtube.com/watch?v=SVX9phxy6dA | Captions Fetched |
| Introduction to Python - If, Elf and Else            | https://www.youtube.com/watch?v=Um117N2_kS8 | Captions Fetched |
| Introduction to Python - Example: FizzBuzz           | https://www.youtube.com/watch?v=lcgcJJsJPk8 | Captions Fetched |

Fig 65.Video database index + fetching transcriptions

LinkLearn

- Dashboard
- Users
- Blog Post
- Messaging
- Instructors
- Courses
- VideoDB

Create account

Search for projects

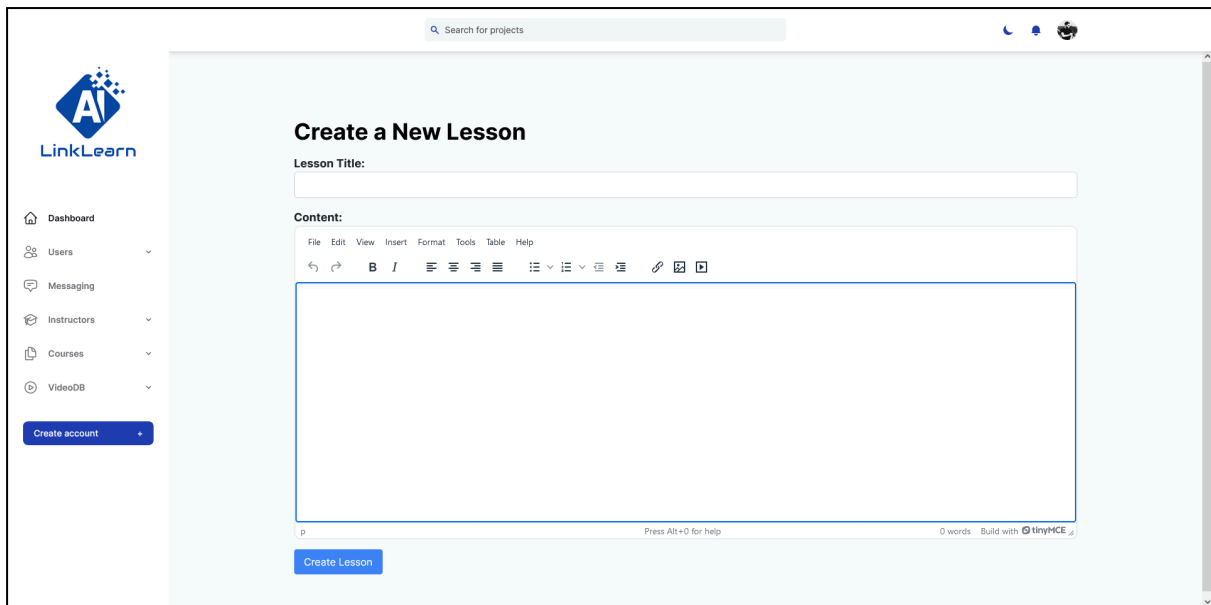
Video Title

YouTube URL

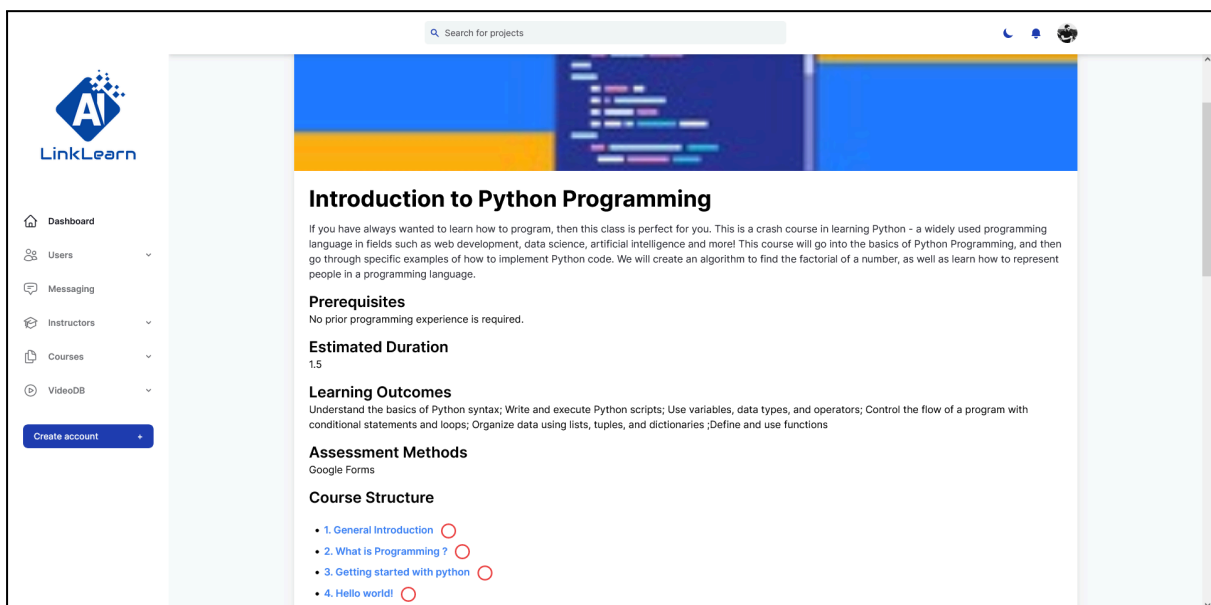
Course ID

Add Video

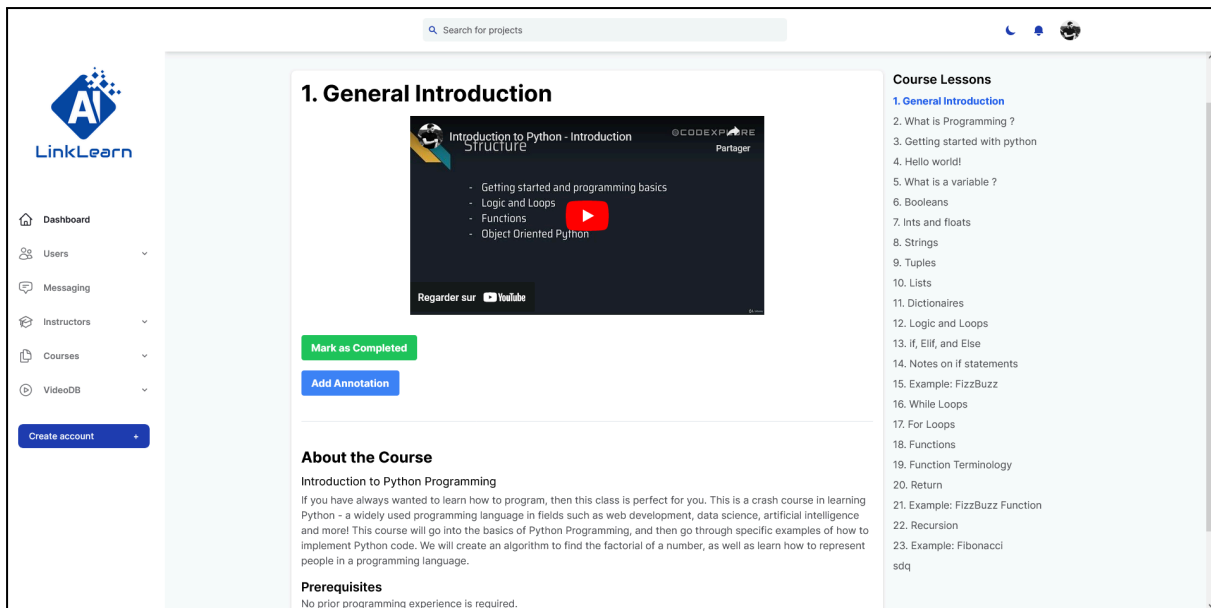
Fig 66.Adding videos metadata



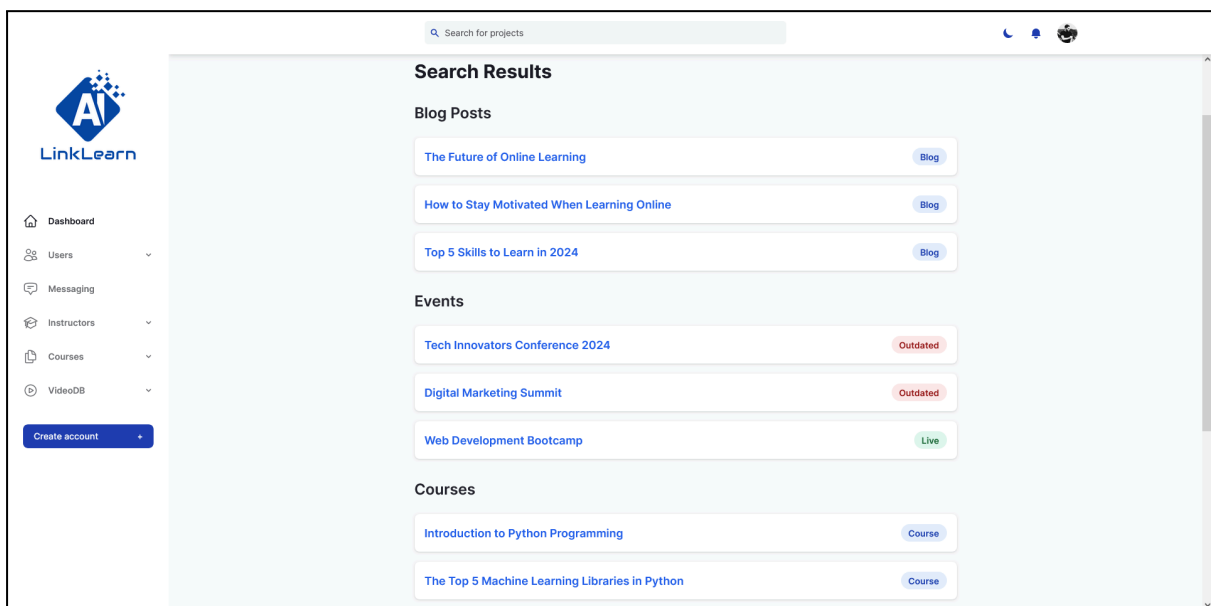
**Fig 67.**New lesson creation for instructors



