

**A  
PROJECT REPORT**

**ON**

**“Study Buddy ”**

**SUBMITTED TO**



**SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

**FOR THE PARTIAL FULLFILLMENT OF THE DEGREE OF  
MASTER OF COMPUTER APPLICATION (MCA)**

**SUBMITTED BY**

**Wadekar Sakshal Sambhaji**



**PIRENS INSTITUTE OF BUSINESS MANAGEMENT AND  
ADMINISTRATION (IBMA)**

**LONI(BK), TAL-RAHATA, DIST.AHMEDNAGAR.**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**2024-2025**



**SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

**CERTIFICATE**

**This to certify that**

**Wadekar Sakshal Sambhaji**

**Seat Number –10367**

**Student of Second Year of Master of Computer Application (MCA)**

**was examined in Project Report entitled**

**“Study Buddy ”**

**On - / / 2025**

**At**

**PIRENS INSTITUTE OF BUSINESS MANAGEMENT AND  
ADMINISTRATION, (IBMA) LONI (BK), TAL- RAHATA, DIST-  
AHMEDNAGAR.**

**(.....)**

**Internal Examiner**

**(... ..)**

**External Examine**

## ACKNOWLEDGEMENT

I wish to take opportunity to express our profound thanks & gratitude to all those who have directly or indirectly helped us in the completion of this project.

I am also extremely thankful grateful to our project guide **Prof. P. C. Gorde** inspiration and constructive suggestions that helpful us in the preparation of this project. We also thanks to our **Prof. R. S. Tanpure , Academic Coordinator of MCA** who have helped in successful completion and certification of the project.

# Index

Chapter No		Details	Page no
<b>1</b>		<b>Introduction</b>	<b>1</b>
	1.1	Company Profile / Institute Profile / Client Profile	2-3
	1.2	Abstract	4
	1.3	Existing System and Need for System	5-6
	1.4	Scope of System	7
	1.5	Operating Environment - Hardware and Software	8-9
	1.6	Brief Description of Technology Used	10
<b>2</b>		<b>Proposed System</b>	<b>11</b>
	2.1	Feasibility Study	12-13
	2.2	Objectives of Proposed System	14
	2.3	Users of System	15
<b>3</b>		<b>Analysis and Design</b>	<b>16</b>
	3.1	Entity Relationship Diagram(ERD)	16
	3.2	Table Structure	17-20
	3.3	Class Diagram	21
	3.4	Use Case Diagram	22
	3.5	Activity Diagram	23
	3.6	Sample Input and Output Screens	24-32
<b>4</b>		<b>Coding</b>	<b>33</b>
	4.1	Code snippets	33-37
<b>5</b>		<b>Testing</b>	<b>38</b>
	5.1	Test Strategy	38
	5.2	Unit Test Plan	39
	5.3	Acceptance Test Plan	40-41
	5.4	Test Case	42-43
<b>6</b>		<b>Drawbacks &amp; Limitations</b>	<b>44-45</b>
<b>7</b>		<b>Proposed Enhancements</b>	<b>46</b>
<b>8</b>		<b>Conclusion</b>	<b>47</b>
<b>9</b>		<b>Bibliography</b>	<b>48</b>

## **Chapter 1**

### **INTRODUCTION**

Study Buddy is a dynamic and feature-rich Learning Management System designed to revolutionize online education. In an era where digital learning is rapidly evolving, Study Buddy aims to create an engaging, accessible, and efficient platform for both educators and learners. This system empowers instructors using Study Buddy to deliver high-quality educational content, while providing students with a seamless and interactive Study Buddy learning experience. Study Buddy is built with a user-centric approach, offering a diverse array of features to facilitate effective online learning, including robust user management, comprehensive course creation and management, advanced learning and progress tracking, engaging communication and collaboration features, intelligent AI-powered tools, and gamification and monetization options. Study Buddy's technical foundation leverages modern technologies to ensure a scalable and reliable platform. The goal of Study Buddy is to provide a valuable tool for anyone involved in online education, fostering a more effective and engaging learning environment.

## **1.1 Company Profile**

### **Creazione Software**

#### **Profile Overview**

1. About Us
2. Vision Statement
3. Mission Statement
4. Services offered

#### **1. About Us**

**Creazione Software** we develop individuals by providing industry-based training to launch their careers. We have been focusing on creating more good tech leaders for the industry within our curriculum itself.

Additionally, we provide training on, in-demand technologies like as Java, and Python. we have an experienced trainer's team with real-world industry experience

Moreover, we offer guaranteed placement support to our students.

As our title line says, "Talent meets opportunity", we are giving our best to live up to it providing greater opportunities for students to uplift their lives

#### **2. Vision Statement**

- To provide candidates with Technical knowledge that is at-par with the IT industry, but also in a pocket friendly way so that everyone can take benefit of our courses.
- To provide training to candidates in various technologies by experts who have first-hand industry knowledge so that candidates can start their career and be in sync with the industry.
- To bridge expectation gap between the corporate sector and candidates.
- To provide corporate training to the candidates as per their requirements.

### **3. Mission Statement**

To bridge the gap between candidates and the corporate / IT sector by creating skilled resources who can build their career in the industry.

### **4. Services Offered**

#### **Web Development:**

Web development is the work involved in developing a Web site for the Internet. We believe in designing a user-friendly website to beat the industry standard. Our corporate web design service has designed to suit all your custom web design services. **Creazione Software** is also guaranteeing top results in the search engine ranking of your website with the help of our website development team.

#### **Software Development:**

Our software development company is a result of various innovations and enhancements. Our developed product fulfills our all-client requirements. Also, we follow global standards of software engineering. Software development refers to a set of computer science activities dedicated to the process of creating, designing, delivering, and supporting software.

## 1.2 ABSTRACT

Study Buddy is a modern Learning Management System designed to bridge the gap between traditional education and the evolving landscape of online learning. It provides a comprehensive platform for instructors to create and manage engaging courses, while offering students a seamless and interactive learning experience. Leveraging a user-centric design, Study Buddy incorporates features such as robust user management, detailed progress tracking, interactive quizzes, automated certificate generation, and integrated communication tools. Beyond basic functionality, Study Buddy integrates AI-powered features like personalized course recommendations and an AI chatbot for instant support, enhancing the learning journey. Furthermore, it incorporates gamification elements and monetization options, fostering a dynamic and sustainable educational environment. Built on a scalable and reliable tech stack, Study Buddy aims to empower educators and learners alike, fostering a more effective and accessible online education experience.



## **1.3 EXISTING SYSTEM & NEED FOR SYSTEM**

### **Existing Systems**

The online learning landscape is populated with a variety of Learning Management Systems (LMS), each catering to different needs and user bases. These systems range from open-source platforms to proprietary software, and they serve diverse educational sectors, including academic institutions, corporate training, and individual course creators.

Commonly encountered existing LMS systems include:

- Moodle: A widely used open-source LMS known for its flexibility and customization options, often favored by educational institutions.
  - Canvas: A popular cloud-based LMS that emphasizes user-friendly interfaces and robust course management tools, prevalent in higher education.
  - Blackboard: A comprehensive LMS with a long history, offering a wide range of features for course delivery, assessment, and communication, commonly used in academic and corporate settings.
  - Google Classroom: A simplified LMS integrated with Google Workspace, designed for ease of use and seamless collaboration, particularly popular in K-12 education.
  - Teachable/Thinkific: Platforms focused on enabling individual course creators to build and sell online courses, emphasizing marketing and monetization tools.
  - LMS solutions provided by companies like Adobe Captivate Prime, SAP Litmos, and others
- These types of LMS are normally geared towards business and corporate training.

## **Need for System:**

Despite the presence of various existing Learning Management Systems, a significant gap remains in providing a truly engaging, personalized, and intelligent online learning experience. The need for Study Buddy stems from several critical shortcomings observed in current solutions.

Firstly, many existing platforms struggle to foster active student engagement. They often rely on passive content delivery, lacking interactive elements like gamification, personalized quizzes, and robust communication tools. This leads to decreased student motivation and a less effective learning process.

Secondly, personalization is often overlooked. Generic course delivery fails to cater to diverse learning styles and individual student needs. The ability to provide tailored learning paths, AI-powered recommendations, and adaptive assessments is crucial for maximizing learning outcomes.

Thirdly, the integration of emerging technologies, particularly AI, is underutilized. AI-powered features like automated support, plagiarism detection, and personalized learning recommendations can significantly enhance the learning experience, yet they are often absent or poorly implemented in existing systems.

Additionally, user interfaces are often complex and unintuitive, creating barriers for both instructors and students. A user-friendly and streamlined platform is essential for seamless learning.

Furthermore, instructors seeking to monetize their expertise encounter limitations. Many platforms offer restrictive or complex monetization options, hindering their ability to create sustainable online courses.

## 1.4 SCOPE OF THE SYSTEM:

The scope of the Study Buddy encompasses the development and deployment of a comprehensive online learning platform designed to facilitate effective and engaging education. This system aims to provide a robust and user-friendly environment for both instructors and students, addressing the diverse needs of online learning.

Specifically, the scope of Study Buddy includes:

- User Management:
  - Implementation of a secure and scalable user authentication and authorization system, supporting distinct roles for administrators, instructors, and students.
  - Management of user profiles, including registration, login, and profile editing.
- Course Management:
  - Creation and management of online courses, including modules, lessons, and assignments.
  - Support for diverse content formats, such as videos, PDFs, text documents, and interactive quizzes.
  - Enrollment and management of students in courses.
- Learning and Progress Tracking:
  - Tracking of student progress through lessons and assignments.
  - Implementation of interactive quizzes with automated grading.
  - Generation of course completion certificates.
  - Visual progress indicators.
- Communication and Collaboration:
  - Integration of discussion forums and chat systems for student-instructor interaction.
  - Integration with third-party live class platforms (e.g., Zoom, Google Meet).
  - Notification system for announcements, assignments, and events.
- Intelligent Features:
  - Implementation of AI-powered course recommendations based on student learning history.

## **1.5 OPERATING ENVIRONMENT:**

### **Software Requirements:**

- Operating System:
- Server-Side: Linux-based OS (e.g., Ubuntu, CentOS) or Windows Server (preferred: Linux).
- Client-Side: Modern Web Browsers (Chrome, Firefox, Safari, Edge) compatible with HTML5, CSS3, and JavaScript (ES6+).

### **Database Management:**

- PostgreSQL: For robust data storage and management.

### **Development Tools:**

- Integrated Development Environment (IDE):
- Backend: PyCharm, VS Code (with Python extensions).
- Frontend: VS Code, or IDE specific to the frontend framework chosen (e.g., if React, VS Code with React extensions).
- Programming Languages:
- Backend: Python (Django/Django Rest Framework).
- Frontend: HTML5, CSS3, JavaScript (React.js or Django Templates).

### **Web Server:**

- Nginx or Apache with WSGI (Gunicorn or uWSGI).

### **Authentication:**

- Django's built in authentication, or JWT.
- Third party API's
- API's related to Zoom/Google Meet, Stripe/Paypal, and any other external services.

## **Hardware Requirements:**

### **Server-Side Hardware:**

- Processor: Intel Core i5 or equivalent.
- RAM: 8 GB minimum.
- Storage: 250 GB SSD minimum (for OS and application), 1 TB HDD or equivalent (for data storage and backups).
- Network: Stable internet connection for cloud services or server hosting.

### **Client-Side Hardware:**

- Processor: Any modern processor capable of running modern web browsers smoothly.
- RAM: 4 GB minimum.
- Storage: Sufficient storage for browser cache and user data.
- Network: Stable internet connection for accessing the LMS.

## 1.6 Brief Description of Technology Used

### 1. Frontend(UserInterface):HTML,CSS,JavaScript

These are used to design and style the web pages. HTML structures the content, CSS styles the layout and appearance, and JavaScript adds interactivity such as form validations, dynamic updates, and responsive behavior.

### 2. Backend(Server-side Logic):Python with Flask/Django (*whichever you used*)

The backend is responsible for processing user requests, interacting with the database, and handling core logic such as user authentication, assignment submissions, and attendance tracking.

### 3. Database:MySQL/PostgreSQL/SQLite(mention what you used)

A relational database is used to store and manage data including users, courses, assignments, attendance records, and quizzes. SQL queries are used to retrieve and manipulate this data.

