

**UNIVERSITY OF KARACHI**

**Department of Computer Science**

B.S. Software Engineering

FINAL YEAR PROJECT REPORT

**Batch-2019**

**Academia Assist**

**By**

|  |  |
| --- | --- |
| HUR ABBAS (Lead) | B18158017 |
| ZOHAIB NAVED | B 19103073 |
| YUMNA WARIS | B 19103071 |
| MOIZ ALI | B 19103028 |

**Supervised by**

**Muhammad Saeed**

*W4W8+946 KU, Circular Road, University of Karachi,* *Karachi, Pakistan. Phone:* [*(0992) 613007*](https://www.google.com/search?q=DCS+UBIT&rlz=1C1OPNX_enPK1091PK1091&oq=DCS+UBIT&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDI0NTdqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8)*;*  [*https://uok.edu.pk/faculties/computerscience/ubit.php*](%20https://uok.edu.pk/faculties/computerscience/ubit.php%20)

# Declaration

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UNIVERSITY OF KARACHI or other institutions.

|  |
| --- |
| HUR ABBAS (B18158017): |
| Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| ZOHAIB NAVED (B19103073): |
| Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| YUMNA WARIS (B19103071) |
| Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| MOIZ ALI (B19103028) |
| Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# Acknowledgments

In heartfelt homage to the authors who paved the path of our academic journey, I, Yumna Waris (B19103071), dedicate this FYP report. Your insightful words have been my guiding stars, fostering curiosity and fueling my passion for knowledge. As I traverse my field, your intellectual legacy lives on in the fabric of my ideas and conclusions. This dedication stands as a token of gratitude for the profound impact you've had on my learning. Though physically distant, your influence resonates in every line of this report. Thank you for being the silent mentors in my educational expedition.

# Abstract

Academia Assist, driven by the ChatGPT API, transforms the educational landscape by automating course planning and exam generation. This web application, tailored for educators, streamlines the creation of course outlines and exams, emphasizing efficiency and originality. Named intentionally, Academia Assist not only automates tasks but also champions a cheat-free learning environment, ensuring exam authenticity and fostering creativity in teaching materials. In response to the absence of a Learning Management System (LMS), the platform hosts files on AWS for secure student access. This abstract encapsulates Academia Assist's pivotal role in reshaping academia, providing a dynamic solution for educators to navigate course creation and examinations, ultimately enhancing the overall educational experience.

**Contents**

[Declaration 2](#_Toc157551086)

[Acknowledgments 3](#_Toc157551087)

[Abstract 4](#_Toc157551088)

[1. Introduction 11](#_Toc157551089)

[1.1. Problem Definition 11](#_Toc157551090)

[1.2. Problem Statement 12](#_Toc157551091)

[1.3. Challenges and Pain Points 12](#_Toc157551092)

[1.3.1. Difficulty in Creating Fresh Exams 12](#_Toc157551093)

[1.3.2. Limited Access to Course Materials 12](#_Toc157551094)

[1.3.3. Inefficiencies in Exam Generation 12](#_Toc157551095)

[1.3.4 Missed Opportunities for Enhanced Learning 12](#_Toc157551096)

[1.3.5 Project Aim 12](#_Toc157551097)

[1.4. Expected Impact 13](#_Toc157551098)

[1.4.1. Enhanced Learning Experience 13](#_Toc157551099)

[1.4.2. Streamlined Exam Creation 13](#_Toc157551100)

[1.4.3. Academic Integrity Promotion 13](#_Toc157551101)

[1.4.4. Easy Access to Course Materials 13](#_Toc157551102)

[1.5. Project Executive Summary 14](#_Toc157551103)

[1.5.1. Problem Addressed 14](#_Toc157551104)

[1.5.2. Project Objectives 14](#_Toc157551105)

[1.6. List of symbols and Units 15](#_Toc157551106)

[1.7. System Diagram 17](#_Toc157551107)

[2. Background: 19](#_Toc157551108)

[2.1. Existing Solution: 19](#_Toc157551109)

[2.1.1. Chegg Mate AI 19](#_Toc157551110)

[2.1.2. Quizizz 20](#_Toc157551111)

[2.2. Our Solution: 22](#_Toc157551112)

[2.2.1. Academia Assist 22](#_Toc157551113)

[2.2.2. Comparison Table 23](#_Toc157551114)

[3. Aim and Statement of Problem 25](#_Toc157551115)

[3.1 Problem Statement 25](#_Toc157551116)

[3.2 Scope of Project 25](#_Toc157551117)

[3.3 Solutions for Stakeholders 25](#_Toc157551118)

[4 Hardware, Software Analysis and Requirements 28](#_Toc157551119)

[4.1 Languages and Libraries 28](#_Toc157551120)

[4.1.1. Backend Languages and Libraries: 28](#_Toc157551121)

[4.1.2. Front-end languages and Libraries: 28](#_Toc157551122)

[4.2 Tools 28](#_Toc157551123)

[4.2.1 Star UML 28](#_Toc157551124)

[4.2.2 DRAW.IO 29](#_Toc157551125)

[4.2.3 VS Code 29](#_Toc157551126)

[4.2.4 Prettier: 29](#_Toc157551127)

[4.3 Hardware Requirements 29](#_Toc157551128)

[4.4 Algorithms 29](#_Toc157551129)

[4.5 Libraries 29](#_Toc157551130)

[4.5.1 Fast API 29](#_Toc157551131)

[4.5.2 Scikit-learn (sklearn): 29](#_Toc157551132)

[4.5.3 Langchain: 29](#_Toc157551133)

[4.5.4 OpenAI's API: 30](#_Toc157551134)

[4.6 Revision 30](#_Toc157551135)

[4.6.1 Flexibility 30](#_Toc157551136)

[4.6.2 Maintainability 30](#_Toc157551137)

[4.6.3 Portability 30](#_Toc157551138)

[4.7 (Diagrams) Actor Used Case Diagram: 31](#_Toc157551139)

[4.7.1 Activity Diagram: 32](#_Toc157551140)

[5. Software Design and Modeling 34](#_Toc157551141)

[5.1 Deployment Architecture 34](#_Toc157551142)

[5.2 Class Diagram 35](#_Toc157551143)

[5.2.1 Component Diagram 36](#_Toc157551144)

[5.2.2 State Transition Diagram 37](#_Toc157551145)

[5.2.3 Prototypes 41](#_Toc157551146)

[5.2.4 Operational Diagram 43](#_Toc157551147)

[5.2.5 Deployment Diagram 44](#_Toc157551148)

[6. Implementation 46](#_Toc157551149)

[7.1. FastAPI Backend 47](#_Toc157551150)

[7.2. Streamlit Frontend 47](#_Toc157551151)

[7.3. LangChain and ChatGPT 47](#_Toc157551152)

[7.4. Prompt Templates and Parsers 47](#_Toc157551153)

[7.5. Pydantic Models 47](#_Toc157551154)

[7.6. Services and Async Functions 47](#_Toc157551155)

[7. Testing (Black Box Testing): 48](#_Toc157551156)

[8. Conclusions 50](#_Toc157551157)

[9. Future work 52](#_Toc157551158)

[10. Appendices 54](#_Toc157551159)

[12.1 Appendices A: Executive Summary 54](#_Toc157551160)

[12.1.1. Executive Summary 54](#_Toc157551161)

[12.2 Appendix B: Project Overview 54](#_Toc157551162)

[12.1.2. Project Overview: 54](#_Toc157551163)

[12.3 Appendix C: Project Objectives 54](#_Toc157551164)

[12.4 Appendix D: Project Scope 55](#_Toc157551165)

[12.4.1 In-Scope 55](#_Toc157551166)

[12.4.2 Out of Scope 55](#_Toc157551167)

[12.5 Appendix E: Deliverables Produced 55](#_Toc157551168)

[12.6 Appendix F: Tools And Technologies 55](#_Toc157551169)

[12. References 57](#_Toc157551170)

**List of tables**

COMPARISON TABLE………………………………………………….25

**List of figures**

[Figure 1.1 SYSTEM DIAGRAM](#page12)  [1](#page12)9

Figure [2.1 WEB INTERFACE](#page14) 22

[Figure 4.1 ACTOR USE CASE DIAGRAM](#page29)  41

CHAPTER 1

INTRODUCTION

# Introduction

Welcome to Academia Assist, a revolutionary web application transforming the landscape of academic content creation and examination generation for educators. In the ever-evolving realm of academia, the need for efficient and innovative tools is crucial. Our platform, developed using the ChatGPT API, addresses this need by simplifying the creation of course outlines, exams, and teaching content, enhancing the overall learning experience.

Academia Assist's primary objectives include:

* Formulating course outlines with precise formatting
* Generating exams with proper structure
* Crafting engaging teaching content
* Establishing connections between similar courses across diverse programs

For educators, the platform offers an intuitive interface to effortlessly generate exams, eliminating the repetition of old questions. This approach aims to elevate the quality of learning for students, fostering a cheat-free environment. Furthermore, to overcome the absence of an LMS system in our university, Academia Assist ensures easy student access by hosting generated files on AWS.

The driving force behind Academia Assist is to empower teachers and TAs, providing them with a tool that not only enhances efficiency but also ensures a fair and secure academic environment. Our commitment is to reshape the academic experience, discouraging reliance on outdated materials and fostering originality in teaching and assessment.

With Academia Assist, the future of academia is marked by efficiency, integrity, and accessibility, ensuring a transformative experience for both educators and students alike.

## Problem Definition

In this section, we will provide a detailed definition of the problem our project aims to address. In Pakistan, a significant portion of the population moves to the Middle East for employment opportunities often leaves families split, with parents working abroad while their children pursue higher education back home. This situation presents a unique challenge where families encounter difficulties in managing essential household maintenance and repair services.

In this section, we delineate the problem our project, Academia Assist, aims to solve. Within the academic landscape, universities often lack a streamlined system for generating course outlines, exams, and teaching content. This absence of efficient tools poses challenges for educators, especially when attempting to create assessments without repeating old questions, enhancing the overall learning experience for students.

The prevailing reliance on outdated past papers, accessible to select students, fosters an environment conducive to academic dishonesty. With no dedicated Learning Management System (LMS) in our university, the need for a centralized platform becomes apparent. Academia Assist seeks to fill this void, providing educators with a user-friendly interface for generating exams seamlessly, while also hosting files on AWS for universal student access, thus discouraging academic misconduct.

The core objective is to revolutionize academic experience by empowering teachers and TAs with a tool that ensures integrity, originality, and accessibility in the creation and distribution of course materials. Academia Assist addresses the challenges posed by outdated practices and aims to usher in a new era of efficiency and fairness in academic content generation.

## Problem Statement

In our university, there's no easy way for teachers to make new exams without using old questions, which can lead to cheating. We also don't have a system for students to access their course materials easily. So, we created Academia Assist, a simple web app that uses ChatGPT to help teachers make new course outlines and exams. This way, students get a better learning experience, and files are stored on AWS for easy access. Our goal is to stop cheating and make studying more straightforward for everyone.