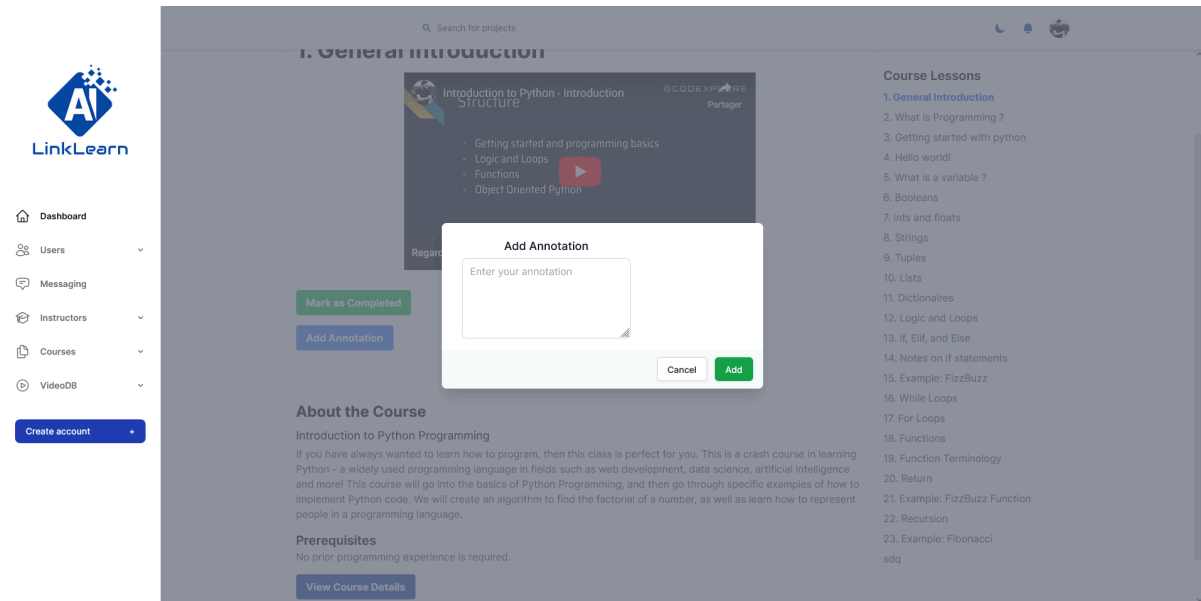
**Fig 68.**Course Preview



**Fig 69.** .Lesson Preview



**Fig 70.** .Search Engine Results



**Fig 71.** Adding Annotations

## CONCLUSION AND FUTURE WORK

### 1. SUMMARY

This dissertation discussed the design and deployment of an AI-powered e-learning platform that enabled the creation of a framework for highly personalized learning experiences and the automation of processes for educational content creation. Arguably, important facets of this delivery were integration of deep models for course recommendation, AI-driven assignment generation and video transcription, with annotation of interaction elements in video content. All this was achieved by combining certain theoretical underpinnings with hand-in-hand implementation to attain the stipulated goals, well illustrating AI with huge potentials in transforming any kind of educational environment.

#### **Key Achievements :**

**Personalized Course Recommendations :** Applied Surprise model from scikit-learn and DLRM models to provide personalized course recommendations for better student engagement and learning outcomes.

**Automated Assignment Generation :** Initiative that generated contextually correct assignments from lecture transcriptions using a Claude model, relieving massive loads of work from educators.

---

**Video Transcription and Annotation :** Implemented YouTubeTranscriptApi for video transcription, with addon tools for interactive video annotation to extend accessibility and make learning more enriching.

## 2. KEY FINDINGS AND IMPLICATIONS

### **Improved Learning Experience :**

Due to the high satisfaction rate of users, it was evident that the platform could deliver properly differentiated personalized learning paths.

The AI-driven tools really did what they promised: write educational content, really freeing teachers up for higher-order teaching tasks.

### **System Performance and Security :**

The performance tests were passed with good marks considering the load was great, page load times were low, and small amounts of downtime occurred.

This platform underwent some security measures, which included vulnerability scanning and penetration tests, none of which returned critical vulnerabilities.

### **User Feedback and Adaptability :**

The positive user feedback referred to the good intuitive interface and the efficiency of the personalized recommendations.

Flexibility—precisely because it can be used in a lot of educational situations at all faculty levels, with different subjects and levels of complexity.