### 4. ORM (Object Relational Mapping): SQLAlchemy / Django ORM

ORM allows seamless communication between Python code and the database using Python classes instead of direct SQL statements, improving code readability and efficiency.

### 5. Authentication & Security: Password Hashing (e.g., bcrypt or Django's in-built tools)

User passwords are securely stored using hashing algorithms to protect sensitive information. Role-based access control ensures that different users (admin, teacher, student) have appropriate access.

### 6. FileStorage:LocalServerStorage/CloudStorage(optional)

Used to store profile pictures, uploaded assignments, and other documents linked to courses and students.

### 7. Framework (if used):

- Flask: A lightweight framework for handling routes, APIs, and backend logic.
- Django: A full-featured web framework with built-in admin panel, authentication system, and ORM.

## **Chapter No 2. PROPOSED SYSTEM**

Study Buddy is proposed as a comprehensive online learning platform designed to revolutionize the educational experience. It aims to address the limitations of existing Learning Management Systems by offering a more engaging, personalized, and efficient environment for both educators and learners. This system prioritizes a user-centric design, ensuring intuitive navigation and tailored functionalities for distinct user roles, including administrators, instructors, and students. Study Buddy will provide robust course management capabilities, enabling instructors to easily create and manage courses, modules, and lessons, while supporting diverse content formats. The learning experience will be enhanced through features like progress tracking with visual indicators, interactive quizzes with automated grading, and automated certificate generation. Integrated communication and collaboration tools, such as discussion forums, real-time chat, and seamless live class platform integration, will foster a vibrant learning community. Study Buddy will leverage intelligent features, including AI-powered course recommendations, an AI chatbot for instant support, and assignment plagiarism detection, to further enhance the learning process. Gamification elements and integrated payment gateways will contribute to increased engagement and monetization opportunities. Built on a scalable and reliable architecture, utilizing Django and Django Rest Framework for the backend, React.js or Django Templates for the frontend, and PostgreSQL for robust data management, Study Buddy aims to provide a modern and reliable online learning environment. The overarching objective is to create a holistic learning ecosystem that empowers educators and learners, ultimately redefining the online education landscape.

## 2.1 Feasibility Study

### a) Technical Feasibility:

- **Assessment:**
  - The technical feasibility of Study Buddy is high. The proposed technology stack (Django/Django Rest Framework, React.js or Django Templates, PostgreSQL) is well-established and widely used for web application development.
  - Cloud-based deployment options (AWS, Azure, Google Cloud) provide scalable and reliable infrastructure.
  - API integrations for third-party services (Zoom, Google Meet, Stripe, PayPal) are readily available and well-documented.
  - AI and machine learning components can be implemented using available libraries and frameworks (e.g., TensorFlow, scikit-learn).
- **Evaluation:**
  - The development team possesses the necessary skills and expertise to implement the proposed technologies.
  - The technology stack is scalable and can handle anticipated user traffic.
  - Security measures can be implemented to protect user data and prevent vulnerabilities.
- **Conclusion:**
  - Study Buddy is technically feasible. The proposed technologies and infrastructure are suitable for building a robust and scalable online learning platform.

### b) Economic Feasibility:

- **Assessment:**
  - The economic feasibility of Study Buddy depends on the cost of development, deployment, and maintenance, as well as the potential revenue generated from course subscriptions, premium features, and instructor commissions.
  - Market research indicates a growing demand for online learning platforms, suggesting a strong potential for revenue generation.
  - Cost analysis of hardware, software, cloud services, and personnel.



- **Evaluation:**

- A detailed financial model should be developed to estimate the return on investment (ROI) and breakeven point.
- Pricing strategies should be developed to attract users while ensuring profitability.
- Explore potential sources of funding, such as venture capital, grants, or bootstrapping.

- **Conclusion:**

- Study Buddy is economically feasible, provided that a sound financial model is developed and implemented. The potential for revenue generation is significant, given the growing demand for online learning.

### c) Operational Feasibility:

- **Assessment:**

- The operational feasibility of Study Buddy depends on the ability to effectively manage and maintain the platform, as well as the acceptance and adoption of the system by users.
- Training and support should be provided to instructors and students to ensure smooth onboarding and utilization of the platform.
- The platform must be easily maintainable, and updates must be able to be rolled out smoothly.

- **Evaluation:**

- A clear operational plan should be developed, outlining roles and responsibilities, support processes, and maintenance procedures.
- User feedback should be collected and incorporated into ongoing improvements.

## **2.2 Objectives of the Proposed System**

The Study Buddy system is designed with several key objectives to create a transformative online learning experience. These objectives are:

### **1. Enhance Accessibility and Flexibility:**

- To provide a platform that enables learning anytime, anywhere, catering to diverse learner needs and schedules.
- To break down geographical barriers to education, making quality learning resources available to a wider audience.

### **2. Improve Engagement and Motivation:**

- To create an interactive and engaging learning environment that fosters active participation and knowledge retention.
- To utilize gamification, personalized learning paths, and interactive content to increase student motivation.

### **3. Facilitate Effective Course Management:**

- To empower instructors with intuitive tools for creating, managing, and delivering high-quality online courses.
- To streamline administrative tasks, allowing instructors to focus on teaching and student engagement.

### **4. Promote Personalized Learning:**

- To leverage AI and data analytics to provide personalized learning experiences tailored to individual student needs and learning styles.
- To offer adaptive assessments and personalized course recommendations.

### **5. Foster Collaboration and Communication:**

- To create a vibrant learning community through integrated communication tools, such as discussion forums and real-time chat.
- To facilitate seamless interaction between instructors and students.

### **6. Ensure Scalability and Reliability:**

- To build a robust and scalable platform that can handle increasing user traffic and data volumes.
- To provide a reliable and secure learning environment

## 2.3 USER OF THE SYSTEM

The Study Buddy is designed to serve a diverse range of users, each with specific needs and roles within the platform. These users can be broadly categorized as follows:

### 1. Administrators:

- Administrators are responsible for the overall management and maintenance of the Study Buddy.
- They have full control over user accounts, course management, system settings, and data security.
- Their primary tasks include user management, system configuration, report generation, and ensuring the smooth operation of the platform.

### 2. Instructors/Teachers:

- Instructors are the content creators and facilitators of online courses.
- They use the platform to create and manage course content, upload materials, assign quizzes and assignments, and track student progress.
- They interact with students through discussion forums, chat systems, and live classes.
- They are also able to monetize their courses through the platform.

### 3. Students/Learners:

- Students are the primary users of the learning content.
- They enroll in courses, access learning materials, take quizzes and assignments, participate in discussions, and track their progress.
- They benefit from personalized learning experiences, AI-powered recommendations, and instant support.

### 4. Technical Support Staff:

- These users are responsible for the technical upkeep of the system.
- They will trouble shoot user problems, maintain the servers, and update the software as needed.

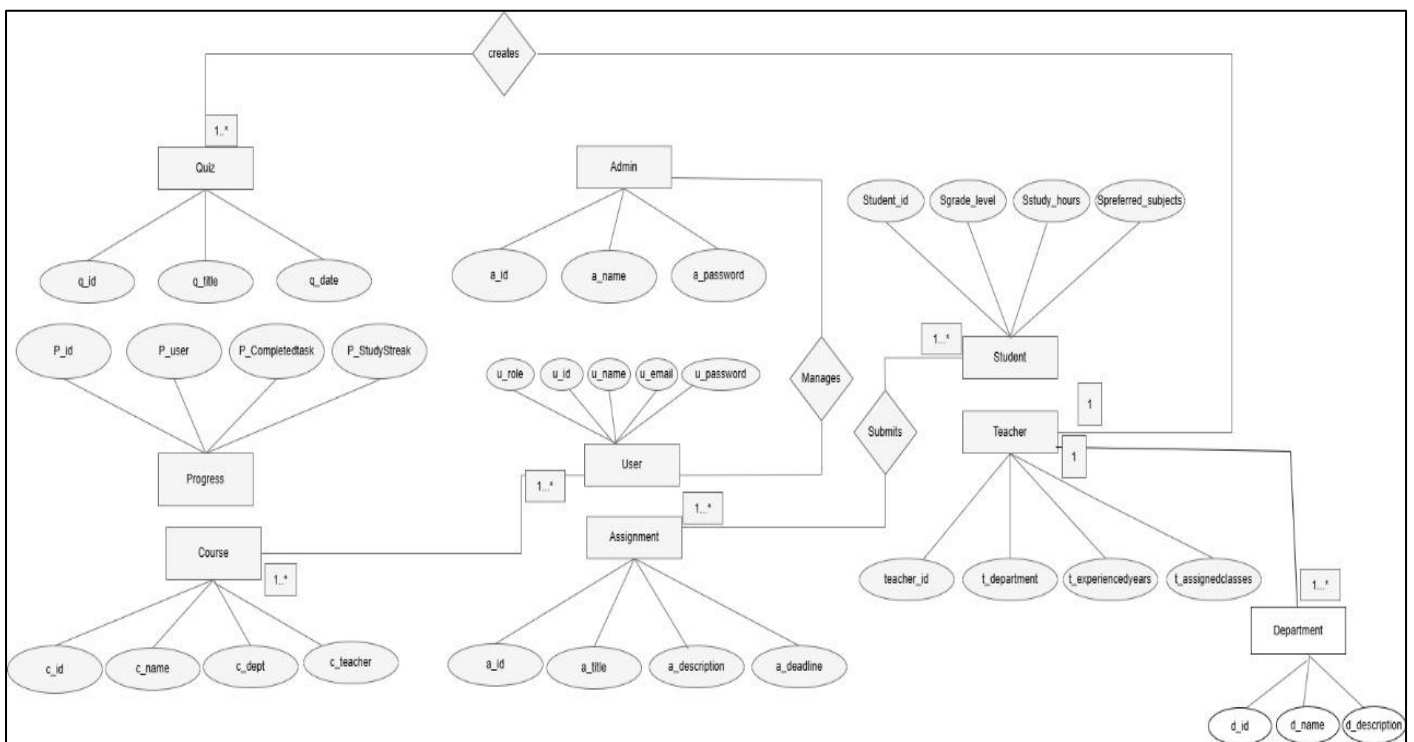
### 5. Potential Course Creators/Entrepreneurs:

- Individuals who wish to create and sell online courses.
- They utilize the monetization features of Study Buddy to generate revenue from their educational content.

## Chapter No 3. ANALYSIS AND DESIGN

### 3.1 ER Diagram:

An entity relationship diagram, also known as an entity relationship model , is a graphical representation that depicts relationships among people , objects, places, within an information technology system . An entity relationship diagram shows the relationship of entity sets stored database.