## Challenges and Pain Points

### Difficulty in Creating Fresh Exams

Teachers at our university face challenges in generating new exam papers without relying on old questions, leading to potential issues of cheating. The lack of a streamlined process hinders the creation of diverse and innovative assessments.

### Limited Access to Course Materials

Students encounter difficulties in accessing their course materials efficiently. The absence of a centralized system makes it challenging for them to retrieve necessary resources, impacting their learning experience.

### Inefficiencies in Exam Generation

Teachers, especially those with limited time due to other commitments, struggle to quickly create well-structured exams. The current process often involves extensive searching and manual compilation, leading to inefficiencies.

### 1.3.4 Missed Opportunities for Enhanced Learning

The absence of a dedicated platform hampers the opportunity for teachers to provide a more enriched learning experience. Customized and varied content could significantly contribute to a better understanding of the subject matter.

### 1.3.5 Project Aim

Academia Assist is our solution to help teachers create exams without using old questions, preventing cheating at our university. This simple web app uses ChatGPT to make course outlines, exams, and teaching content easily. We've also added a feature to check if different courses are too similar. Our aim is to stop cheating, make studying straightforward, and provide accessible files on AWS for students and teachers, especially when our university lacks a proper system for this. Our goal is to make learning better for teachers and students.

## Expected Impact

The successful implementation of the project will have significant positive impacts:

### Enhanced Learning Experience

Academia Assist is expected to empower both teachers and students, offering an easy way to create exams and access course materials. Teachers can make fresh exams without repeating old questions, ensuring a fair assessment process, while students benefit from a better learning experience through diverse content generation.

### Streamlined Exam Creation

The platform's impact includes streamlining the exam creation process for teachers. With Academia Assist, educators can efficiently generate course outlines and exams with proper formatting, eliminating the reliance on outdated questions. This efficiency is crucial for educators with time constraints.

### Academic Integrity Promotion

Academia Assist's similarity analysis feature ensures academic integrity by preventing the reuse of questions across different courses. This contributes to a fair examination environment, addressing the issue of potential cheating arising from the misuse of old past papers.

### Easy Access to Course Materials

The platform's integration with AWS ensures easy access to generated files for both teachers and students. This addresses the lack of a dedicated learning management system (LMS) in our university, providing a solution for accessible and organized course materials.

In summary, Academia Assist aims to revolutionize exam creation and learning accessibility. The expected impact encompasses an enhanced learning experience, streamlined exam creation, promotion of academic integrity, and easy access to course materials. This initiative aligns with addressing the challenges identified and aims to make studying more straightforward for everyone at our university.

## Project Executive Summary

Our project, Academia Assist, is a groundbreaking initiative to address the challenges faced by teachers and students at our university. With the absence of an easy method for creating fresh exams and a system for students to access course materials, we've developed a simple web app leveraging ChatGPT to streamline the exam creation process and enhance learning experiences. Our objectives revolve around making studying more straightforward and eliminating cheating. Here's a concise summary.

### Problem Addressed

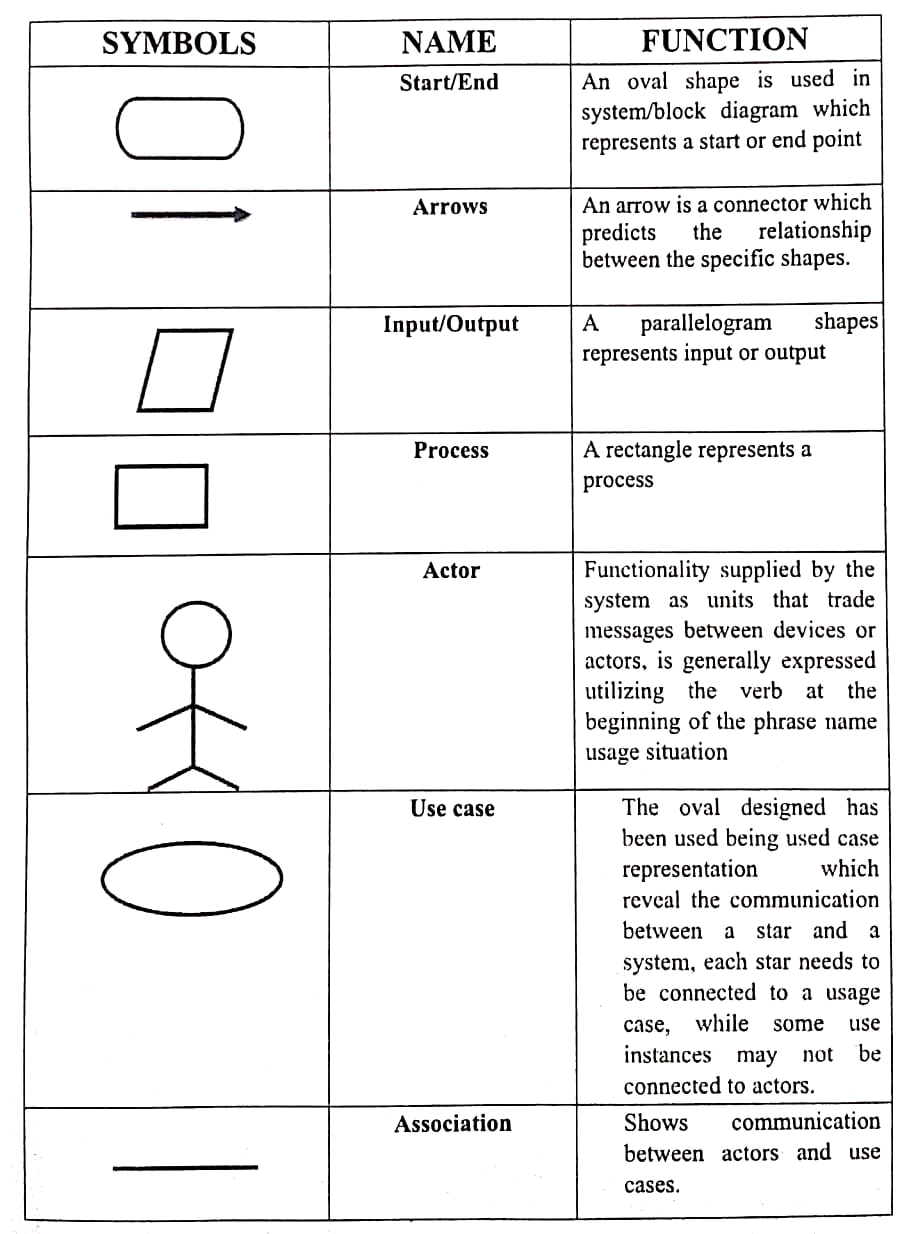
Our project directly tackles the issue of exam creation challenges faced by teachers, leading to potential cheating. Additionally, the lack of a system for students to access course materials easily contributes to a less-than-optimal learning experience. Academia Assist aims to eliminate these problems, ensuring a fair and enriching academic environment.

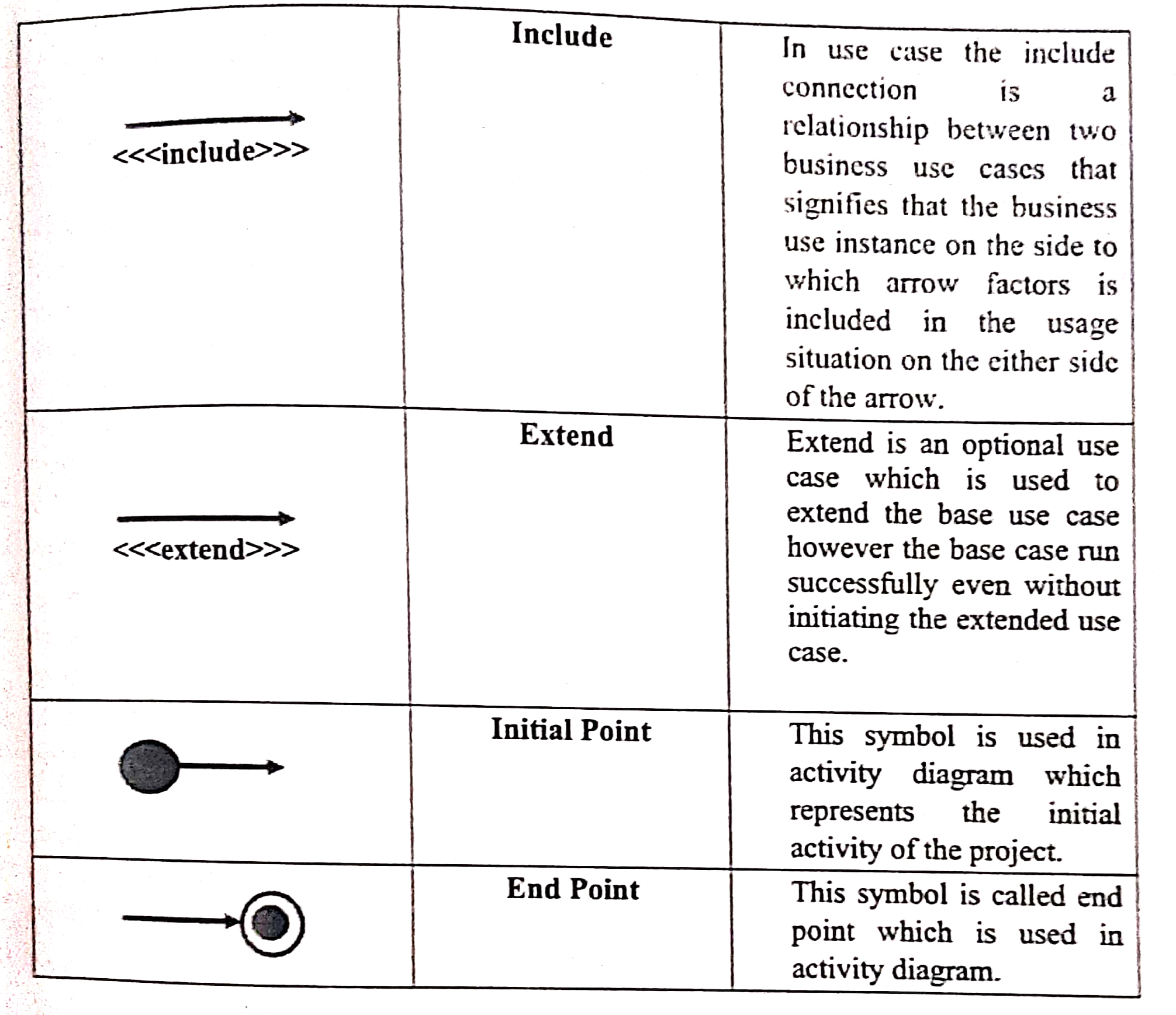
### Project Objectives

Our main objectives are as follows:

* Establishing a simple and efficient platform for teachers to create course outlines and exams without relying on old questions, ensuring fairness in assessments.
* Improving the overall learning experience for students by generating diverse and engaging educational content, addressing the limitations of using outdated materials.
* Storing files on AWS for easy access, addressing the absence of a learning management system and preventing the misuse of old past papers.

