

# **EduManage Pro**

Project Synopsis Report

*Submitted in partial fulfilment of the requirement of the degree of*

**BACHELORS OF TECHNOLOGY**

**in**

**CSE with Specialization in AI-ML**

*To*

**K.R Mangalam University**

*By*

**Deepak (2301730222)**

**Shivani Sharma(2301730268)**

**Vanshika Agrawal(2301730263)**

Under the supervision of

**Dr. Tanu Gupta**



Department of Computer Science and Engineering

School of Engineering and Technology

K.R Mangalam University, Gurugram- 122001, India

January 2025

## CERTIFICATE

This is to certify that the Project Synopsis entitled, "**Student- Teacher Portal**" submitted by **Deepak(2301730222) Shivani(2301730268), Vanshika(2301730263)** to **K.R Mangalam University, Gurugram, India**, is a record of bonafide project work carried out by them under my supervision and guidance and is worthy of consideration for the partial fulfilment of the degree of **Bachelor of Technology in Computer Science and Engineering** of the University.

### **Type of Project : Industry Problem**

Signature of Internal supervisor :

Name and designation of supervisor :

Signature of Project Coordinator

Dr Vandna Batra

Date: 30<sup>th</sup> April 2025

# INDEX

<b>Sr No.</b>		<b>Page No.</b>
1.	Abstract	3
2.	Introduction (description of broad topic)	4
3.	Motivation	5
4.	Literature Review	6-7
5.	Gap Analysis	8
6.	Problem Statement	9-10
7.	Objectives	11-12
8.	Tools/platform Used	13-16
9.	Methodology	17-20
10.	References	21

## Abstract

The rise of online services has transformed how businesses operate and how consumers interact with them. In recent years, the e-commerce industry has witnessed significant growth, driven by the increasing penetration of the internet, mobile devices, and digital payment systems. One of the primary goals of e-commerce platforms is to provide users with a seamless and secure shopping experience, ensuring convenience and accessibility while safeguarding sensitive data.

This project focuses on building an e-commerce website that serves as a comprehensive platform for users to browse, search, and purchase products with ease. Developed using Python and Django, the website integrates features such as dynamic product listings, responsive design, and user-friendly navigation to enhance the shopping experience. The project leverages robust database management to efficiently store and retrieve information about products, user accounts, and order histories.

Additionally the website incorporates advanced search functionality to allow users to find products based on keywords, categories, and filters. The design prioritizes responsiveness to ensure compatibility across devices, from desktops to mobile phones. Emphasis is placed on optimizing the user interface and backend processes to achieve a fast and secure online shopping platform. Although financial transactions are excluded in this project, the structure lays a strong foundation for future integration with payment gateways.

As e-commerce continues to evolve, this project demonstrates the potential for further development in areas such as personalized recommendations, inventory management, and customer relationship management. By building this website, we aim to provide an adaptable and scalable solution for small businesses venturing into the digital marketplace.

**KEYWORDS: E-commerce, Django, Python, Responsive Design, Database Management**

# Introduction

With the rapid evolution of technology and the internet, education has transformed significantly, paving the way for innovative methods of knowledge delivery. However, accessibility to quality education remains a challenge, particularly in underdeveloped and developing regions. The COVID-19 pandemic further exacerbated this issue, forcing educational institutions to shift to online learning platforms. Despite this shift, many students faced challenges such as lack of immediate assistance and personalized interaction with teachers, making it evident that traditional online education platforms were insufficient to address every learner's needs.

The emergence of interactive online platforms bridges the gap by offering real-time assistance and personalized learning experiences. This project introduces an **Online Educational Assistance System** to create a seamless interface where students can post academic queries and receive prompt responses from educators. It integrates features like personalized profiles, real-time notifications, and an intuitive text editor, providing students and teachers a structured way to interact.

Education is a cornerstone for societal growth, and effective communication between educators and students is fundamental. The platform aims to enhance academic support, especially for students who require tailored help in specific subjects. With the rise of smart applications and the integration of AI/ML technologies, such platforms are becoming more prevalent and necessary. This project focuses on leveraging Django's robust backend and Python's flexibility to create a user-friendly, secure, and scalable educational support system.

# Motivation

In recent years, the field of education has faced several challenges, ranging from accessibility issues to the lack of personalized learning approaches. Urbanization, growing economic disparities, and limited resources in underprivileged areas have further widened the educational gap. Additionally, the COVID-19 pandemic exposed the vulnerabilities of traditional education systems, disrupting the learning process for millions of students globally.

Online learning platforms have emerged as a solution, but many of them fail to address the individual needs of students effectively. These platforms often lack features such as personalized assistance, real-time query resolution, and direct interaction with subject experts. As a result, students continue to face difficulties in comprehending complex concepts and accessing tailored guidance.

The demand for online educational assistance systems has been growing steadily. According to a report by MarketsandMarkets, the global e-learning market size is projected to grow from USD 226 billion in 2020 to USD 375 billion by 2026, at a compound annual growth rate (CAGR) of 8.56%. This significant growth highlights the increasing reliance on digital tools to bridge educational gaps and provide quality learning experiences to students worldwide.

Figure 1 illustrates the rising demand for online educational platforms, particularly during the pandemic, as schools and colleges shifted to remote learning. Despite this surge, there remains a critical need for systems that prioritize interactivity, accessibility, and user-friendly interfaces for students and educators alike.

This project aims to address these gaps by creating an **Online Educational Assistance System** that fosters a seamless interaction between students and educators. By incorporating features such as real-time query notifications, user-friendly text editors, and personalized profiles, this system ensures effective academic support. The primary goal of this project is to create a platform that not only helps students with their academic challenges but also makes the learning process engaging and accessible for all.