### 3.2 Table Structure:

#### Users Table:

Sr.no	Fieldname	Datatype	Fieldsize	Constraints	Description
1	user_id	INT	10	Primary Key	Unique identifier for each user
2	username	VARCHAR	255	Unique	User's chosen username
3	email	VARCHAR	255	Unique	User's email address
4	password_hash	VARCHAR	255		Hashed password for security
5	first_name	VARCHAR	255		User's first name
6	last_name	VARCHAR	255		User's last name
7	role	VARCHAR	50		User's role(admin, teacher, student)
8	course	VARCHAR	50		Select Course
9	registration_date	TIMESTAMP			Date and time of user registration
10	profile_picture	VARCHAR	255	Optional	Path to user's profile picture
11	bio	TEXT		Optional	User's biography

#### Teacher Table :

Sr.no	Fieldname	Datatype	Fieldsize	Constraints	Description
1	Faculty id	INT		Unique	Unique identifier for each faculty member
2	Name	VARCHAR			Faculty member's full name
3	Email	VARCHAR		Unique	Faculty member's email address
4	Password	VARCHAR			Faculty member's password (likely hashed)
5	Department	INT		Foreign Key	Department the faculty member belongs to
6	Role	VARCHAR			Faculty member's role (defaults to "Faculty")

**Student Table:-**

Sr.no	Fieldname	Datatype	Fieldsize	Constraints	Description
1	Student id	INT		Unique	Unique identifier for each student
2	Name	VARCHAR			Student's full name
3	Email	VARCHAR		Unique	Student's email address
4	Password	VARCHAR			Student's password (likely hashed)
5	Role	VARCHAR			Student's role (defaults to "Student")
6	Course	VARCHAR			Courses the student is enrolled in (multiple courses possible)
7	Photo	VARCHAR		Optional	Path to student's photo
8	Department	VARCHAR			Student's department

**Department Table:-**

Sr.no	Fieldname	Datatype	Fieldsize	Constraints	Description
1	Department id	INT		Unique	Unique identifier for each department
2	Name	VARCHAR			Department name
3	Description	TEXT			Department description

**Courses Table:-**

Sr.no	Fieldname	Datatype	Fieldsize	Constraints	Description
1	course_code	INT	10	Primary Key	Unique identifier for each course
2	Name	VARCHAR	255		Course name
3	department	TEXT			Course department
4	teacher_name	VARCHAR	255		Name of the teacher who created the course
5	Student_key	INT	10	UNIQUE	Access Key For Students

**Assignment Table:-**

Sr.no	Fieldname	Datatype	Field size	Constraints	Description
1	Course code	INT		Foreign Key	Code of the course the assignment belongs to
2	Title	VARCHAR			Assignment title
3	Description	TEXT			Assignment description
4	Deadline Date	DATE			Date the assignment is due
5	Deadline Time	TIME			Time the assignment is due
6	File	VARCHAR		Optional	Path to the assignment file
7	Marks	INT			Maximum marks for assignment

**Attendance Table:-**

Sr.no	Fieldname	Datatype	Fieldsize	Constraints	Description
1	Student	INT		Foreign Key	ID of the student whose attendance is being recorded
2	Course	INT		Foreign Key	ID of the course for which attendance is being recorded
3	Date	DATE			Date of the attendance

<b>4</b>	<b>Status</b>	<b>VARCHAR</b>			<b>Attendance status (e.g., Present, Absent, Late)</b>
----------	---------------	----------------	--	--	--------------------------------------------------------

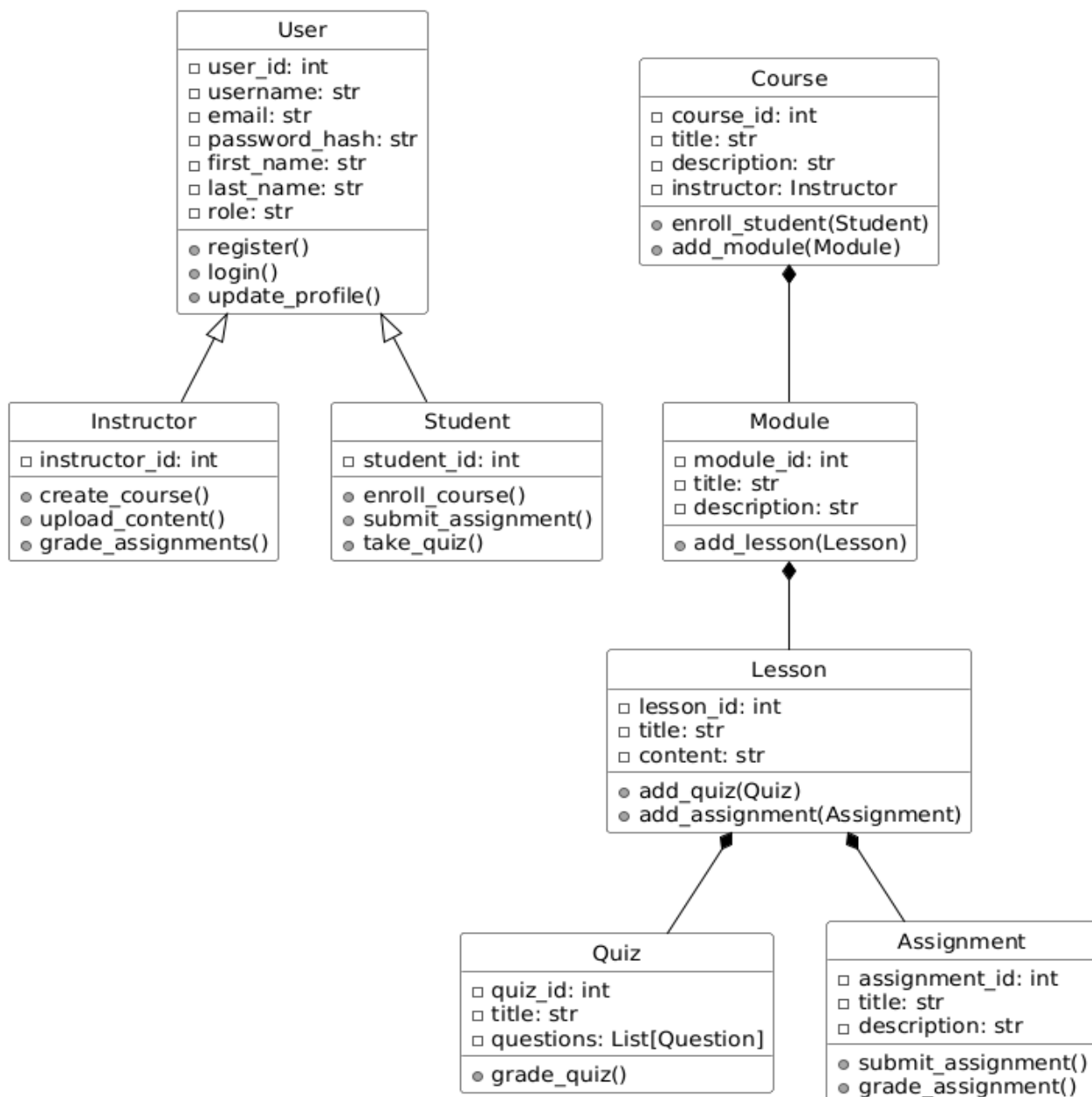
**Quiz Table:-**

<b>Sr.no</b>	<b>Fieldname</b>	<b>Datatype</b>	<b>Fieldsize</b>	<b>Constraints</b>	<b>Description</b>
<b>1</b>	<b>Title</b>	<b>VARCHAR</b>			<b>Quiz title</b>
<b>2</b>	<b>Description</b>	<b>TEXT</b>			<b>Quiz description</b>
<b>3</b>	<b>Course</b>	<b>INT</b>		<b>Foreign Key</b>	<b>ID of the course the quiz belongs to</b>
<b>4</b>	<b>Start Date</b>	<b>DATE</b>			<b>Date the quiz starts</b>
<b>5</b>	<b>Start Time</b>	<b>TIME</b>			<b>Time the quiz starts</b>
<b>6</b>	<b>End Date</b>	<b>DATE</b>			<b>Date the quiz ends</b>
<b>7</b>	<b>End Time</b>	<b>TIME</b>			<b>Time the quiz ends</b>
<b>8</b>	<b>Publish status</b>	<b>BOOLEAN</b>			<b>Indicates if the quiz is published (Yes/No or True/False)</b>

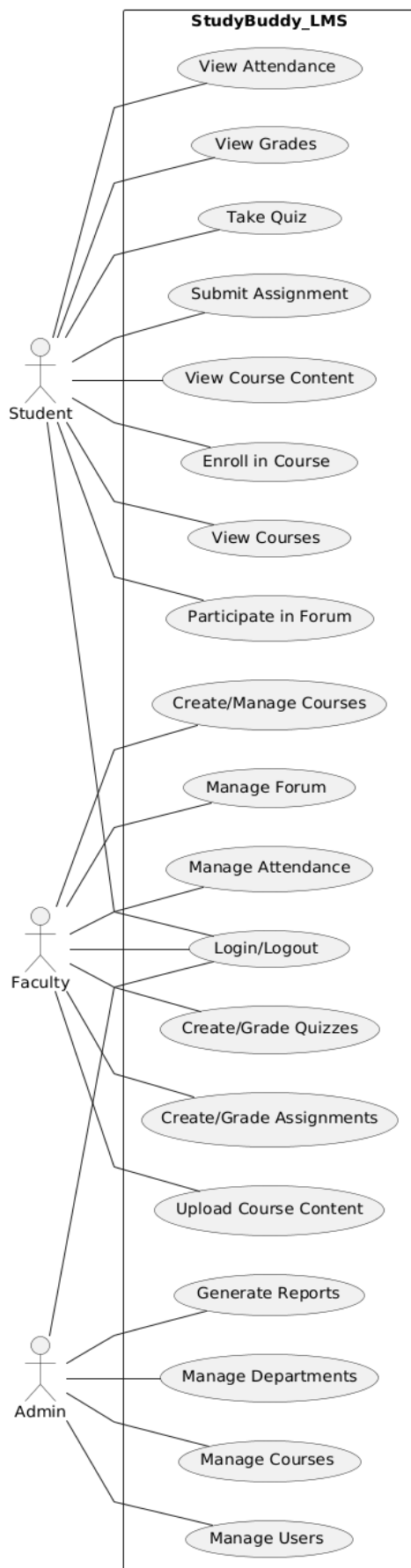


### 3.3 Class Diagram:

The class diagram is a static diagram. It represent the static view of an application or a system The class diagram describe the attributes an operation of a class an also the constraint imposed on the system. The class diagram is widely used in the modelling of the object oriented system.