## List of symbols and Units





## System Diagram

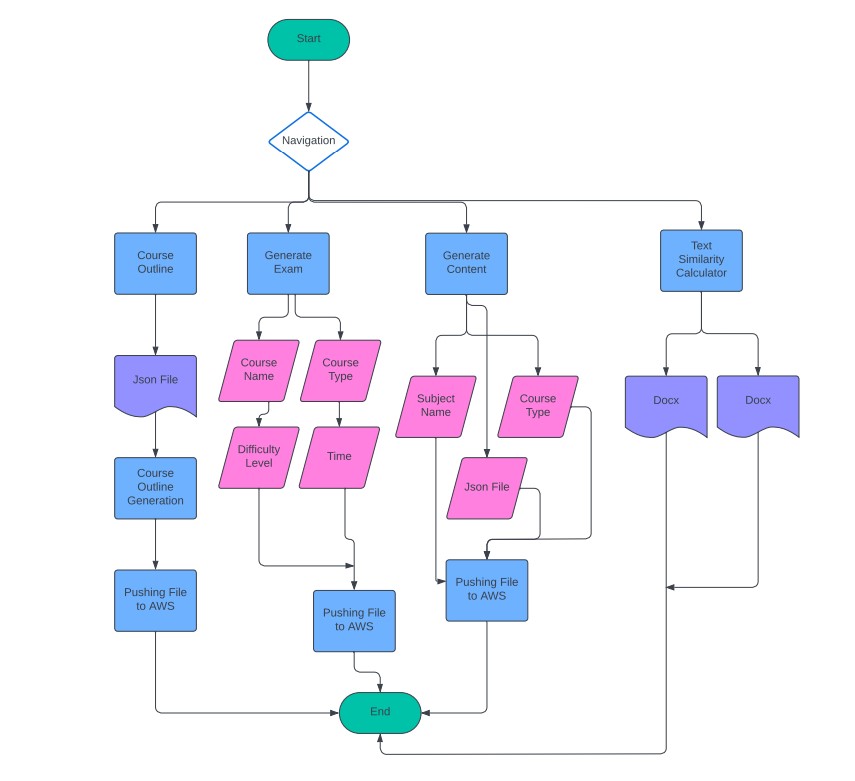


Figure 1.1 System Diagram

CHAPTER 2

BACKGROUND

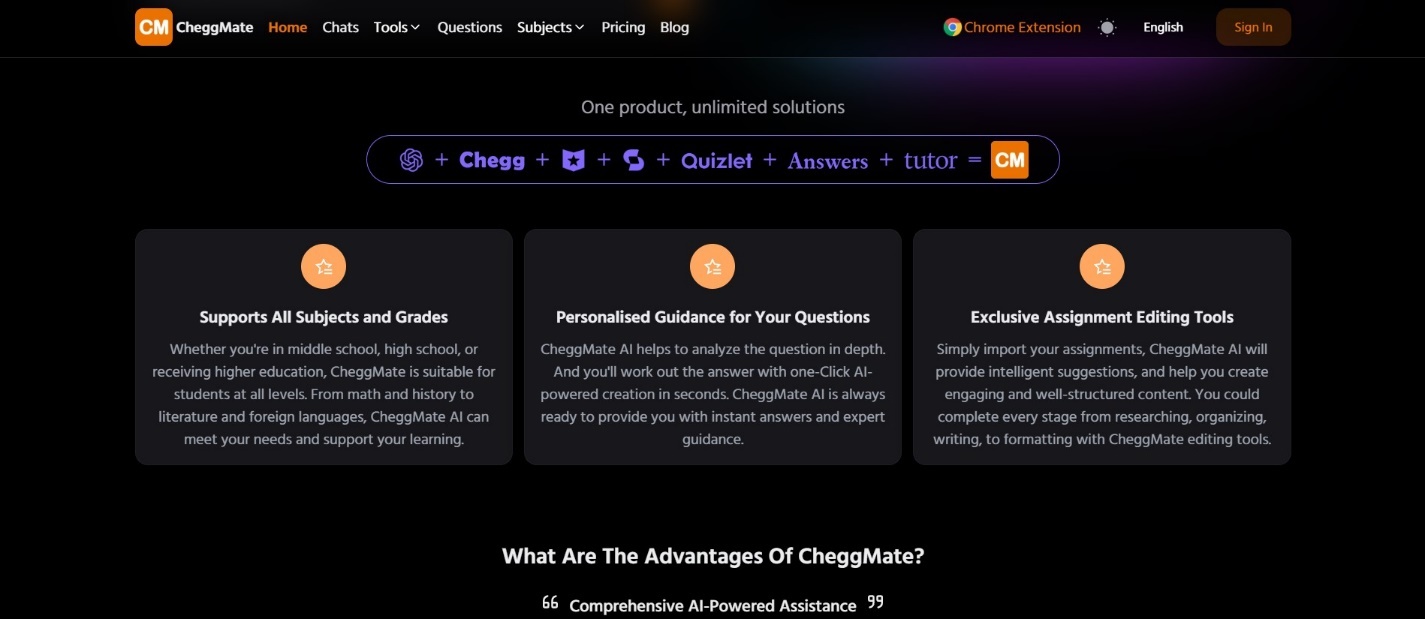
# Background:

In a university where the creation of new exams and accessibility to course materials presented considerable challenges, Academia Assist emerged as a revolutionary solution. The lack of a straightforward method for teachers to produce fresh exams without relying on old questions raised concerns about potential cheating. Additionally, the absence of a system for students to easily access course materials impeded their learning experience. Developed as a user-friendly web app powered by a Large Language Model, Academia Assist addresses these challenges by streamlining the exam creation process, generating compelling teaching content, and preventing the misuse of outdated past papers. Teachers, constrained by time due to research responsibilities, were using antiquated materials, in contrast to top universities exploring cutting-edge research. Academia Assist not only simplifies exam creation but also updates teaching content, ensuring an improved learning experience. Files are securely stored on AWS for convenient access, mitigating the absence of a learning management system. Our objective is to facilitate straightforward studying, eradicate cheating, and empower teachers and students in the continually evolving academic landscape.

## Existing Solution:

### Chegg Mate AI

Chegg Mate AI stands as an innovative AI-powered educational platform, harnessing the capabilities of advanced AI models such as ChatGPT and GPT-3.5/4. This platform aims to personalize learning experiences, aid with homework, and provide extensive study materials. With features like one-click AI solutions, personalized guidance, and round-the-clock expert assistance, Chegg Mate AI ensures tailored support and enhanced learning efficiency. Users gain access to a vast repository of 75+ million verified answers, interactive tutorials, practice exercises, and study guides. The platform seamlessly integrates with popular educational platforms like Chegg, Course Hero, StuDocu, Quizlet, and Tutor.com, offering comprehensive resources. Additionally, Chegg Mate AI prioritizes accessibility, functioning seamlessly across all devices, and provides flexible subscription plans to cater to different needs and budgets. create profiles, upload resumes, and apply for positions that match their skills and qualifications.

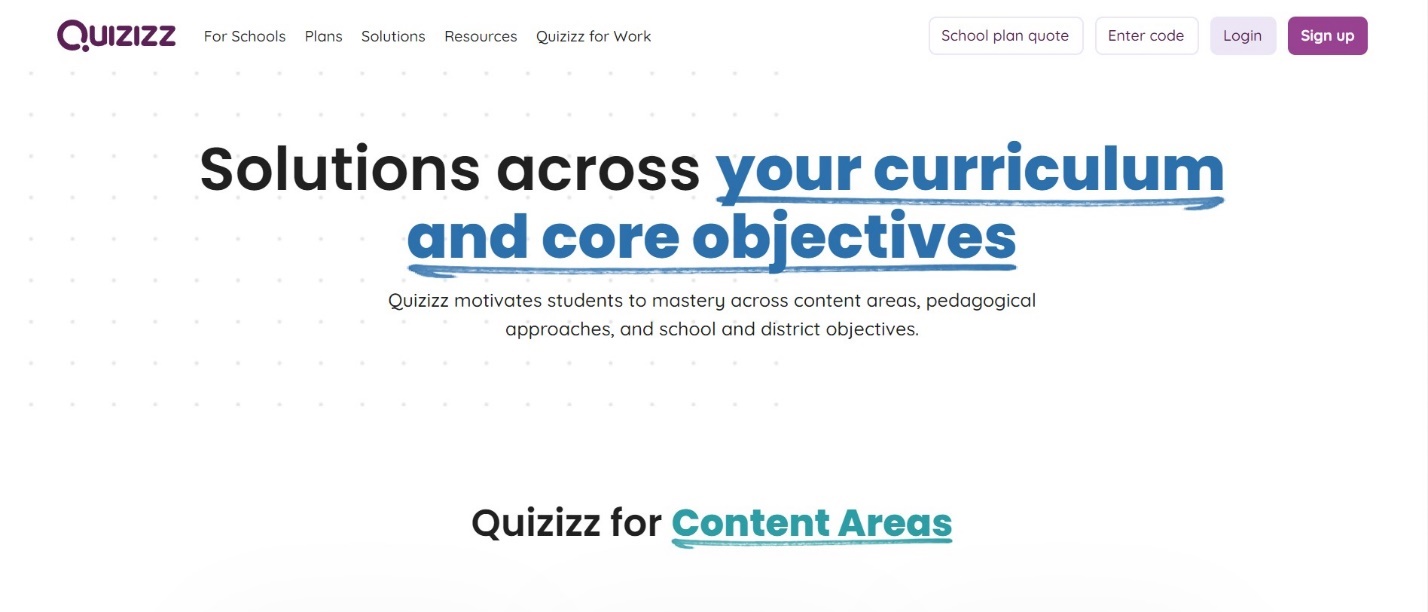


**Limitations:**

* CheggMate AI is not a free service, requiring paid subscriptions, which can be a barrier for some students.
* AI models can perpetuate biases and generate inaccurate information. Students should carefully evaluate the information provided by CheggMate AI and cross-check it with other sources.
* No Course outline and Exam Generation.

### Quizizz

Quizizz stands out as a prominent online platform designed for educators, providing a dynamic and engaging space for creating quizzes, assessments, and games for students. With a focus on making learning enjoyable and effective, Quizizz offers extensive customization options, allowing the creation of quizzes with diverse question types, including multiple choice, true/false, fill-in-the-blank, matching, and open-ended questions, enriched with images, videos, and audio. The platform introduces gamification elements such as points, leaderboards, and timers to instill a competitive spirit and motivate students. Real-time feedback is a notable feature, enabling teachers to monitor students' progress and understanding through reports and live results. Quizizz finds versatile applications, serving as a tool for formative assessments, summative quizzes, pre-class warm-ups, exit tickets, and even gamified presentations. Ensuring accessibility across all devices, offering both free and premium plans, and supporting multiple languages, Quizizz caters to diverse educational needs. However, potential limitations may include a focus on recall-based learning and less emphasis on in-depth understanding of concepts.