## 3. LIMITATIONS AND CHALLENGES

### **Simulated User Testing :**

All testing phases were conducted with simulated users and, therefore, may lack the richness and variability of real usage that occurs in use. Future work needs to include extensive testing with real users to get more tangible feedback.

### **Bias and Fairness of Model :**

Publishers of AI-using products and associated social impact: Ensuring that AI models are fair and less biased is itself a challenge. Continuous audits are required, and fairness-aware algorithms have to be embedded inside them.

---

**Concerns with Scalability :**

Most of the performance under simulated loads was good, but actual deployment in the real world may add scalability and performance challenges. This will require monitoring and continuous optimization.

## **4. RECOMMENDATIONS FOR FUTURE WORK**

**Real-User Testing:**

- Conducting comprehensive testing with real users will provide valuable insights into user behavior and preferences, leading to further improvements in the platform's design and functionality.

**Advanced AI Models:**

- Exploring more advanced AI models and techniques can enhance the platform's capabilities, particularly in natural language processing and recommendation systems.

**Extended Functionality:**

- Integrating additional features, such as adaptive learning paths and real-time feedback mechanisms, can further personalize and improve the learning experience.

**Ethical Considerations:**

- Ongoing attention to data privacy, security, and ethical considerations will be crucial as the platform evolves. Implementing transparent data usage policies and obtaining informed consent from users will help maintain trust and compliance with regulations.

## **5. CONCLUSION**

The AI-enhanced e-learning platform developed in this research represents a significant advancement in leveraging AI technologies to improve educational experiences. By providing personalized course recommendations, automating assignment generation, and enriching video content with interactive annotations, the platform addresses key challenges in modern education. The positive results from usability, performance, and security testing indicate the platform's potential to significantly enhance student engagement and instructor efficiency.

---

This research underscores the transformative potential of AI in education, paving the way for future innovations that can create more personalized, efficient, and interactive learning environments. By addressing the identified limitations and continuing to explore advanced AI techniques, the platform can further evolve to meet the dynamic needs of educators and learners in an increasingly digital world.