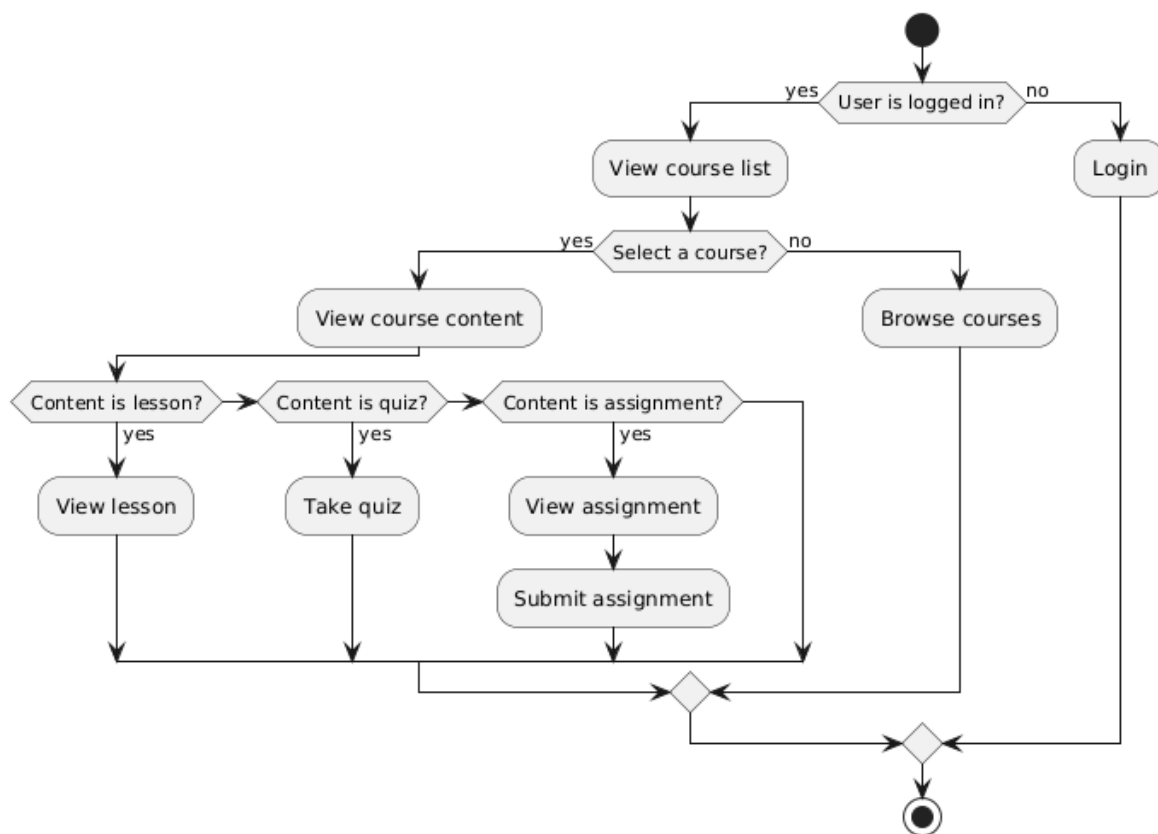


### 3.4 Use Case Diagram



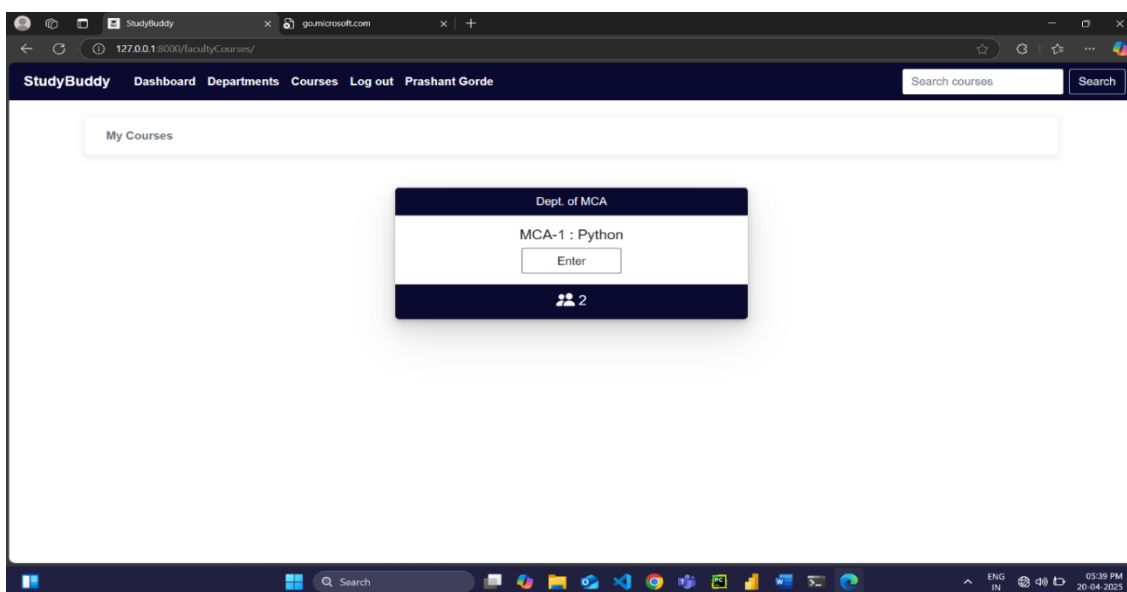
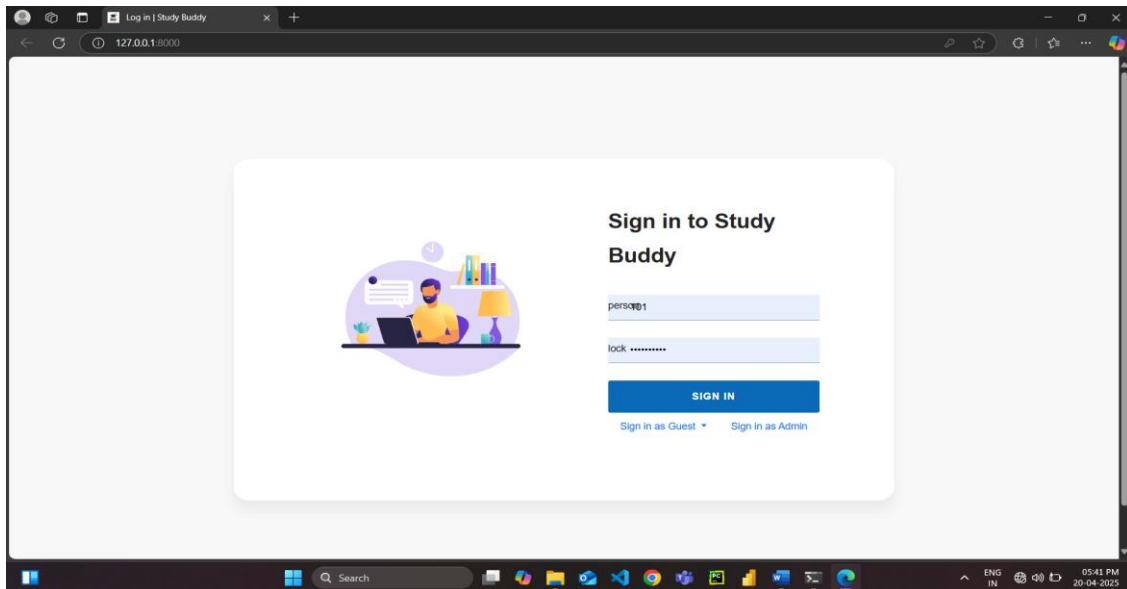
### 3.5 Activity Diagram:-

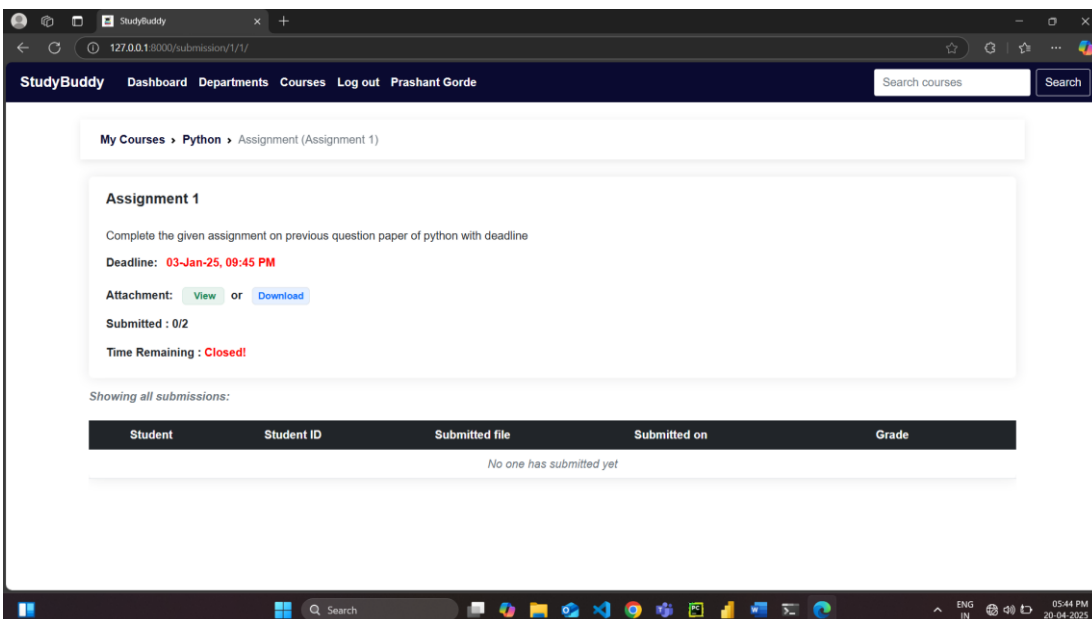
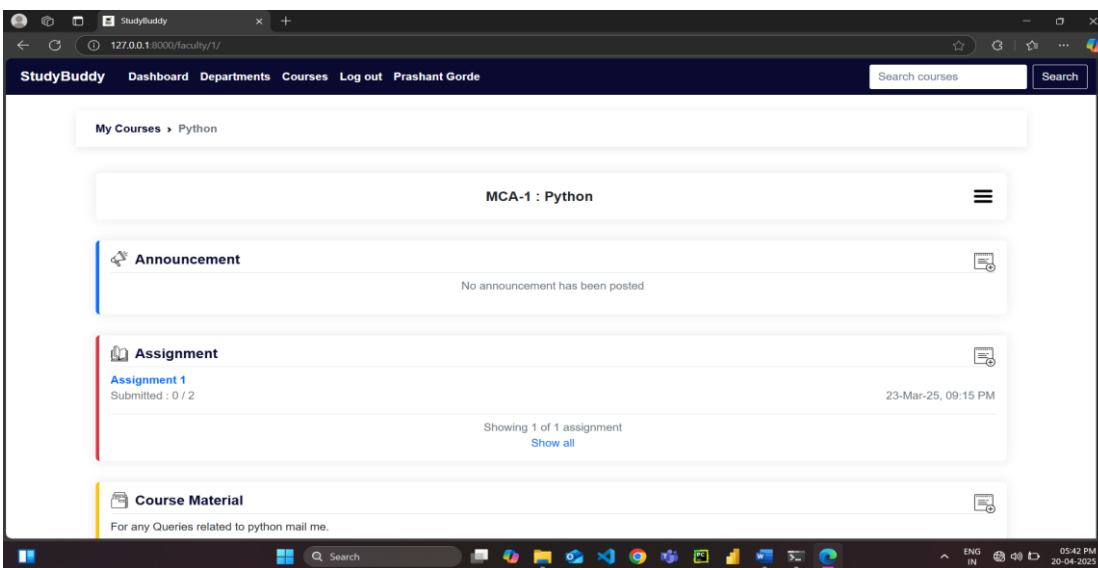
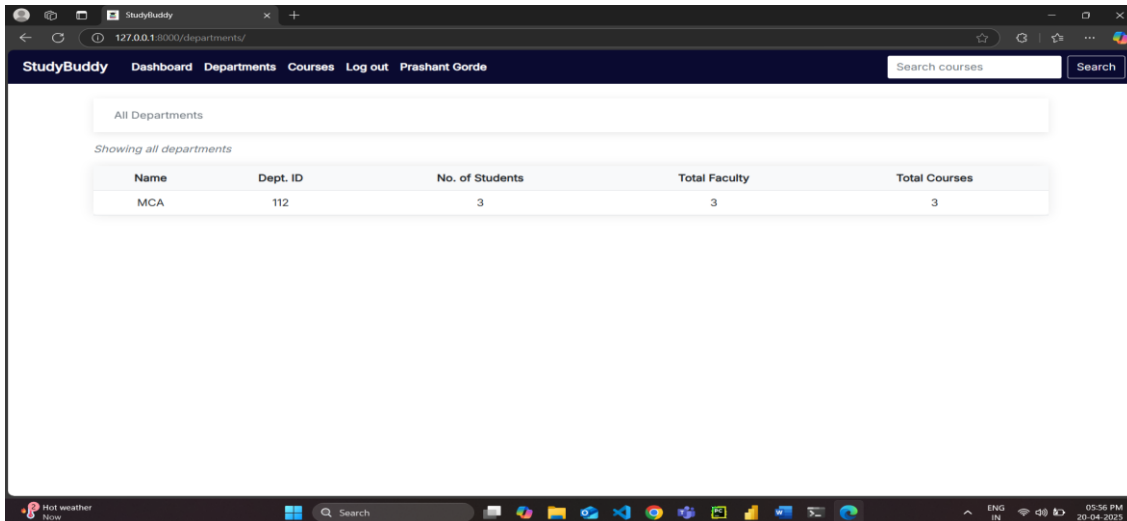
Activity diagram is another important behavioral diagram in UML . diagram to describe dynamic aspect of the system . Activity diagram is essentially an advanced version of the flow chart that modeling the flow one activity to another activity.