****

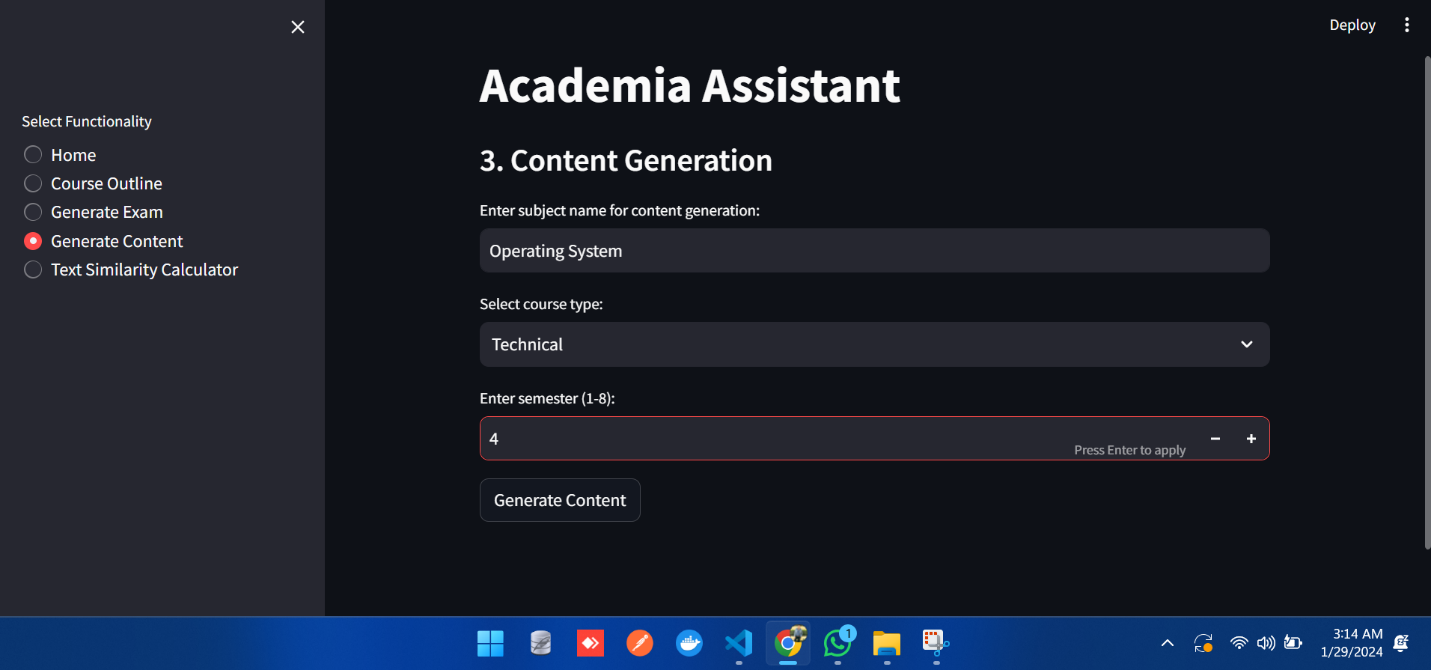
**Limitations:**

* The free plan limits certain features like question types, reports, and class size. Premium plans are required for full functionality.
* While engaging, Quizizz primarily focuses on quiz formats and may not be ideal for deeper learning activities or open-ended discussions.
* Occasional technical glitches with platform functionality can disrupt testing or learning experiences.
* No Generative AI and No Exam and Course Outline Generation.

## Our Solution:

### Academia Assist

Academia Assist is a groundbreaking initiative to address the challenges faced by teachers and students at our university. With the absence of an easy method for creating fresh exams and a system for students to access course materials, we've developed a simple web app leveraging ChatGPT to streamline the exam creation process and enhance learning experiences. Our objectives revolve around making studying more straightforward and eliminating cheating.



**FIGURE 1.2 ACADEMIA ASSIST WEB INTERFACE**

**Limitations**:

* Limited Features
* No Login Functionality
* Limited scope
* Dependency on internet connectivity for platform access

### Comparison Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **CheggMate AI** | **Quizizz** | **Academia Assist** |
| AI Based Features | Yes | No | Yes |
| User-friendly interface for job requests | Yes | Yes | Yes |
| Course Outline Generation | No | No | Yes |
| Exam Generation | Yes | Yes | Yes |
| Course Content Generation | Yes | No | Yes |

CHAPTER 3

AIM & STATEMENT OF PROBLEM

# Aim and Statement of Problem

## 3.1 Problem Statement

In our university, teachers encounter a significant hurdle: the lack of a simple method to create new exams without falling back on old questions, which increases the risk of cheating. Moreover, students face difficulties accessing their course materials easily. To tackle these challenges, we introduced Academia Assist, a user-friendly web app powered by ChatGPT. Our primary focus was on developing three key features: crafting well-formatted course outlines, generating exams, and creating teaching content. We also addressed the issue of repeated content in different courses across various programs. Academia Assist seeks to elevate the learning experience, curb cheating, and simplify the studying process. The absence of a dedicated learning management system in our university, coupled with the misuse of old past papers, necessitated the creation of this solution. Our aim is to empower educators to effortlessly create exams without redundancy, ultimately enhancing the overall learning journey for students. Named Academia Assist, this application securely stores generated files on AWS for convenient access, presenting a transformative solution for academic needs.

## 3.2 Scope of Project

Academia Assist is a groundbreaking initiative designed to transform the educational landscape within our university. The project's scope revolves around simplifying the exam creation process for teachers and enhancing access to course materials for students. By leveraging ChatGPT, the web app focuses on three main components: generating well-structured course outlines, crafting exams with proper formatting, and creating engaging teaching content. The project's overarching goal is to curb cheating and streamline the studying experience for everyone involved. Academia Assist addresses the absence of a learning management system in our university, providing educators and students with an intuitive platform that encourages efficient exam creation and access to educational resources. The scope encompasses the seamless storage of files on AWS, ensuring convenient retrieval and fostering an environment that promotes academic integrity and a more straightforward learning journey. Academia Assist's scope is not just about resolving current challenges but also about establishing a foundation for a more efficient and secure educational experience within our university.

## 3.3 **Solutions for Stakeholders**

**Course Outline and Exam Generation:** Academia Assist offers a streamlined solution for teachers and TA's, empowering them to effortlessly create course outlines and exams without relying on old questions. The application utilizes ChatGPT to generate well-formatted course outlines, ensuring a fresh and engaging learning experience for students.

**Content Generation and Course Similarity:** The platform goes beyond exam creation by generating dynamic teaching content and ensuring the uniqueness of different courses across various programs. Academia Assist addresses the issue of outdated content by providing educators with up-to-date materials, aligning our teaching standards with cutting-edge research prevalent in top universities.

**File Accessibility on AWS:** Academia Assist simplifies file storage and access by securely storing generated course materials on AWS. This feature ensures easy access to files for both educators and students, eliminating the need for a traditional learning management system. The application's user-centric design aims to enhance the overall efficiency of the academic process.

**Preventing Cheating and Enhancing Learning:** Our primary goal is to eliminate cheating by offering educators the tools to create unique exams. Academia Assist fosters a better learning experience by providing students with fresh content, aligning with the ever-evolving landscape of academic knowledge. The platform aims to transform the academic environment, making studying straightforward and promoting academic integrity among stakeholders.

CHAPTER 4

HARDWARE &

SOFTWARE ANALYSIS AND REQUIREMENTS

# 4 Hardware, Software Analysis and Requirements

The following are the software and technologies that would be used.

## 4.1 Languages and Libraries

### Backend Languages and Libraries:

* **Uvicorn (Python):** Uvicorn is an ASGI (Asynchronous Server Gateway Interface) server that serves as a lightning-fast way to run web applications in Python. It is particularly known for its speed and ability to handle asynchronous code efficiently, making it a popular choice for running web frameworks.
* **Prompt Engineering:** Prompt engineering involves crafting effective prompts for language models. It is a crucial aspect of fine-tuning models like GPT (Generative Pre-trained Transformer) to generate desired and contextually relevant outputs based on user input.
* **JSON Parser**: A JSON parser is a tool or module that processes JSON (JavaScript Object Notation) data. It parses JSON strings and converts them into a format that can be easily used and manipulated in a programming language, enabling seamless integration of JSON data into applications.
* **LLM Chain (Language Model Chain):** LLM Chain likely denotes a sequence or chain of language models. This could involve chaining multiple language models together to perform a series of tasks or to enhance the capabilities of a single language model.
* **Pydantic:** Pydantic is a data validation and settings management library for Python. It allows developers to define data schemas using Python type hints, enabling automatic validation, parsing, and documentation generation. Pydantic is commonly used in projects where data integrity and validation are critical.
* **Requests:** Requests is a popular Python library for making HTTP requests. It simplifies the process of sending HTTP requests and handling responses, making it a go-to choose for developers when interacting with APIs or fetching data from web servers.

### Front-end languages and Libraries:

* **Streamlit:** Streamlit is a Python library that enables rapid development of web applications for data science and machine learning. It simplifies the process of creating interactive and shareable web apps with just a few lines of code, making it accessible to both beginners and experienced developers.

## Tools

### 4.2.1 Star UML

Star UML is a software engineering tool for system modeling using the Unified Modeling Language, as well as Systems Modeling Language, and classical modeling notations. We used STARUML for the purpose of creation of UML diagrams of our project.

### DRAW.IO

Draw.io is one of the most well-known online platforms that enables individuals to create professional diagrams and workflow for their projects we used the Draw.io for the purpose of creation of diagram and workflow for our project.

### VS Code

This extension add-on offers AI support for a variety of languages, including React, JavaScript, and Java. It is compatible with a variety of JavaScript frameworks, including Angular, React, and Next. If the language you are writing on supports it, it will automatically complete the procedure or code

### Prettier:

Prettier is a code formatter that helps maintain a consistent code style across the project.

## 4.3 Hardware Requirements

The hardware to be used for the implementation of the project is quite minimal i.e.

We only need a PC (personal computer) the major requirements of a project are just on the software side of things.

## 4.4 Algorithms

## 4.5 Libraries

### 4.5.1 Fast API

Fast API is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. It is designed to be easy to use and to provide automatic interactive documentation, making it a powerful tool for quickly developing robust APIs.

### 4.5.2 Scikit-learn (sklearn):

Scikit-learn is a popular machine learning library in Python. It provides simple and efficient tools for data mining and data analysis, built on NumPy, SciPy, and Matplotlib. Scikit-learn is widely used for tasks such as classification, regression, clustering, and more.

### 4.5.3 Langchain:

LangChain simplifies the integration of Language Model (LLM) services with a user-friendly approach, allowing seamless connection to preferred LLM providers. The platform introduces modular components, such as functions and object classes, serving as fundamental building blocks for common LLM-related tasks. Users can effortlessly chain these modules together to construct personalized application logic. LangChain excels in abstracting technical complexities associated with LLM interaction, enabling developers to focus on enhancing application functionality and user experience. The platform empowers users with customizable prompts, granting complete control over guiding LLM responses. Additionally, LangChain enriches applications by seamlessly integrating with external data sources like Google Search for real-time news and Wikipedia for factual references. Beyond simplicity, LangChain emphasizes contextual understanding, reasoning capabilities, and employs various tools to refine LLM responses, ensuring heightened accuracy and relevance in application outputs.