---

## BIBLIOGRAPHY

- [1] *learning management system (LMS)*. (2023, September 8). CIO. Retrieved June 3, 2024, from <https://www.techtarget.com/searchcio/definition/learning-management-system>
- [2] *Massive open online course*. (2024, June 3). Wikipedia. Retrieved June 3, 2024, from [https://en.wikipedia.org/wiki/Massive\\_open\\_online\\_course](https://en.wikipedia.org/wiki/Massive_open_online_course)
- [3] *What is a Virtual Learning Environment (VLE)? Types & Advantages*. (2024, March 25). Cloud Assess. Retrieved June 3, 2024, from <https://cloudassess.com/blog/virtual-learning-environment/>
- [4] *Adaptive Learning | eCampusOntario*. (2023, September 13). eCampusOntario. Retrieved June 3, 2024, from <https://www.ecampusontario.ca/adaptive-learning/>
- [5] *10 Benefits of Online Learning & The Advantages of Learning Platforms*. (2024, January 18). WorkRamp. Retrieved June 3, 2024, from <https://www.workramp.com/blog/benefits-online-learning-platforms-for-business/>
- [6] Rawashdeh, A. Z. A., Mohammed, E. Y., Arab, A. R. A., Alara, M., Al-Rawashdeh, B., & Al-Rawashdeh, B. (2021). Advantages and Disadvantages of Using e-Learning in University Education: Analyzing Students' Perspectives. *Electronic Journal of e-Learning*, 19(3), 107–117. <https://doi.org/10.34190/ejel.19.3.2168>
- [7] Islam, N., Beer, M., & Slack, F. (2015). E-Learning Challenges Faced by Academics in Higher Education: A Literature Review. *Journal of Education and Training Studies*, 3(5). <https://doi.org/10.11114/jets.v3i5.947>
- [8] Naresh, B., & Reddy, D. B. S. (2015). Current Trends in E-Learning and Future Scenario. *Mediterranean Journal of Social Sciences*. <https://doi.org/10.5901/mjss.2015.v6n5p484>

- 
- [9] *Ai in Education Statistics - Analyzing the Impact of Artificial Intelligence on Educational Outcomes*. (2023, December 3). AI News. Retrieved June 3, 2024, from <https://aquariusai.ca/blog/ai-in-education-statistics-analyzing-the-impact-of-artificial-intelligence-on-educational-outcomes>
- [10] *New global data reveal education technology's impact on learning | Unesco IIEP Learning Portal*. (n.d.). Retrieved June 3, 2024, from <https://learningportal.iiep.unesco.org/es/biblioteca/new-global-data-reveal-education-technologies-impact-on-learning>
- [11] *The Transformative Power of Deep Learning in Education*. (2024, February 21). Retrieved June 4, 2024, from <https://www.linkedin.com/pulse/transformative-power-deep-learning-education-luis-a-barroso-tnlie>
- [12] Khoei, T. T., Slimane, H. O., & Kaabouch, N. (2023). Deep learning: systematic review, models, challenges, and research directions. *Neural Computing & Applications*, 35(31), 23103–23124. <https://doi.org/10.1007/s00521-023-08957-4>
- [13] Nguoi, Catherine & Habil, Hadina. (2021). The Use of Video Annotation in Education: A Review. *Asian Journal of University Education*. 17. 84. 10.24191/ajue.v17i4.16208.
- [14] Vrochidis, Stefanos & Patras, Ioannis & Kompatsiaris, Ioannis. (2017). Gaze movement-driven random forests for query clustering in automatic video annotation. *Multimedia Tools and Applications*. 76. 10.1007/s11042-015-3221-1.
- [15] *Video annotation: what is it, and how does it work?* (2023, June 19). Retrieved June 4, 2024, from <https://www.springbord.com/blog/video-annotation-what-is-it-and-how-does-it-work/>