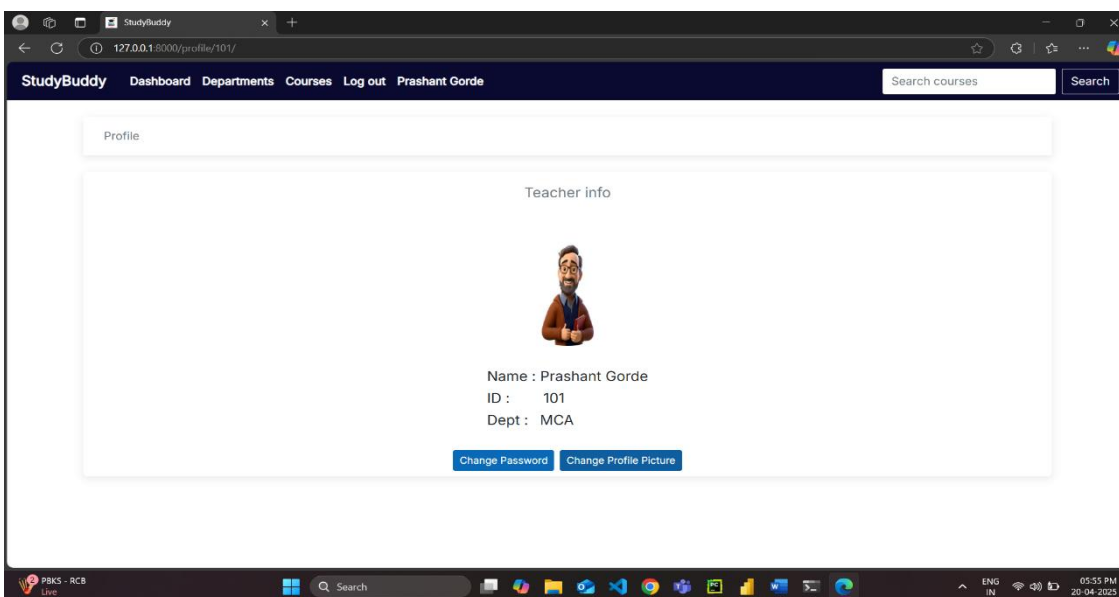
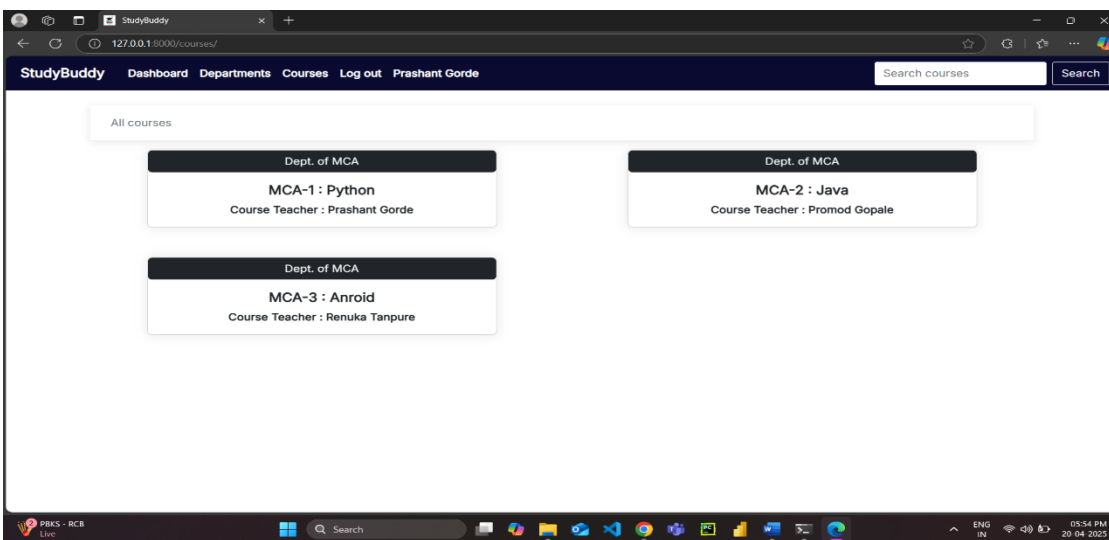
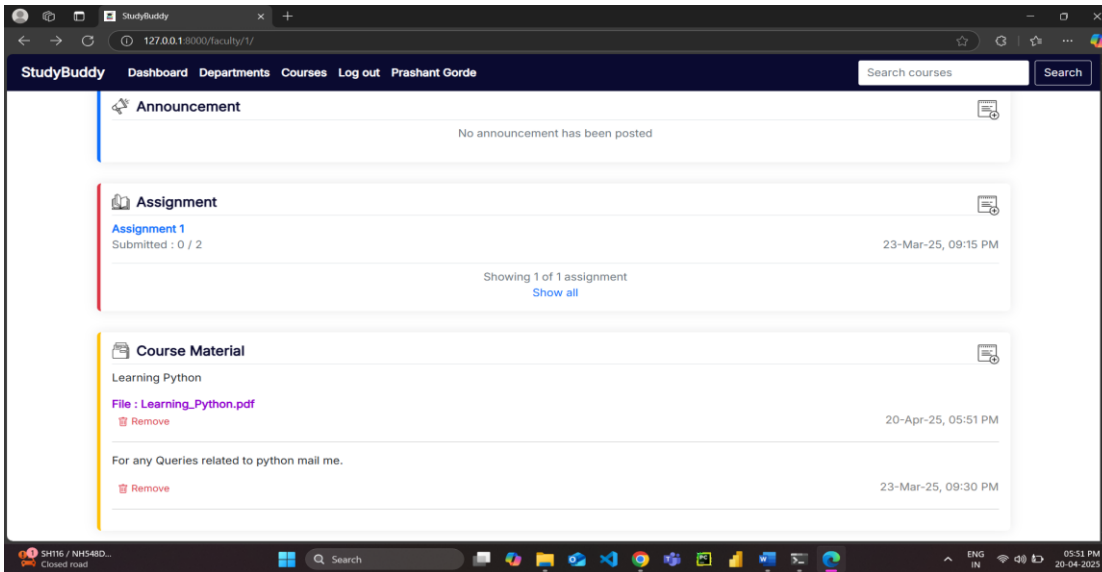


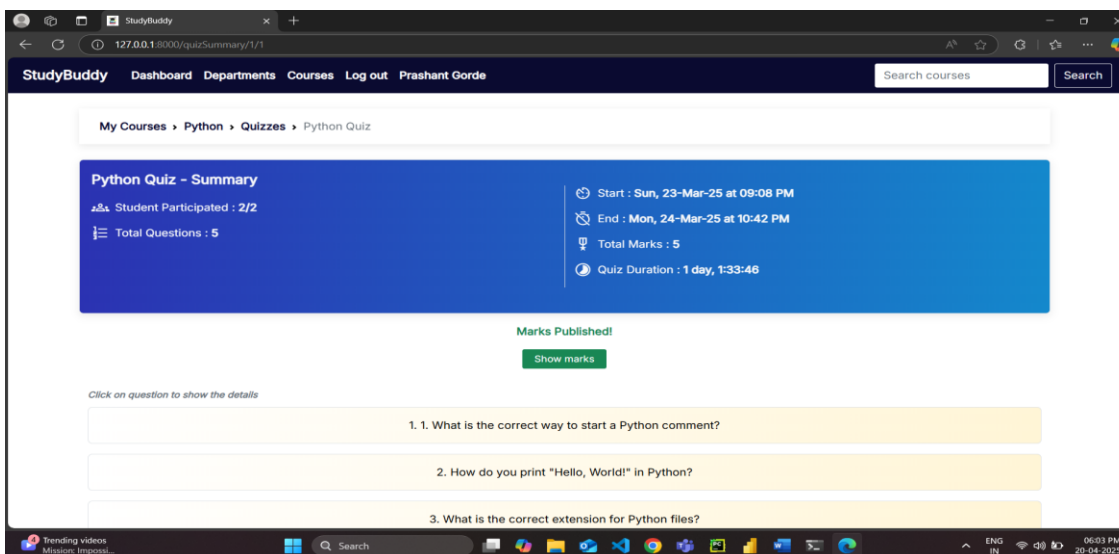
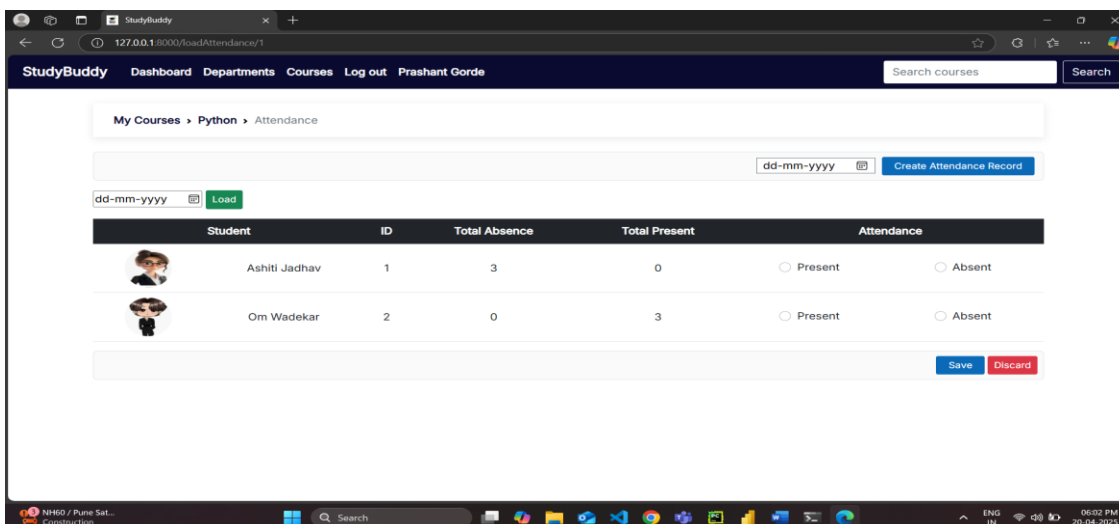
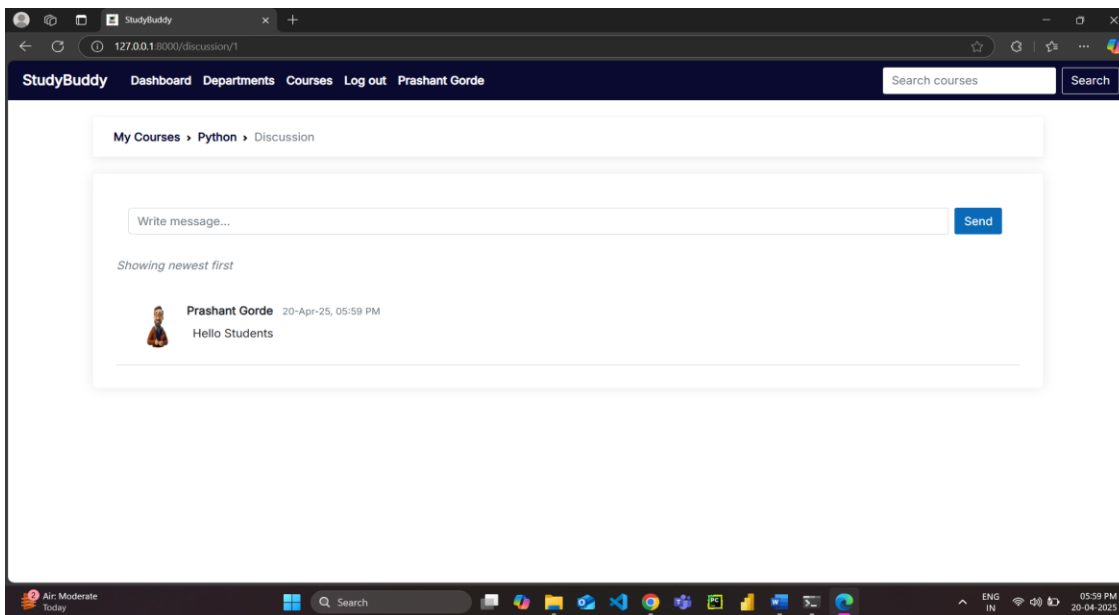
### 3.6 Output Screen:

#### Teacher Screens:-









StudyBuddy

DashboardDepartmentsCoursesLog outPrashant Gorde



Search courses

Search

Marks Published!

Hide marks

Marks of students

SL	Student	ID	Marks	Submitted on
1	 Ashiti Jadhav	1	4 / 5	Mon, 24-Mar-25 at 08:40 PM
2	 Om Wadekar	2	4 / 5	Mon, 24-Mar-25 at 08:56 PM

Click on question to show the details

1. 1. What is the correct way to start a Python comment?