Top of Form

### 4.5.4 OpenAI's API:

A ChatGPT, driven by OpenAI's GPT-3, harnesses transformers, a specialized neural network architecture, to excel in natural language processing (NLP). The transformer's core lies in its encoder-decoder structure, with the encoder analyzing input sequences and the decoder predicting subsequent words, utilizing a potent self-attention mechanism. Multi-head attention enhances context comprehension by focusing on different aspects. GPT-3, a Large Language Model (LLM), achieves its prowess through vast training datasets and further refinement via fine-tuning. The ChatGPT API acts as a programmatic interface for developers to access the fine-tuned GPT-3, allowing integration into applications with customization options. Technical benefits include efficiency, scalability, and adaptability. However, challenges like bias, explainability, and security require careful consideration for responsible AI use. In conclusion, understanding the technical intricacies empowers developers to responsibly leverage ChatGPT API's capabilities and contribute to ethical AI development.

## Revision

### 4.6.1 Flexibility

The system should be flexible to the modifications that may be required in the future.

### Maintainability

The system should be designed and formed in a way that the fixing and findings of bugs and its maintenance can be done in minimum resources and time.

### Portability

The system should be compatible with the various versions of the Operating Systems.

## 4.7 (Diagrams) Actor Used Case Diagram:

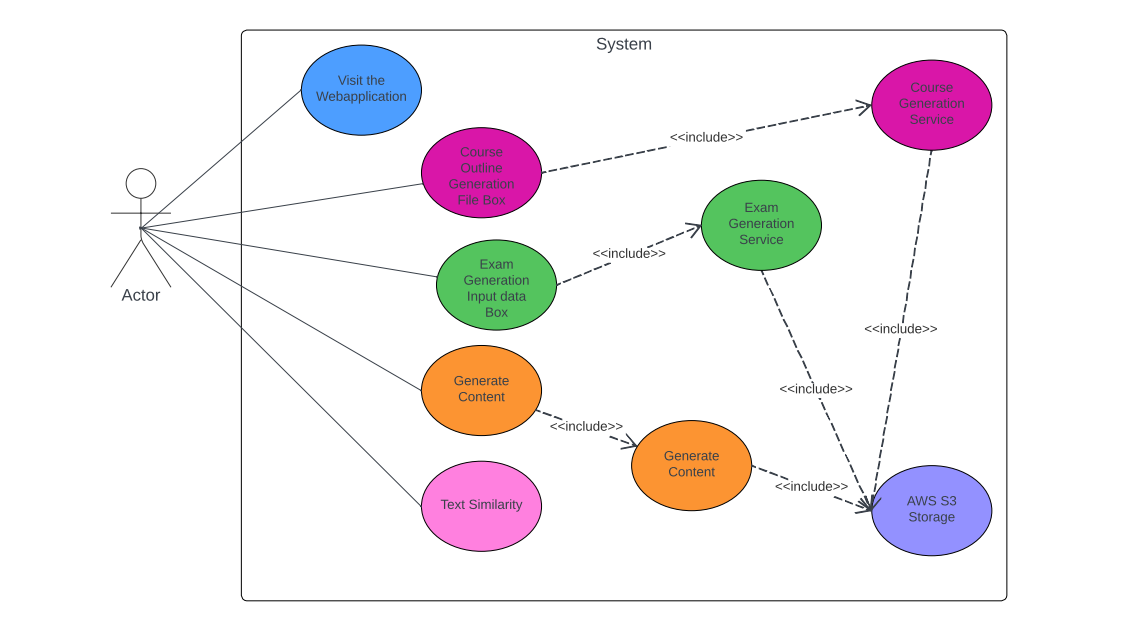


FIGURE 1.3 use case diagram

### Activity Diagram:

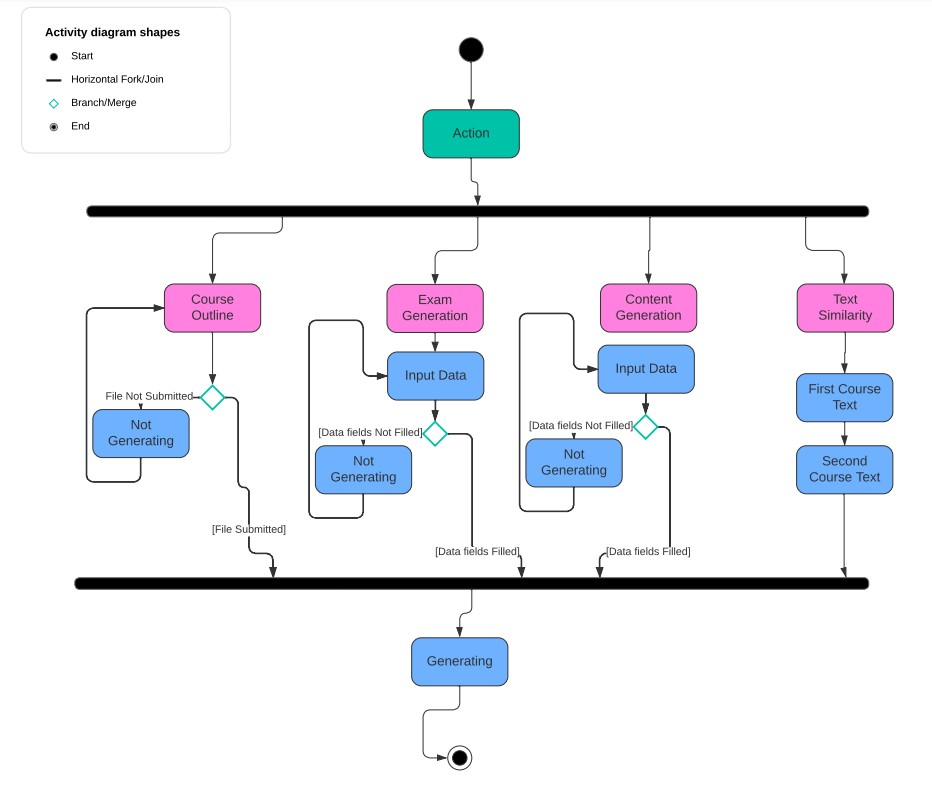


FIGURE 1.4 Activity diagram

CHAPTER 5

SOFTWARE DESIGN AND MODELING

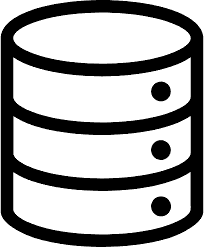
# Software Design and Modeling

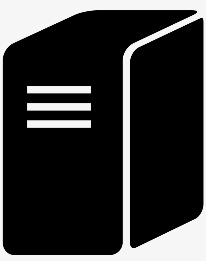
## 5.1 Deployment Architecture

The Below figure is vital when displaying the physical parts which is arranged in the framework. It can be taken as the idea for the hardware component to acknowledge which software elements are deployed by which hardware elements. The below figure displays the concept that which hardware components were used by us to deploy the project, the application is a web-based one where all the processing and functionalities are done and the resources. The desired output is displayed on the desktop screen of the individuals once the processing is done.