- 
- [16] Hansch, Anna & Hillers, Lisa & McConachie, Katherine & Newman, Christopher & Schildhauer, Thomas & Schmidt, Philipp. (2015). Video and Online Learning: Critical Reflections and Findings from the Field. SSRN Electronic Journal. 10.2139/ssrn.2577882.
- [17] Liao, H., Chen, L., Song, Y., & Ming, H. (2016). Visualization-Based Active Learning for Video Annotation. *IEEE Transactions on Multimedia*, 18, 2196-2205.
- [18] Kim, J., Gwon, R., Park, J., Kim, H., & Kim, Y. (2013). A Semi-Automatic Video Annotation Tool to Generate Ground Truth for Intelligent Video Surveillance Systems. *Advances in Mobile Multimedia*.
- [19] Adhikari, B., Peltomäki, J., Puura, J., & Huttunen, H. (2018). Faster Bounding Box Annotation for Object Detection in Indoor Scenes. *2018 7th European Workshop on Visual Information Processing (EUVIP)*, 1-6.
- [20] *Video annotation: what is it, and how does it work?* (2023, June 19). Retrieved June 3, 2024, from <https://www.springbord.com/blog/video-annotation-what-is-it-and-how-does-it-work/>
- [21] Moodle US (2023). What is an LMS? Learning management systems explained. Retrieved From <https://moodle.com/news/what-is-an-lms-learning-management-systems-explained/>
- [22] *What Is Blackboard Learn?* (n.d.). Retrieved From [https://help.blackboard.com/Learn/Instructor/Ultra/Getting\\_Started/What\\_Is\\_Blackboard\\_Learn](https://help.blackboard.com/Learn/Instructor/Ultra/Getting_Started/What_Is_Blackboard_Learn)
- [23] *Coursera*. (2023, November 1). Retrieved From <https://fr.wikipedia.org/wiki/Coursera>
- [24] Koumadoraki, A., & Koumadoraki, A. (2024, June 3). *10 eLearning Trends That Will Go Big in 2023*. LearnWorlds. <https://www.learnworlds.com/elearning-trends/>

- 
- [25] Nurfadillah, R. (2022, January 29). Get To Know with Surprise - tiket.com - Medium.  
Medium. <https://medium.com/tiket-com/get-to-know-with-surprise-2281dd227c3e>
- [26] Iterators, Iterators, & Iterators. (2021, July 15). *Collaborative Filtering In Recommender Systems: Learn All You Need To Know* |. Iterators.  
<https://www.iteratorshq.com/blog/collaborative-filtering-in-recommender-systems/>
- [27] *Singular value decomposition*. (2024, May 24). Wikipedia.  
[https://en.wikipedia.org/wiki/Singular\\_value\\_decomposition](https://en.wikipedia.org/wiki/Singular_value_decomposition)
- [28] Hinno, R. (2021, December 29). Simple SVD algorithms - Towards Data Science.  
Medium. <https://towardsdatascience.com/simple-svd-algorithms-13291ad2eef2>
- [29] *Power iteration*. (2023, December 5). Wikipedia.  
[https://en.wikipedia.org/wiki/Power\\_iteration](https://en.wikipedia.org/wiki/Power_iteration)
- [30] *Singular Value Decomposition Part 2: Theorem, Proof, Algorithm*. (2016, May 16).  
Math  $\cap$  Programming.  
<https://jeremykun.com/2016/05/16/singular-value-decomposition-part-2-theorem-proof-algorithm/>
- [31] *What is Unified Modeling Language (UML)?* (n.d.).  
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>
- [32] *Rational Software Architect 9.6.1*. (n.d.).  
<https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case>
- [33] GeeksforGeeks. (2024b, May 1). *Class Diagram Unified Modeling Language (UML)*.  
GeeksforGeeks.  
<https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/>
- [34] *What is a Sequence Diagram?* (n.d.).  
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