2. How do you print "Hello, World!" in Python?

3. What is the correct extension for Python files?

4. Which of the following is a valid variable name in Python?

Today's moment  
It's Easter


Search

ENG IN06:03 PM20-04-2025

## Student Screens

Log in | Study Buddy

127.0.0.1:8000/login/



Sign in to Study Buddy

2

SIGN IN

Sign in as Guest

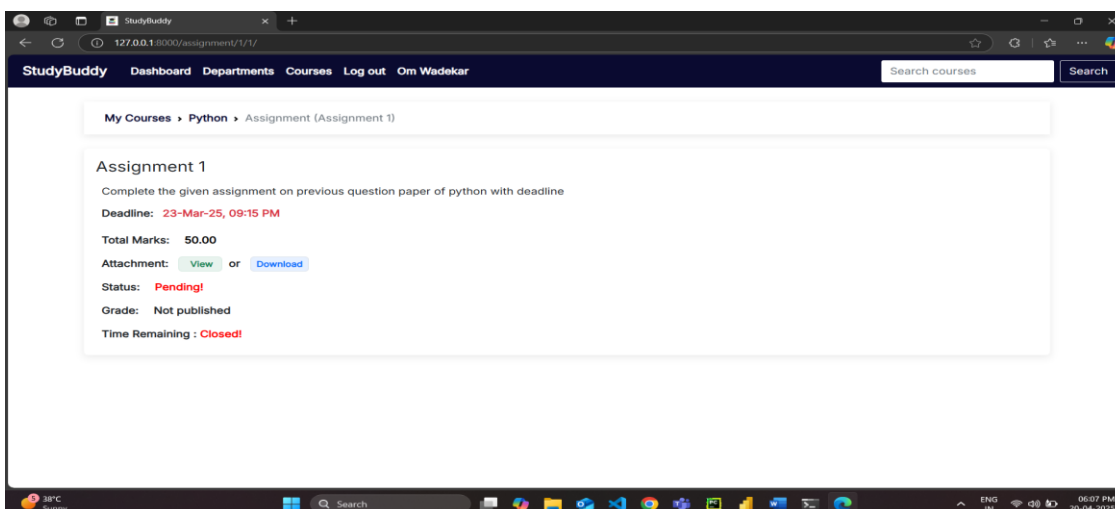
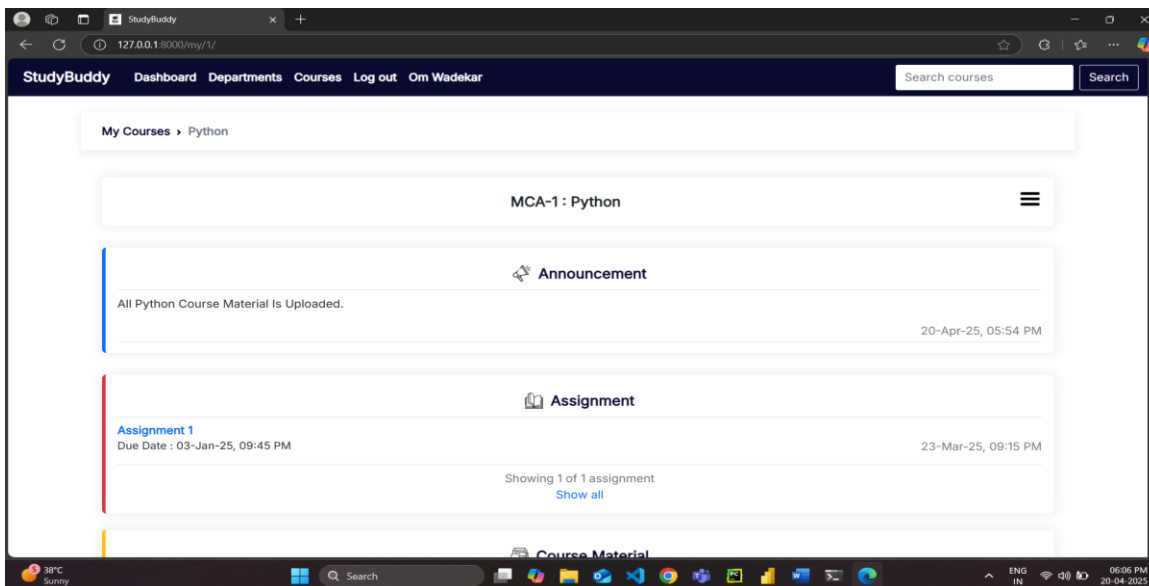
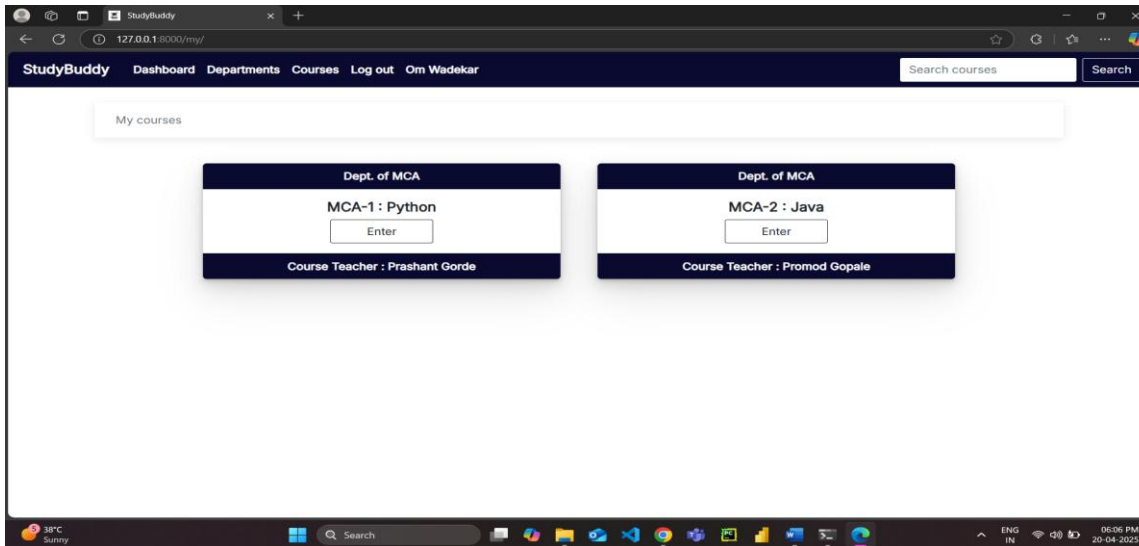
Sign in as Admin

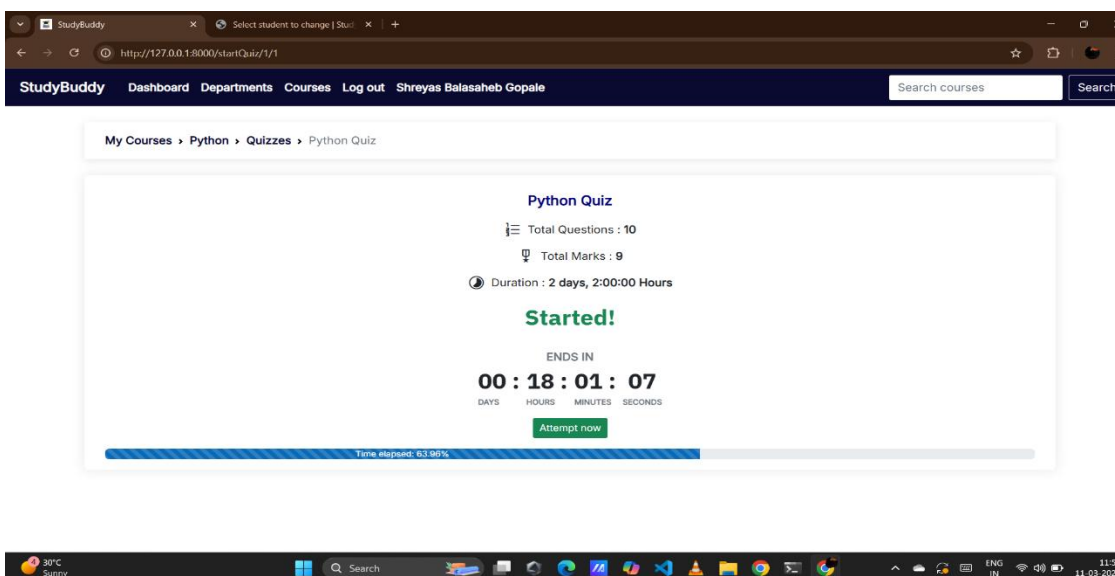
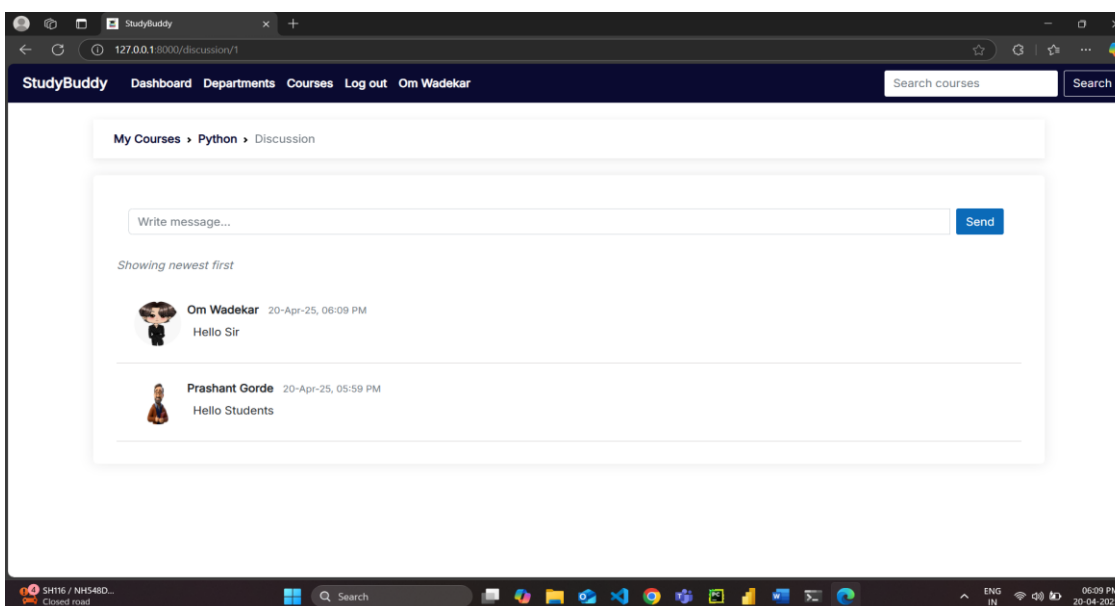
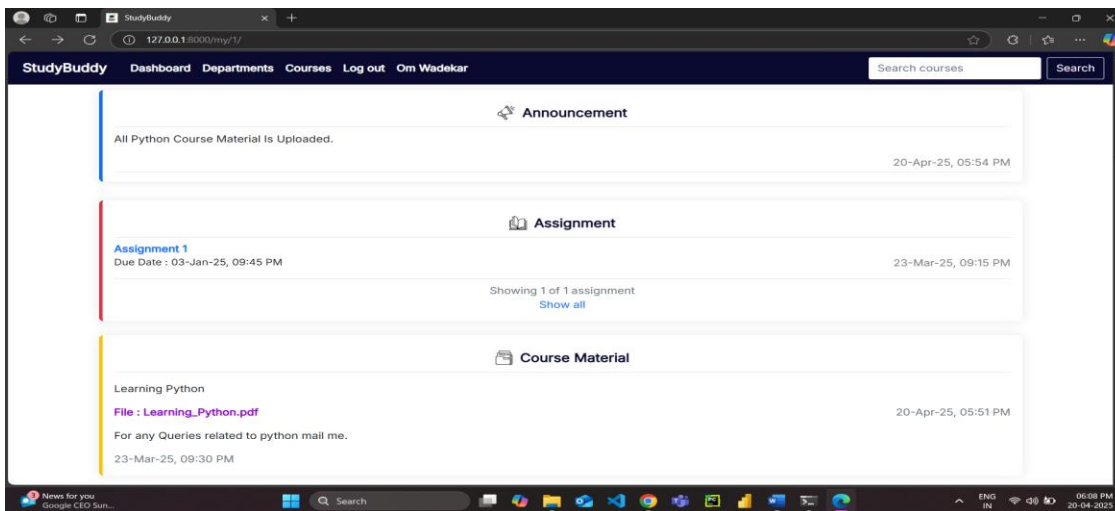
Temps to rise  
Tomorrow