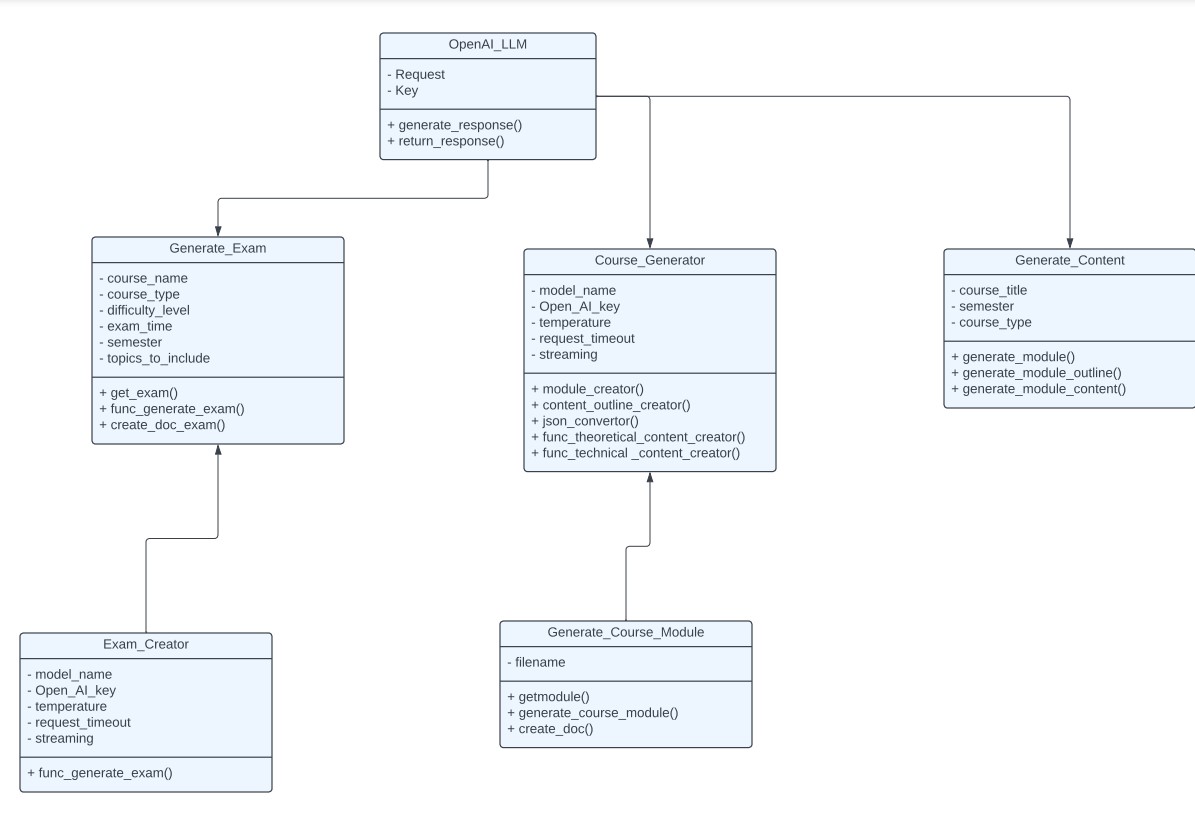






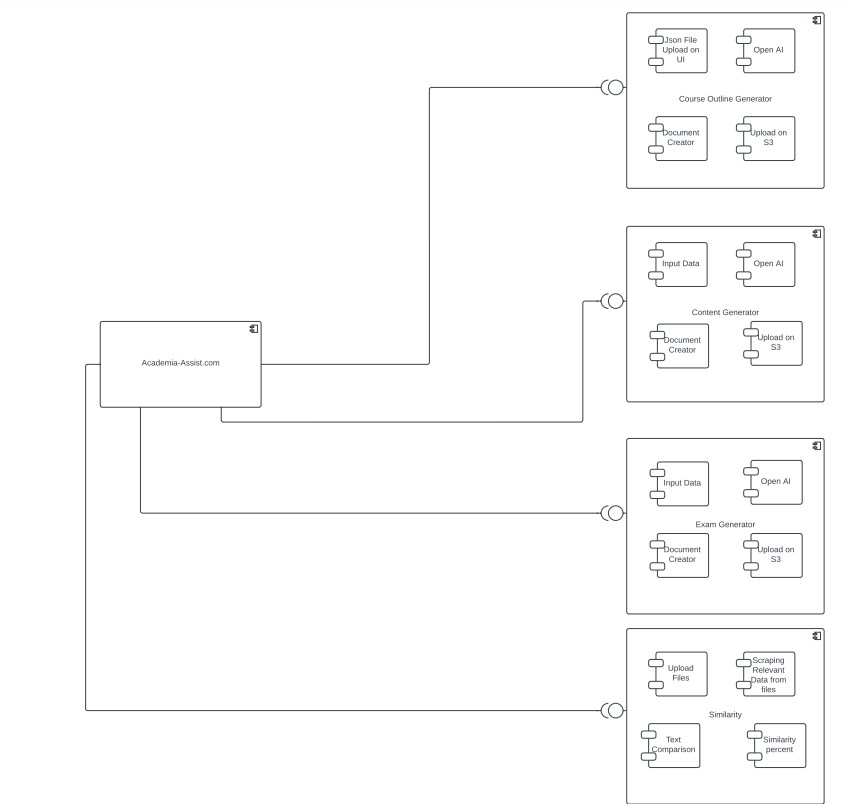
5.1 DEPLOYMENT ARCHITECTURE FIGURE

## 5.2 Class Diagram

****

5.2 Class Diagram

### Component Diagram

****

5.3 COMPONENT DIAGRAM

### State Transition Diagram

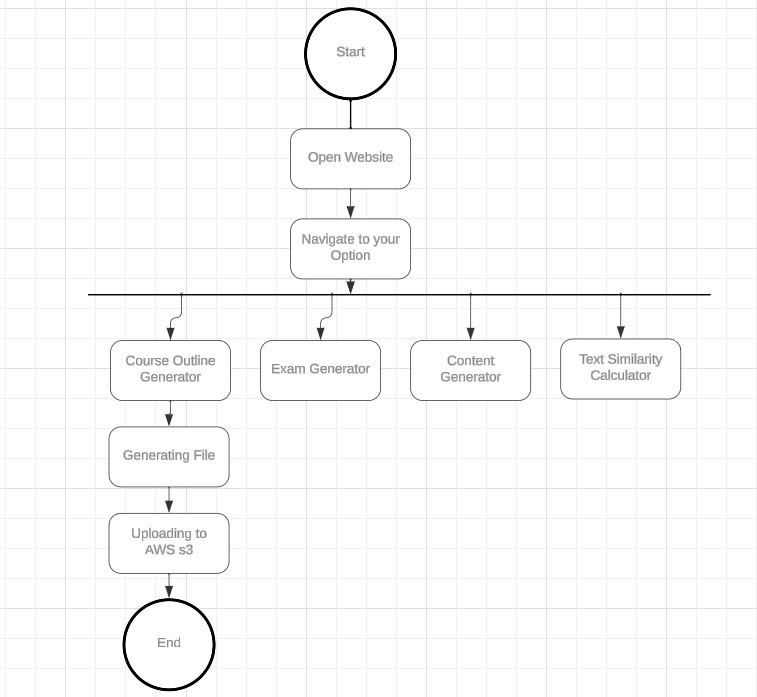


Fig 5.4 state

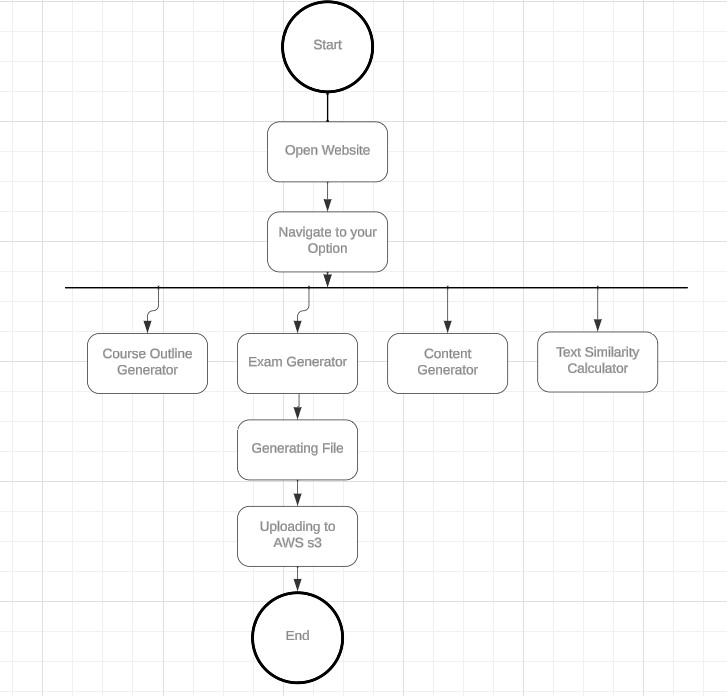
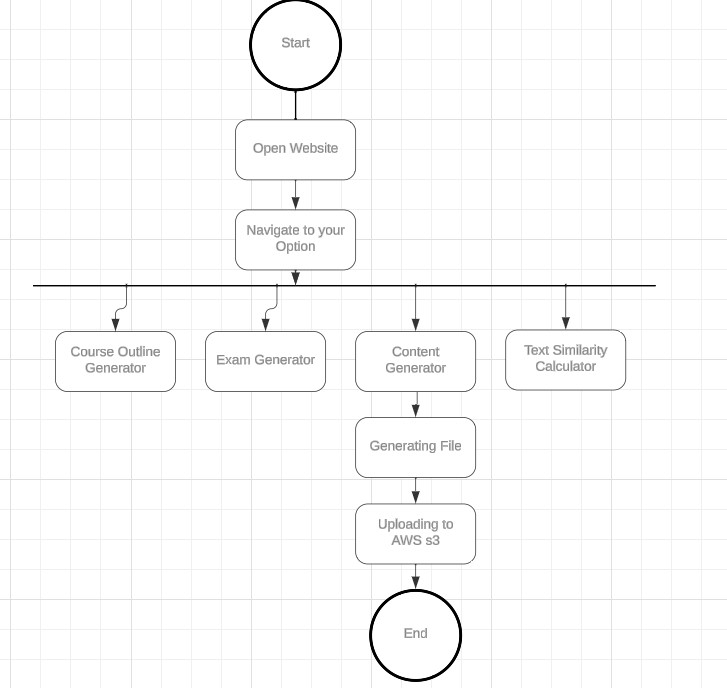
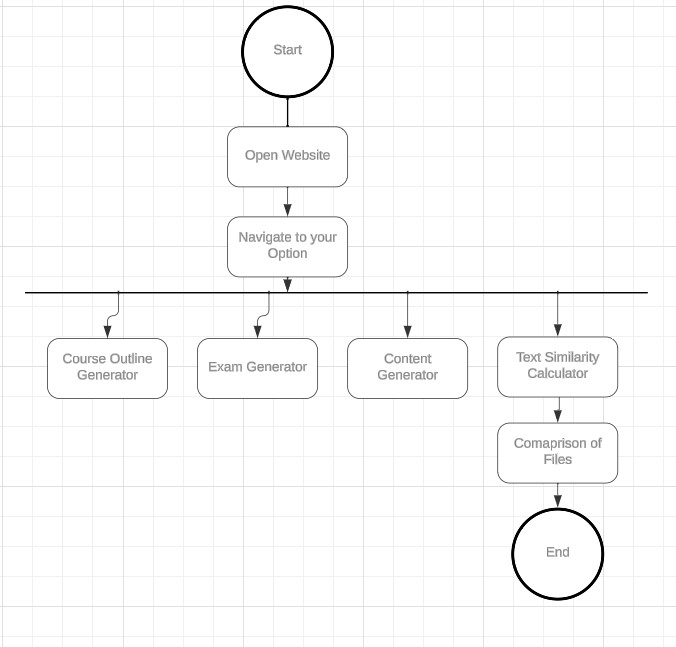