- 
- [35] *Welcome to Claude - Anthropic*. (n.d.). Anthropic.  
<https://docs.anthropic.com/en/docs/intro-to-claude>
- [36] Hawkins, B. (2024, April 19). *Claude AI Model Versions Explained - Claude101*.  
Claude101. <https://claude101.com/claude-ai-model-versions-explained/>
- [37] *Linked Data - W3C Wiki*. (n.d.). <https://www.w3.org/wiki/LinkedData>
- [38] - *Wiki Education Dashboard*. (n.d.).  
<https://dashboard.wikiedu.org/training/wikidata-professional/databases-and-linked-data/advantages-linked-data>
- [39] *RDF - Semantic Web Standards*. (n.d.). <https://www.w3.org/RDF/>
- [40] *RDFS - Semantic Web Standards*. (n.d.). <https://www.w3.org/2001/sw/wiki/RDFS>
- [41] *OWL - Semantic Web Standards*. (n.d.). <https://www.w3.org/2001/sw/wiki/OWL>
- [42] *SPARQL - Semantic Web Standards*. (n.d.). <https://www.w3.org/2001/sw/wiki/SPARQL>
- [43] *FOAF Vocabulary Specification*. (n.d.). <http://xmlns.com/foaf/spec/>
- [44] *SKOS - Semantic Web Standards*. (n.d.). <https://www.w3.org/2001/sw/wiki/SKOS>
- [45] Cudré-Mauroux, P. (2017, January 31). *Linked Data: Present & Future - SocietyByte*.  
Society Byte. <https://www.societybyte.swiss/2017/02/01/linked-data-present-future/>
- [46] *Windows 10 version history*. (2024, June 12). Wikipedia.  
[https://en.wikipedia.org/wiki/Windows\\_10\\_version\\_history](https://en.wikipedia.org/wiki/Windows_10_version_history)
- [47] *Visual Studio Code*. (2024, May 24). Wikipedia.  
[https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)
- [48] *XAMPP*. (2024, June 6). Wikipedia. <https://en.wikipedia.org/wiki/XAMPP>
- [49] *Npm*. (2024, June 4). <https://en.wikipedia.org/wiki/Npm>
- [50] *Apache Spark*. (2024, June 5). Wikipedia. [https://en.wikipedia.org/wiki/Apache\\_Spark](https://en.wikipedia.org/wiki/Apache_Spark)

- 
- [51] *CodeIgniter - Overview*. (n.d.).  
[https://www.tutorialspoint.com/codeigniter/codeigniter\\_overview.htm](https://www.tutorialspoint.com/codeigniter/codeigniter_overview.htm)
- [52] GeeksforGeeks. (2024, April 16). *MVC Framework Introduction*. GeeksforGeeks.  
<https://www.geeksforgeeks.org/mvc-framework-introduction/>
- [53] *Python (programming language)*. (2024, June 11). Wikipedia.  
[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [54] *PHP*. (2024, June 11). Wikipedia. <https://en.wikipedia.org/wiki/PHP>
- [55] *HTML5*. (2024, June 6). Wikipedia. <https://en.wikipedia.org/wiki/HTML5>
- [56] *CSS*. (2024, May 6). Wikipedia. <https://en.wikipedia.org/wiki/CSS>
- [57] *JavaScript*. (2001, November 19). Wikipedia. <https://en.wikipedia.org/wiki/JavaScript>
- [58] *Tailwind CSS*. (2024, May 16). Wikipedia. [https://en.wikipedia.org/wiki/Tailwind\\_CSS](https://en.wikipedia.org/wiki/Tailwind_CSS)
- [59] *MySQL*. (2024, May 15). Wikipedia. <https://en.wikipedia.org/wiki/MySQL>
- [60] *youtube-transcript-api*. (2023, December 27). PyPI.  
<https://pypi.org/project/youtube-transcript-api/>
- [61] Hub, C. A. (2024, March 29). *Beginner's Guide to Claude API - How to Use | Claude AI Hub*. Claude AI Hub. <https://claudeaihub.com/claude-ai-api-guide/>
- [62] *Video.js - Make your player yours*. (n.d.). Video.js. <https://videojs.com/>
- [63] *Keras*. (2024, June 11). Wikipedia. <https://en.wikipedia.org/wiki/Keras>
- [64] Team, K. (n.d.). *Keras documentation: Adam*. <https://keras.io/api/optimizers/adam/>
- [65] Stewart, K. (2024, May 24). *Mean squared error (MSE) | Definition, Formula, Interpretation, & Facts*. Encyclopedia Britannica.  
<https://www.britannica.com/science/mean-squared-error>

---

[66] *SAP Help Portal*. (n.d.).

[https://help.sap.com/docs/SAP\\_PREDICTIVE\\_ANALYTICS/41d1a6d4e7574e32b815f1cc87c00f42/5e5198fd4afe4ae5b48fefe0d3161810.html](https://help.sap.com/docs/SAP_PREDICTIVE_ANALYTICS/41d1a6d4e7574e32b815f1cc87c00f42/5e5198fd4afe4ae5b48fefe0d3161810.html)

[67] *Flask (web framework)*. (2024, May 7). Wikipedia.

[https://en.wikipedia.org/wiki/Flask\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework))

[68] *Precision and recall at  $K$  in ranking and recommendations*. (n.d.).

<https://www.evidentlyai.com/ranking-metrics/precision-recall-at-k>