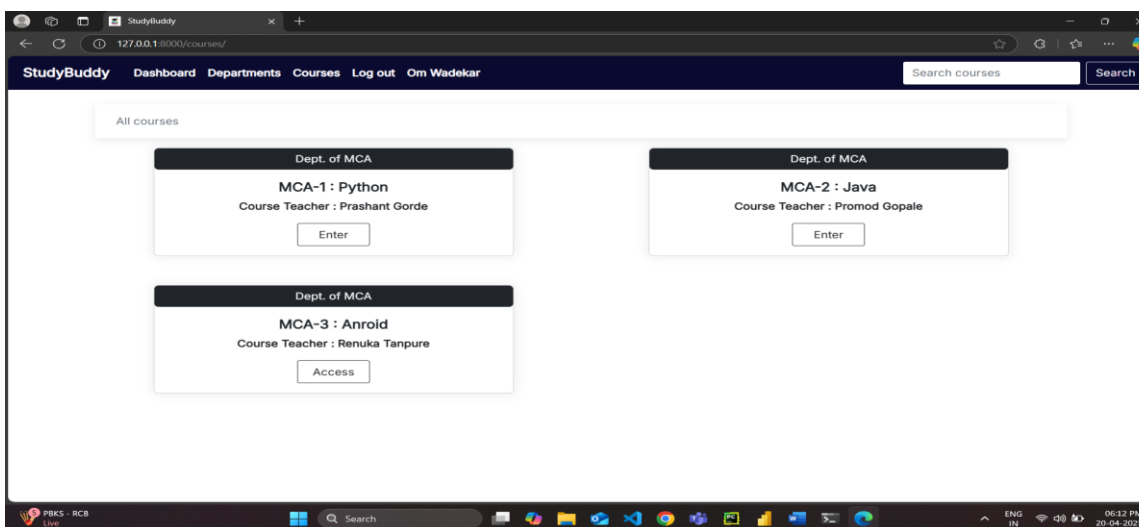
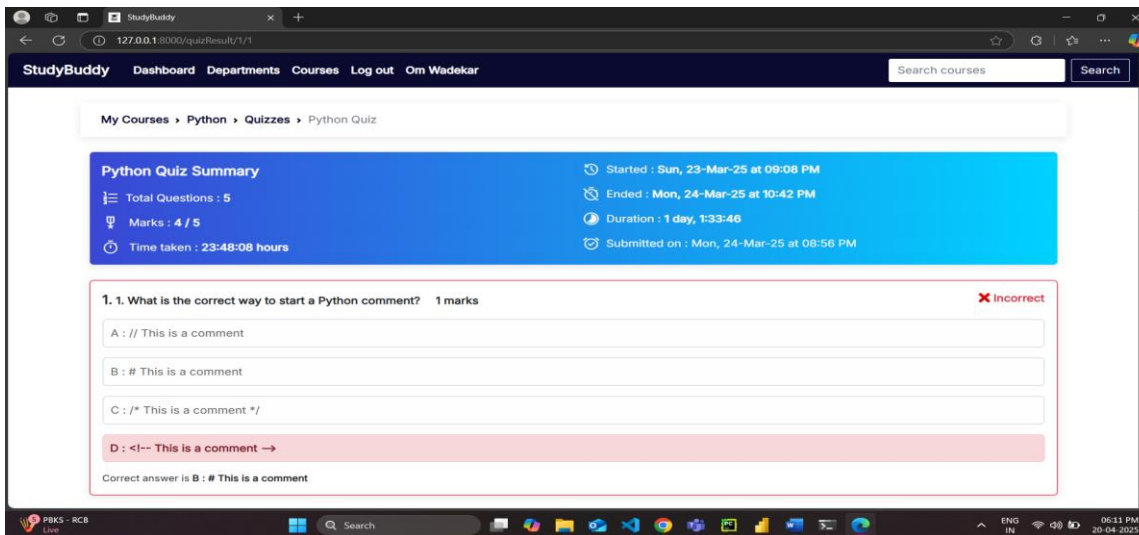
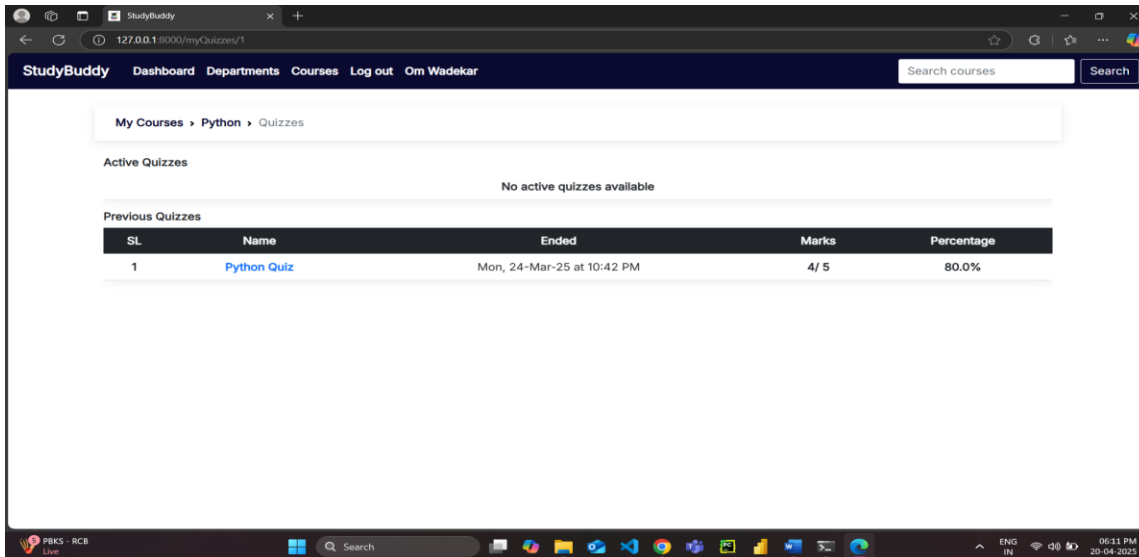
Search

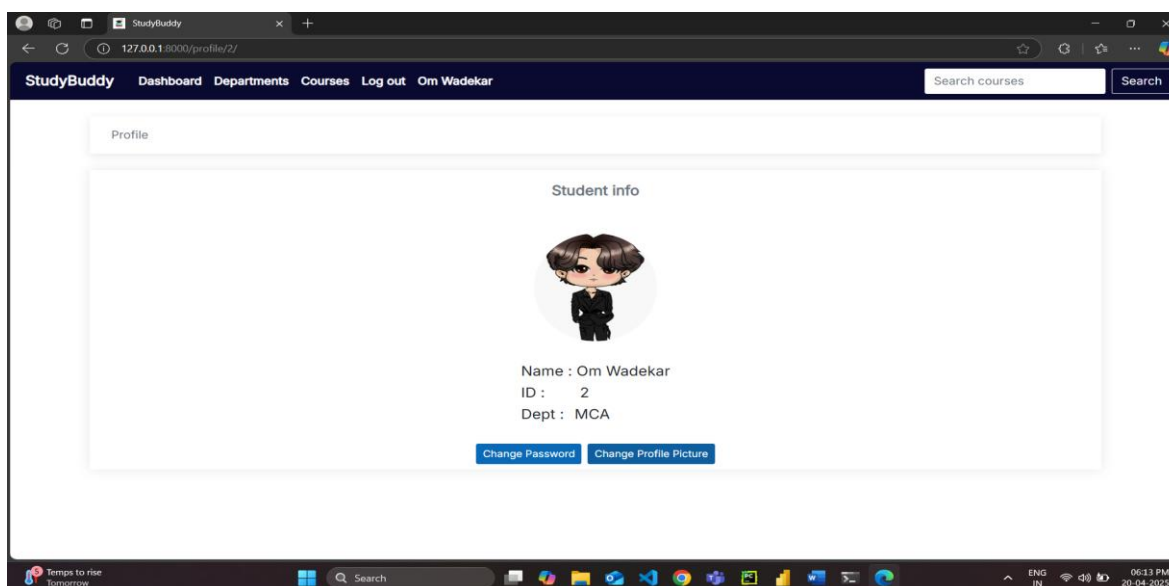
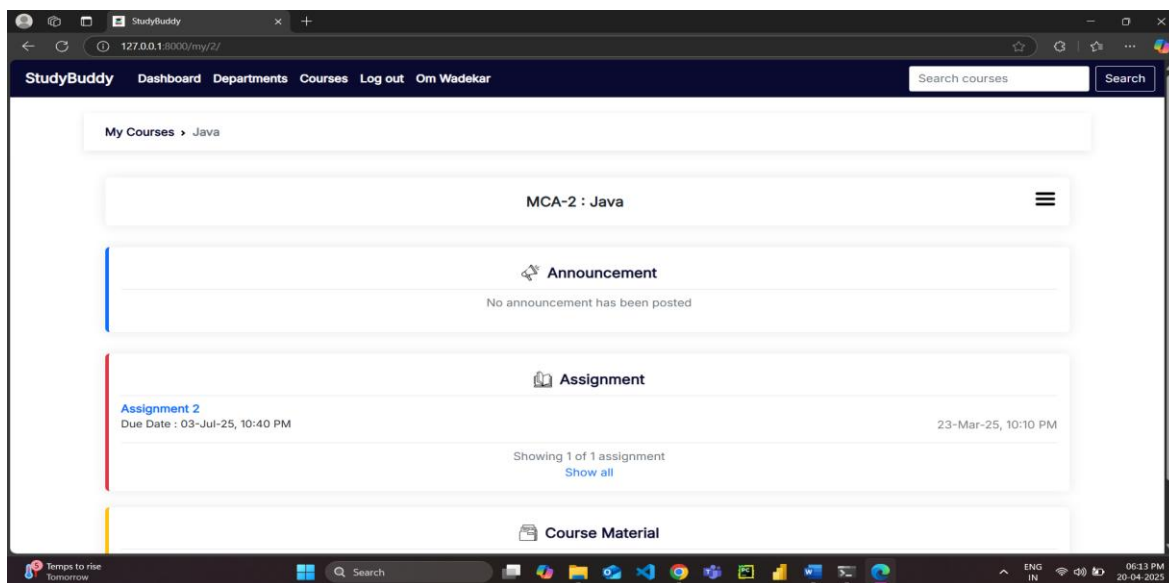
ENG IN06:14 PM20-04-2025











## Chapter No. 4 : Coding

Main/models.py

```
from django.db import models
```

```
from froala_editor.fields import FroalaField
```

```
# Create your models here.
```

```
class Student(models.Model):
```

```
    student_id = models.IntegerField(primary_key=True)
```

```
    name = models.CharField(max_length=100, null=False)
```

```
    email = models.EmailField(max_length=100, null=True, blank=True)
```

```
    password = models.CharField(max_length=255, null=False)
```

```
    role = models.CharField(
```

```
        default="Student", max_length=100, null=False, blank=True)
```

```
    course = models.ManyToManyField(
```

```
        'Course', related_name='students', blank=True)
```

```
    photo = models.ImageField(upload_to='profile_pics', blank=True,
```

```
        null=False, default='profile_pics/default_student.png')
```

```
    department = models.ForeignKey(
```

```
        'Department', on_delete=models.CASCADE, null=False, blank=False,
```

```
        related_name='students')
```

```
    def delete(self, *args, **kwargs):
```

```

if self.photo != 'profile_pics/default_student.png':

    self.photo.delete()

super().delete(*args, **kwargs)

class Meta:

    verbose_name_plural = 'Students'

    def __str__(self):

        return self.name

class Faculty(models.Model):

    faculty_id = models.IntegerField(primary_key=True)

    name = models.CharField(max_length=100, null=False)

    email = models.EmailField(max_length=100, null=True, blank=True)

    password = models.CharField(max_length=255, null=False)

    department = models.ForeignKey(

        'Department', on_delete=models.CASCADE, null=False, related_name='faculty')

    role = models.CharField(

        default="Faculty", max_length=100, null=False, blank=True)

    photo = models.ImageField(upload_to='profile_pics', blank=True,

                               null=False, default='profile_pics/default_faculty.png')

    def delete(self, *args, **kwargs):

        if self.photo != 'profile_pics/default_faculty.png':

```

```
self.photo.delete()
```

```
    super().delete(*args, **kwargs)
```

```
class Meta:
```

```
    verbose_name_plural = 'Faculty'
```

```
def __str__(self):
```

```
    return self.name
```

```
class Department(models.Model):
```

```
    department_id = models.IntegerField(primary_key=True)
```

```
    name = models.CharField(max_length=100, null=False)
```

```
    description = models.TextField(null=True, blank=True)
```

```
class Meta:
```

```
    verbose_name_plural = 'Departments'
```

```
def __str__(self):
```

```
    return self.name
```

```
def student_count(self):
```

```
    return self.students.count()
```

```
def faculty_count(self):
```

```
    return self.faculty.count()
```

```
def course_count(self):
```

```
    return self.courses.count()
```

```

class Course(models.Model):

    code = models.IntegerField(primary_key=True)

    name = models.CharField(max_length=255, null=False, unique=True)

    department = models.ForeignKey(

        Department, on_delete=models.CASCADE, null=False, related_name='courses')

    faculty = models.ForeignKey(

        Faculty, on_delete=models.SET_NULL, null=True, blank=True)

    studentKey = models.IntegerField(null=False, unique=True)

    facultyKey = models.IntegerField(null=False, unique=True)

    class Meta:

        unique_together = ('code', 'department', 'name')

        verbose_name_plural = "Courses"

    def __str__(self):

        return self.name

class Announcement(models.Model):

    course_code = models.ForeignKey(

        Course, on_delete=models.CASCADE, null=False)

    datetime = models.DateTimeField(auto_now_add=True, null=False)

    description = FroalaField()

    class Meta:

```



```

    verbose_name_plural = "Announcements"

    ordering = ['-datetime']

def __str__(self):

    return self.datetime.strftime("%d-%b-%y, %I:%M %p")

def post_date(self):

    return self.datetime.strftime("%d-%b-%y, %I:%M %p")

class Assignment(models.Model):

    course_code = models.ForeignKey(

        Course, on_delete=models.CASCADE, null=False)

    title = models.CharField(max_length=255, null=False)

    description = models.TextField(null=False)

    datetime = models.DateTimeField(auto_now_add=True, null=False)

    deadline = models.DateTimeField(null=False)

    file = models.FileField(upload_to='assignments/', null=True, blank=True)

    marks = models.DecimalField(max_digits=6, decimal_places=2, null=False)

    class Meta:

        verbose_name_plural = "Assignments"

        ordering = ['-datetime']

```

## **Chapter No. 5: TESTING**

### **5.1. TEST STRATEGY**

#### **1. Test Objectives:**

- Ensure the StudyBuddy LMS meets all specified functional and non-functional requirements.
- Identify and resolve defects to ensure the system's reliability, performance, and security.