Fig 5.6 state transition diagram for Grammar check

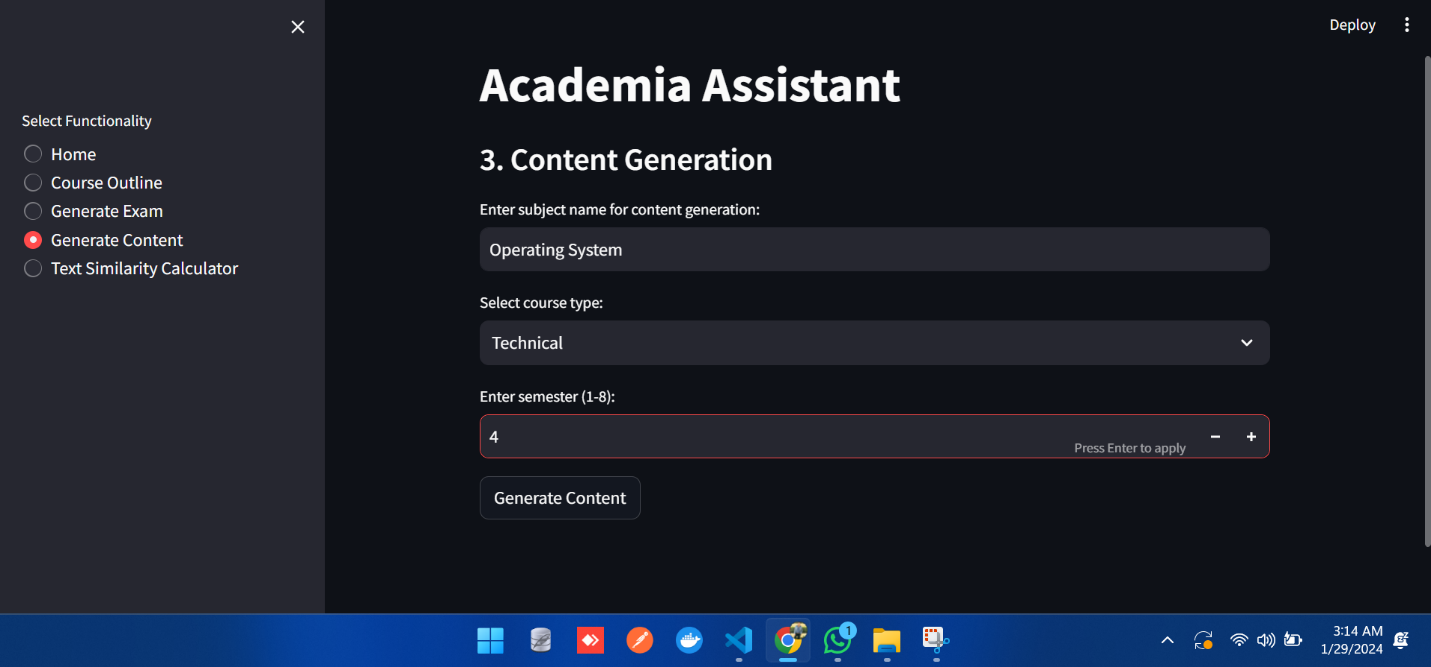




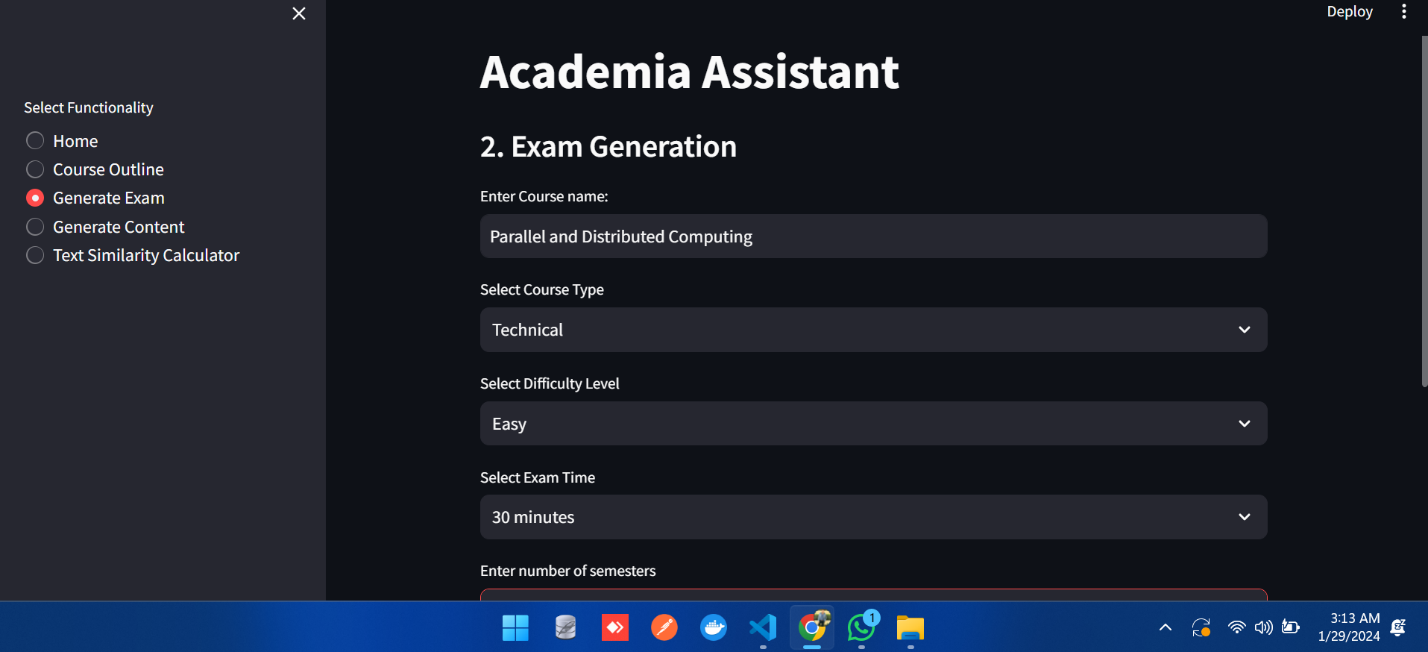
### Prototypes

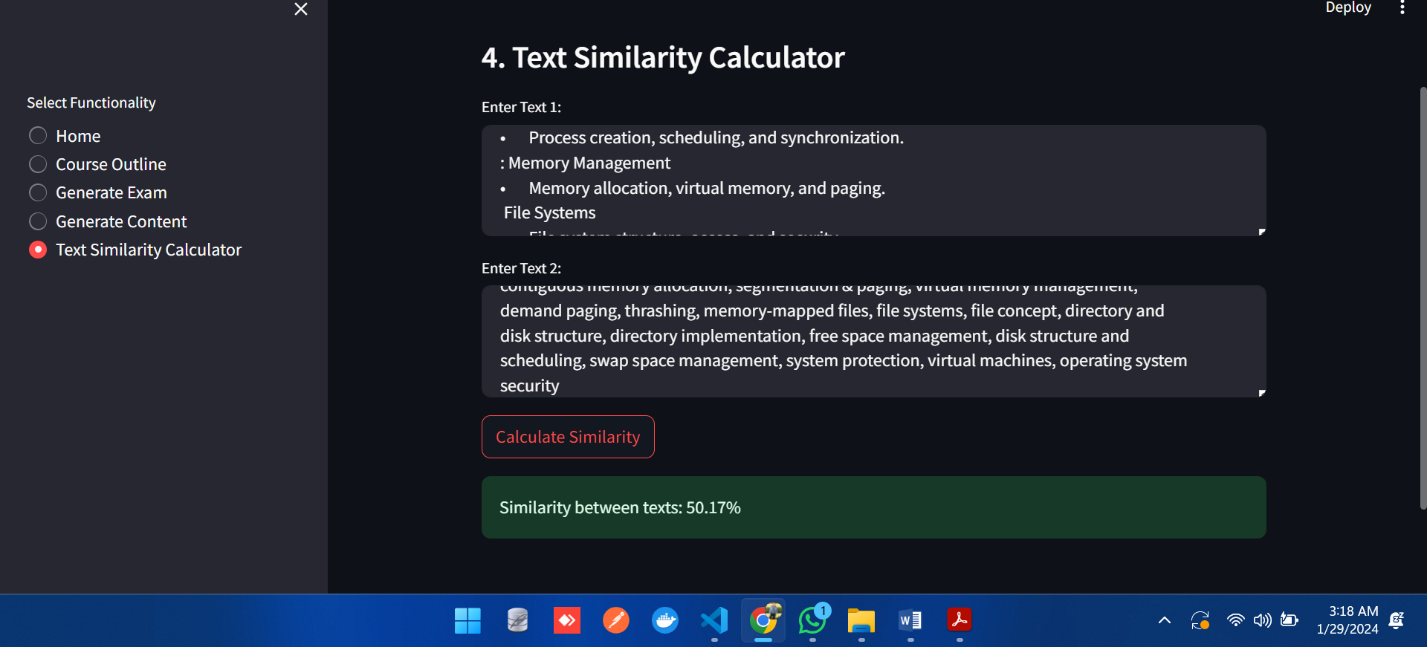
Prototypes are the visual representation of the final product of the website.

The figure Displays the home page of our platform. It is displayed when the user comes to our platform, and the page provides individuals to continue their journey further for desired results.

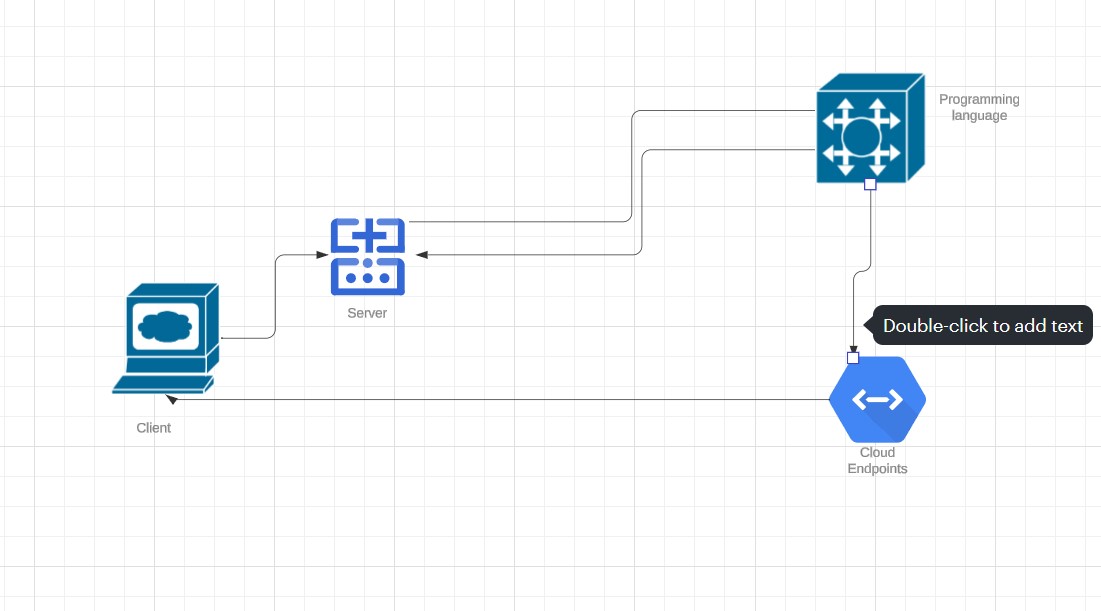
****





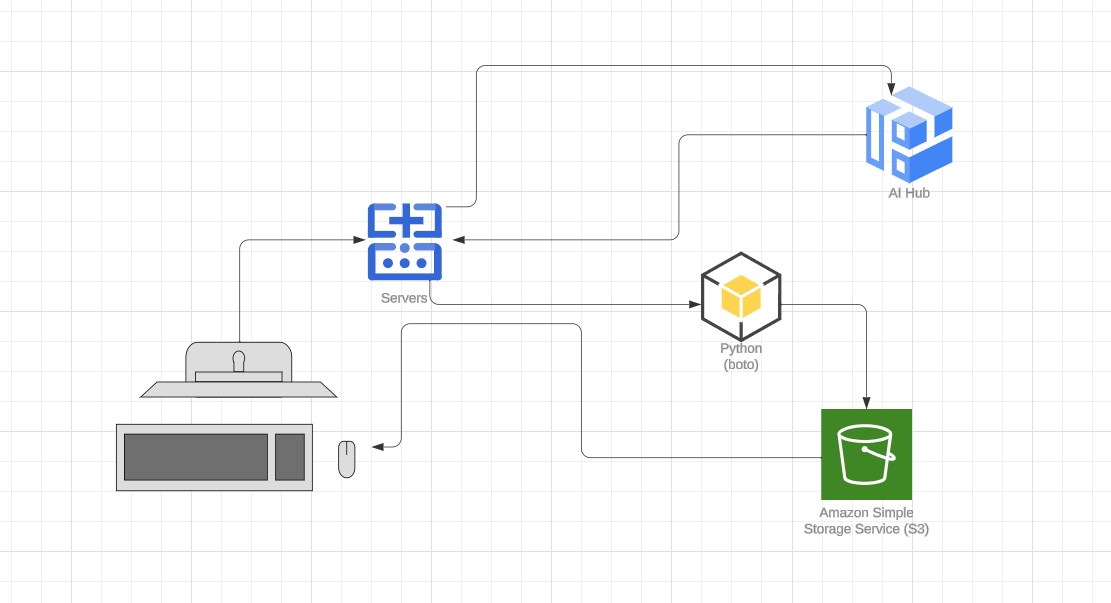


### Operational Diagram



Operational Diagram

### Deployment Diagram



Deployment Diagram

CHAPTER 6

IMPLEMENTATION

# Implementation

The "FastAPI Backend (api.py)" script establishes a robust FastAPI server with three distinct endpoints catering to the generation of course modules, exams, and content. Leveraging the FastAPI framework, the script ensures efficient handling of incoming requests and seamless interaction with core generation functions from the 'services' module. Pydantic models, namely 'ModuleCreatorRequest,' 'ModuleResponse,' 'ExamGeneratorRequest,' and 'ModuleContentCreatorRequest,' facilitate data validation, enhancing the reliability of the API. The server, orchestrated by Uvicorn, runs on http://0.0.0.0:7000/, providing a RESTful interface for course-related operations. This architecture enables a scalable and maintainable solution for managing academic content and assessments through a standardized API. The "Streamlit Frontend (app1.py)" script creates a user-friendly web application using Streamlit. This app interacts with the FastAPI backend to perform various academic tasks. Users can upload a JSON file for course outlines, generate exams, and create content. The script uses the Streamlit library for building the web interface and the requests module to communicate with the FastAPI backend. The main function defines different sections for each functionality, such as uploading course outlines, generating exams, and creating content. Users can navigate through these sections using a sidebar. The user inputs are sent to the FastAPI backend through HTTP requests, and success messages are displayed accordingly. The script is structured to provide a seamless and intuitive experience for managing academic tasks. In simpler terms, it's like a user-friendly website where you can upload files for your courses, generate exams with specific details, and create content for different subjects. The website communicates with another part (backend) to do the heavy lifting and sends you messages when tasks are completed successfully. The "Course Generator (course\_generator.py)" file houses the CourseGenerator class, responsible for leveraging the OpenAI language model to generate diverse components of a course. The class is equipped with functions for creating modules, outlines, content, and converting JSON. It utilizes prompt templates and language model chains to interact with the OpenAI model, allowing users to dynamically generate course elements by providing inputs such as course title, semester, course name, code, and credit hours. The class offers distinct functions for generating theoretical and technical content based on sub-sections and detailed descriptions. This modular approach facilitates the streamlined generation of comprehensive course materials, making it a versatile tool for academic content creation. In simpler terms, it's like a smart assistant that helps teachers or course creators quickly generate different parts of a course, from module outlines to detailed content, using a sophisticated language model. The "Course Content Parsers (parsers.py)" file consists of a single function, parse\_json, which is designed to parse JSON-formatted text and return the parsed data as a dictionary. The function utilizes the json module to attempt parsing, and if successful, it returns the parsed JSON as a dictionary. In case of a JSON decoding error, such as an improperly formatted JSON text, the function catches the json.JSONDecodeError exception, prints an error message, and returns None. This functionality is crucial for handling JSON content within the course generation process, ensuring a smooth conversion of JSON data into a usable Python dictionary. In simpler terms, it's like having a tool that can understand and organize information from JSON files, making it easier for the course generation system to work with structured data. The "Schema Definitions (schema.py)" file plays a crucial role in the FastAPI application by defining Pydantic models for data validation. Pydantic is a library that helps ensure the correctness of data structures in Python applications. In this file, various models are created to represent the structure of data exchanged in the FastAPI application. For instance, `ModuleCreatorRequest` model specifies the format for requests to generate modules from a file, while `ModuleResponse` represents the response containing the generated module. Each class inherits from the `BaseModel` class of Pydantic, allowing for automatic validation of incoming data. In simpler terms, these models act as templates, ensuring that data received or sent by the application follows specific rules, preventing errors and enhancing the reliability of the overall system. The "Services Definitions (services.py)" file acts as a bridge between the FastAPI application and the underlying logic for generating course modules, exams, and content. It imports necessary modules, including `ExamCreator` and `CourseGenerator`, and creates instances of these classes. The file defines functions for document creation, such as `create\_doc` for generating a Word document summarizing course details. Additionally, asynchronous functions like `generate\_module` and `generat\_exam` are defined for efficient concurrent processing of module and exam generation tasks. These functions utilize the core logic from other files, enabling the FastAPI application to interact seamlessly with the services responsible for course-related functionalities. In simpler terms, this file orchestrates the behind-the-scenes operations that fulfill requests made through the FastAPI endpoints, ensuring a smooth and responsive user experience.

### FastAPI Backend

### Streamlit Frontend

### LangChain and ChatGPT

### Prompt Templates and Parsers

### Pydantic Models

### Services and Async Functions

# Testing (Black Box Testing):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC-009 | Performance | Should convert speech to text quickly and efficiently | Non-functional | Converts speech to text effectively | Positive |
| TC-010 | Security | To check if the application is secure and protected. | Non-functional | Secure from illegal activities | Negative |
| TC-011 | Compatibility | To verify if the application is compatible on different browsers | Non-functional | Compatible on all browsers | Negative |
| TC-012 | Maintainability | Test that the application is easy to maintain and update | Non-functional | Application is easy to maintain and update | Positive |
| TC-013 | Performance | To check that the application does not crash or slows down in handling multiple requests | Non-functional | Application does not crash | Positive |
| TC-014 | Reliability | Verify that the application is reliable and available at all times. It should be able to handle errors and exceptions gracefully without losing data or crashing. | Non-functional | Application is reliable and available at all times | Positive |
| TC-015 | Performance | Should convert speech to text quickly and efficiently | Non-functional | Converts speech to text effectively | Positive |

CHAPTER 8

CONCLUSION

# Conclusions

Academia Assist is a game-changer for our university's academic landscape. With a clear mission to combat cheating and simplify the learning experience, we've successfully developed a user-friendly web app that empowers teachers to create new course outlines and exams effortlessly. By harnessing the power of ChatGPT, Academia Assist ensures that exams and course content are generated with proper formatting, reducing the reliance on old questions that may lead to cheating. The application also addresses the lack of a learning management system (LMS) for students by hosting files on AWS, providing easy access to course materials. In essence, Academia Assist is not just a tool; it's a step towards creating a fairer and more straightforward educational environment for both teachers and students. Our commitment to improving academic experience is evident in the platform's features and our dedication to preventing academic dishonesty. As Academia Assist evolves, our focus remains on fostering a better learning journey for everyone involved in academia.

CHAPTER 9

FUTURE WORK

# Future work

In this chapter, we explore the potential avenues for future work that could be undertaken in subsequent study projects to build upon the progress and achievements of this current project. In the coming phases, the project is focused on enhancing user experiences, refining algorithms for better experience, and implementing Context (Vector DB Pinecone). The platform prioritizes safety measures, aims to develop a mobile app for convenient access, and plans community engagement for user education and feedback. Additionally, ongoing research will drive innovation, exploring sustainability initiatives.

CHAPTER 10

APPENDICES

# Appendices

### Appendices A: Executive Summary

### Executive Summary

Our project, Academia Assist, is a groundbreaking initiative to address the challenges faced by teachers and students at our university. With the absence of an easy method for creating fresh exams and a system for students to access course materials, we've developed a simple web app leveraging ChatGPT to streamline the exam creation process and enhance learning experiences. Our objectives revolve around making studying more straightforward and eliminating cheating.

### Appendix B: Project Overview

### Project Overview:

Academia Assist, a revolutionary web application transforming the landscape of academic content creation and examination generation for educators. In the ever-evolving realm of academia, the need for efficient and innovative tools is crucial. Our platform, developed using the ChatGPT API, addresses this need by simplifying the creation of course outlines, exams, and teaching content, enhancing the overall learning experience.

Academia Assist's primary objectives include:

* Formulating course outlines with precise formatting
* Generating exams with proper structure
* Crafting engaging teaching content
* Establishing connections between similar courses across diverse programs

For educators, the platform offers an intuitive interface to effortlessly generate exams, eliminating the repetition of old questions. This approach aims to elevate the quality of learning for students, fostering a cheat-free environment. Furthermore, to overcome the absence of an LMS system in our university, Academia Assist ensures easy student access by hosting generated files on AWS.

The driving force behind Academia Assist is to empower teachers and TAs, providing them with a tool that not only enhances efficiency but also ensures a fair and secure academic environment. Our commitment is to reshape the academic experience, discouraging reliance on outdated materials and fostering originality in teaching and assessment.

With Academia Assist, the future of academia is marked by efficiency, integrity, and accessibility, ensuring a transformative experience for both educators and students alike.

### Appendix C: Project Objectives

Our main objectives are as follows:

* Establishing a simple and efficient platform for teachers to create course outlines and exams without relying on old questions, ensuring fairness in assessments.
* Improving the overall learning experience for students by generating diverse and engaging educational content, addressing the limitations of using outdated materials.
* Storing files on AWS for easy access, addressing the absence of a learning management system and preventing the misuse of old past papers.

### Appendix D: Project Scope

Academia Assist is our solution to help teachers create exams without using old questions, preventing cheating at our university. This simple web app uses ChatGPT to make course outlines, exams, and teaching content easily. We've also added a feature to check if different courses are too similar. Our aim is to stop cheating, make studying straightforward, and provide accessible files on AWS for students and teachers, especially when our university lacks a proper system for this. Our goal is to make learning better for teachers and students.

### In-Scope

**User-Worker Interface Development:** Creating an intuitive platform to facilitate seamless interactions.

* Course Generation
* Exam Generation
* Content Generation

### Out of Scope

The following features and functionalities are considered out of scope for the current project:

**Vector Database Pinecone**: The platform does not perform any context-based storage for that we use Pinecone in the future.

### Appendix E: Deliverables Produced

primary deliverables encompass digital. The project's main point is the development of an intuitive website ensuring seamless accessibility across multiple devices.

### Appendix F: Tools And Technologies

Frontend: Streamlit

Backend: Fast api and other tools

CHAPTER 11

REFERENCES

# 12. References

["Automated Essay Scoring with Knowledge Transfer" by Li et al. (2020):](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/) [[https://arxiv.org/pdf/2401.05655](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)](https://arxiv.org/pdf/2401.05655)

["AI-Driven Exam Generation for Programming Courses" by Ye et al. (2022):](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/) [[https://ispace2023.uic.edu.cn/course/view.php?id=11959](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)](https://ispace2023.uic.edu.cn/course/view.php?id=11959)

*["Can AI Write Good Exams? Challenges and Opportunities of AI-Assisted Assessment" by Davies et al. (2023):](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)* [*[https://lydia-arnold.com/2023/02/15/ai-and-assessment-in-higher-education-reflections/](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)*](https://lydia-arnold.com/2023/02/15/ai-and-assessment-in-higher-education-reflections/)

*["Improving Student Engagement through Adaptive Learning Technologies" by Clark (2018):](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)* [*[https://journals.sagepub.com/doi/10.1177/0734282919836853](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)*](https://journals.sagepub.com/doi/10.1177/0734282919836853)

["Detecting Essay Plagiarism with Deep Learning" by Yang et al. (2020):](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/) [[https://arxiv.org/abs/2103.11909](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)](https://arxiv.org/abs/2103.11909)

*["Promoting Academic Integrity through Technology-Enhanced Assessment" by Rustad et al. (2022):](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)* [*[https://academicintegrity.org/resources/blog/94-2022?start=30](https://www.rozee.pk/blog/2021/04/rozgar-pk-online-job-platform-for-blue-collar-workers/)*](https://academicintegrity.org/resources/blog/94-2022?start=30)