#### **2. Test Approach:**

- Implement a comprehensive testing strategy including unit, integration, system, performance, and user acceptance testing.
- Utilize a combination of manual and automated testing methods to achieve thorough test coverage.

#### **3. Test Environment:**

- Server-side: Linux-based environment (Ubuntu, CentOS), PostgreSQL, Django/DRF setup.
- Client-side: Modern web browsers (Chrome, Firefox, Safari, Edge) on various operating systems.

#### **4. Test Coverage:**

- Functional: Test all modules, including user management, course management, learning management, communication, and payment processing.
- Non-functional: Evaluate performance, usability, security, and scalability.

#### **5. Defect Management:**

- Log and categorize defects based on severity (critical, major, minor).
- Use a defect tracking tool (e.g., Jira, GitHub Issues) to manage and track defect resolution.

#### **6. Test Execution:**

- Execute test cases and scripts, recording pass/fail results. Perform regression testing after bug fixes and updates to ensure system stability.

## 5.2. UNIT TEST PLAN

### 1. Objective:

- Validate that individual components and modules of the StudyBuddy LMS function correctly in isolation.
- Ensure each module performs as expected before integration with other components.

### 2. Scope:

- Covers all critical backend modules (Django views, models, APIs) and frontend components (React components or Django templates).
- Focuses on testing core functionalities like user authentication, course creation, and data processing.

### 3. Approach:

- Use unit testing frameworks (e.g., Python's unittest, pytest; JavaScript's Jest) to automate unit tests.
- Test inputs, outputs, and error handling for each function and method in isolation.

### 4. Test Environment:

- Local development environment with Python and JavaScript testing frameworks.
- Mocking and stubbing to isolate modules from external dependencies (database, APIs).

### 5. Test Data:

- Predefined mock data for users, courses, and other entities.
- Simulated test cases for both valid and invalid inputs.

### 6. Test Execution:

- Execute unit tests as part of the continuous integration (CI) process.
- Track test results and identify failures during testing.

### 5.3 Acceptance Test Plan

Project Name: Study Buddy

Test Plan ID: ATP-SB-01

#### Objective:

To verify that the StudyBuddy Learning Management System meets all functional and non-functional requirements as expected by the stakeholders.

#### Acceptance Criteria:

- **ATP001 – User Module:** Users should be able to register, log in, and manage their profiles.
- **ATP002 – Course Module:** Teachers should be able to create and manage courses; students should be able to view and join courses.
- **ATP003 – Assignment:** Teachers should be able to upload assignments; students should be able to access and view them.
- **ATP004 – Attendance:** Attendance must be recorded accurately per student, per course, and per date.
- **ATP005 – Quiz:** Teachers should be able to create and publish quizzes; students should only view published quizzes.
- **ATP006 – Security:** Passwords must be securely hashed; users must only access features permitted by their role.
- **ATP007 – Student Access:** Students should be able to enroll in courses using a valid student key.

**Entry Criteria:**

- All development activities are complete.
- Unit and integration testing have been conducted.
- No critical or high-severity bugs remain open.

**Exit Criteria:**

- All acceptance test cases have passed.
- System is stable and meets documented requirements.
- The product is ready for deployment and user training.

**Test Environment:**

- Operating System: Windows
- Web Browser: Google Chrome / Microsoft Edge
- Backend: Python with Flask/Django
- Database: MySQL
- Server: Localhost or hosted environment

**Approval:**

- Project Lead : - Wadekar Sakshal
- Project Mentor :- Prof P. C. Gorde
- Approval Date - 30-4-2025

#### 5.4. Test Case

Test Case ID	Module	Test Scenario	Test Steps	Expected Result	Actual Result	Status
TC001	Users	Register a new user	Fill in form with valid details	User is created and saved in DB	As expected	Pass
TC002	Users	Duplicate email registration	Register with an existing email	Error message shown	As expected	Pass
TC003	Users	Invalid email format	Try registering with invalid email	Validation error	As expected	Pass
TC004	Users	Login with valid credentials	Enter correct email & password	Login success, redirect to dashboard	As expected	Pass
TC005	Users	Login with wrong password	Enter valid email, wrong password	Login fails with message	As expected	Pass
TC006	Course	Teacher creates course	Fill in course form with valid data	Course saved with unique course_code	As expected	Pass
TC007	Course	Duplicate student access key	Create course with same key	Error for unique constraint	As expected	Pass
TC008	Assignment	Add assignment to course	Upload assignment under course	Assignment is visible in that course	As expected	Pass
TC009	Assignment	Past deadline entry	Enter past date for deadline	Error or warning shown	As expected	Pass

<b>TC010</b>	<b>Atten dance</b>	<b>Mark attendance</b>	<b>Mark student as Present/Absent</b>	<b>Attendance recorded per date</b>	<b>As expected</b>	<b>Pass</b>
<b>TC011</b>	<b>Quiz</b>	<b>Teacher creates quiz</b>	<b>Input title start/end time,course</b>	<b>Quiz saved and linked</b>	<b>As expected</b>	<b>Pass</b>
<b>TC012</b>	<b>Quiz</b>	<b>Publish quiz</b>	<b>Toggle publish status</b>	<b>Quiz visible to students</b>	<b>As expected</b>	<b>Pass</b>
<b>TC013</b>	<b>Stude nts</b>	<b>Enroll in course</b>	<b>Use valid student key</b>	<b>Student added to course</b>	<b>As expected</b>	<b>Pass</b>
<b>TC014</b>	<b>File Uploa ds</b>	<b>Upload profile picture</b>	<b>Upload .jpg/.png</b>	<b>File saved and path stored</b>	<b>As expected</b>	<b>Pass</b>
<b>TC015</b>	<b>Secur ity</b>	<b>Password hashing</b>	<b>Register/login user</b>	<b>Password stored hashed, not plain</b>	<b>As expected</b>	<b>Pass</b>

## Chapter No 6. Drawbacks And Limitations

### Drawbacks:

#### 1. **LimitedScalability:**

The current system may not perform efficiently under a very high number of users or large datasets unless optimized and hosted on scalable infrastructure.

#### 2. **NoReal-TimeFeatures:**

Features like live chat, notifications, or real-time attendance tracking are not included, which may reduce interactivity.

#### 3. **BasicUI/UXDesign:**

The user interface may be functional but lacks modern design elements, making it less engaging or intuitive for end-users.

#### 4. **RoleManagementLimitations:**

Role-based access is predefined (admin, teacher, student) and may not support custom roles or permissions easily.

#### 5. **LimitedReportingandAnalytics:**

The system lacks advanced reporting features such as detailed performance analytics, student progress tracking over time, or visual dashboards.

#### 6. **NoMobileAppSupport:**

The system is web-based and may not be optimized for mobile use or available as a dedicated app.

#### 7. **FileStorageConcerns:**

If using local server storage, uploaded files (assignments, photos) can lead to storage issues over time unless regularly maintained or integrated with cloud storage.

#### 8. **NoNotificationSystem:**

There is no email or in-app notification feature to alert users about upcoming deadlines, quizzes, or course update



## **Limitations:**

**1.Limited User Roles:** The current version of Study Buddy supports three primary user roles: Administrator, Instructor, and Student. This may limit flexibility in the future if additional roles, such as teaching assistants, moderators, or guest users, are required.

**2.Reliance on Third-Party Integrations:** Study Buddy relies on third-party integrations for live classes (Zoom, Google Meet) and payment processing (Stripe, PayPal). Any limitations or disruptions in these third-party services could affect the functionality of the LMS.

**3.AI Capabilities:** The effectiveness of the AI-powered features (course recommendations, chatbot) depends on the quality of data and algorithms. These features may have limitations in handling complex or nuanced queries, and their accuracy may improve over time with more data and refinement.

**4.Content Dependency:** The success of Study Buddy relies heavily on the quality and engagement of the courses created by instructors. The platform itself cannot guarantee effective learning outcomes if the content is inadequate.

**5.Potential Scalability Challenges:** While designed for scalability, handling extremely high user traffic and data volumes might require further infrastructure optimization and investment.

**6.Limited Offline Access:** Study Buddy primarily functions as an online platform. Offline access to learning materials may be limited, potentially affecting users with unreliable internet connections.

**7.Accessibility Features:** While basic accessibility considerations are incorporated, further enhancements may be needed to cater to users with specific disabilities.

**8.No Mobile Application:** Currently, Study Buddy is designed as a web application. A dedicated mobile application could enhance user experience and accessibility on mobile devices.

## Chapter No. 7: PROPOSED ENHANCEMENTS LMS

1. **Expanded User Roles:** Introduce new user roles, such as Teaching Assistant, Moderator, and Guest User, to provide more flexibility and control over user permissions and access within the platform.
2. **Enhanced AI Capabilities:** Continuously improve the AI-powered features by refining algorithms, expanding the knowledge base of the chatbot, and incorporating more sophisticated machine learning models for personalized recommendations.
3. **Offline Access:** Develop offline access capabilities for learning materials, allowing users to download content and access it even without an internet connection. This will improve accessibility for users with limited or unreliable internet access.
4. **Advanced Gamification:** Implement more advanced gamification features, such as points, rewards, challenges, and interactive elements, to further enhance user engagement and motivation.
5. **Interactive Content Creation Tools:** Provide instructors with more interactive content creation tools, such as simulations, interactive exercises, and branching scenarios, to create more engaging and effective learning experiences.
6. **Integration with Learning Analytics Dashboards:** Develop dashboards that provide instructors with detailed insights into student performance, learning patterns, and engagement metrics. This will help them tailor their teaching strategies and improve learning outcomes.
7. **Enhanced Accessibility Features:** Conduct thorough accessibility audits and implement features to cater to users with diverse needs, such as screen reader compatibility, keyboard navigation, and adjustable font sizes.

## **Chapter No 8. CONCLUSION**

The Study Buddy Learning Management System (LMS) is poised to be a comprehensive and robust platform designed to enhance the online learning experience for both instructors and students. The system's proposed features, such as user-friendly course management, interactive learning tools, AI-powered assistance, and gamified elements, address the limitations of existing LMS solutions, creating a more engaging and personalized learning environment. The technical feasibility of the project is supported by the utilization of a well-established technology stack, including Django, React.js (or Django Templates), and PostgreSQL, ensuring scalability and reliability. Economic feasibility is evident in the growing demand for online learning platforms, with Study Buddy offering monetization opportunities for instructors and a potentially sustainable business model. Operational feasibility is ensured through a well-defined operational plan, ongoing support, and continuous improvement based on user feedback. While limitations such as reliance on third-party integrations and the need for high-quality content exist, proactive mitigation strategies can address these challenges effectively. Ultimately, Study Buddy has the potential to revolutionize online education by providing a dynamic, accessible, and effective platform for learners and educators alike.

## **Chapter No 9. BIBLIOGRAPHY**

- Teachers Guide
- [www.google.com](http://www.google.com)
- [www.youtube.com](http://www.youtube.com)