# Literature Review

The development of online educational platforms has evolved significantly over the past decade, emphasizing accessibility, interactivity, and personalized learning. Studies have shown that traditional learning management systems (LMS) often fail to meet the dynamic needs of students, especially in underprivileged regions where resources are scarce. Research highlights the limitations of existing systems, such as the lack of real-time query resolution, inadequate personalization, and insufficient integration with modern tools. Platforms with features like real-time notifications, personalized profiles, and intuitive interfaces have proven more effective in bridging educational gaps. The integration of AI/ML technologies further enhances these platforms by providing tailored learning experiences and data-driven insights into student progress. However, many systems remain constrained by issues such as scalability, high costs, and limited technical support. The COVID-19 pandemic underscored these challenges, driving a surge in demand for innovative solutions. This project builds on existing research, leveraging Django and Python to address identified gaps by creating a scalable, user-centric, and interactive Online Educational Assistance System that ensures efficient academic support and fosters engagement between students and educators.

Some of the similar existing systems studied are mentioned in the table below (Table 1).

Factors	Evaluation Criteria	EduMax	LearnEase	SmartStudy
User Interface	- Intuitiveness and user-friendliness	Highly Intuitive	Moderate	Intuitive
	- Availability of customization options	High	Moderate	Limited
Real-Time Features	- Instant query notification to teachers	Yes	No	Yes
	- Quick response mechanism	Comprehensive	Limited	Comprehensive
Factors	Evaluation Criteria	EduMax	LearnEase	SmartStudy
Personalization	- User profile customization	Yes	No	Yes
	- Tracking learning progress	High	Moderate	Moderate
Security & Authentication	- Secure login and data management	Excellent	Good	Excellent
	- Compliance with privacy standards	Yes	Yes	Yes

<b>Integration &amp; Compatibility</b>	- Compatibility with existing LMS systems	High	Moderate	Moderate
	- Integration with third-party tools	Seamless	Limited	Seamless
<b>Cost &amp; Scalability</b>	- Upfront costs	Moderate	High	Moderate
	- Scalability for additional users	Highly Scalable	Limited	Scalable
<b>Support &amp; Maintenance</b>	- Availability of technical support	Excellent	Good	Moderate
	- Frequency of updates	Regular	Irregular	Regular
<b>Feedback Mechanism</b>	- Ability for students to rate teacher responses	Yes	No	Yes

**Table 1. Comparison of Existing Educational Assistance Systems**



# Gap Analysis

## Features of the System

The student-Teacher Portal includes the following core features:

1. **Progress Report Management** – Teachers can upload and update students' academic progress and grades in real time.
2. **Attendance Tracking** – An automated system to mark and track student attendance records.
3. **Assignment Submission & Evaluation** – Students can submit assignments online, and teachers can evaluate and grade them efficiently.
4. **Study Material Sharing** – Teachers can upload resources such as PDFs, lecture notes, and video tutorials for students.
5. **User Authentication & Role Management** – Secure login system with role-based access control for students, teachers, and administrators.

## Technology Used

The portal is being developed using the following technologies:

- **Frontend:** HTML, CSS, JavaScript for a responsive user interface.
- **Backend:** Java for server-side logic and database connectivity.
- **Database:** MySQL for storing student and teacher records securely.

# **Problem Statement**

## **1. Scalability Challenges in AI-Driven Educational Platforms**

### **Problem Overview:**

AI-driven educational platforms offer promising improvements in personalized learning, including automatically generated questions and assessments tailored to individual student needs. However, these systems have largely been tested in small-scale, controlled environments—such as specific classrooms or well-funded institutions—and their effectiveness and feasibility in large-scale, resource-constrained environments remain underexplored.

### **Core Issue:**

The scalability challenge arises from the fact that many AI-driven systems are built on complex models that require high computational power, specialized expertise, and substantial infrastructure. In many regions, particularly in under-resourced or rural areas, educational institutions may lack the technology, funding, or expertise to implement and maintain such systems. Furthermore, issues like internet access, bandwidth limitations, and varying levels of digital literacy further complicate the deployment of AI-based learning platforms in these settings.

### **Recommendation:**

To address this scalability gap, research should focus on developing lightweight AI models and adaptive learning systems that require fewer computational resources and can function effectively even in environments with limited technological infrastructure. Additionally, the cost of implementation needs to be reduced through open-source software and collaborative efforts that allow institutions to share resources and knowledge.

## **2. Behavioural and Psychological Factors in Q&A Platforms**

### **Problem Overview:**

While Q&A platforms, particularly those enhanced with gamification and peer-to-peer elements, are designed to increase student engagement, they often fail to account for the behavioral and psychological factors that influence student participation. These factors include stress, anxiety, motivation, and competition, all of which can significantly affect how students interact with the platform and, ultimately, their learning outcomes.

### **Core Issue:**

Gamification elements like leaderboards, rewards, and competition may work for some students, increasing motivation and fostering a sense of achievement. However, for other students, these same elements can cause anxiety, increase stress, or promote unhealthy competition, ultimately leading to disengagement. Additionally, peer-driven Q&A platforms assume that students will feel comfortable participating and sharing knowledge with their peers.

However, students who are shy, introverted, or socially anxious may avoid engaging, even when the platform could benefit their learning.

# Objective of the Project

This project is aimed at developing a comprehensive, AI-powered online platform designed to facilitate seamless interaction between students and teachers. The platform will enable effective communication and knowledge exchange through a specialized Question-and-Answer (Q&A) system, wherein both students and teachers will have their own designated domains with tailored access and features.

The objectives of the project can be categorized into several key areas:

## 1. Development of a Dual Domain System

A central objective is to create a platform with two separate domains: one for students and another for teachers. This segmentation ensures that the specific needs of both groups are addressed in a specialized manner.

- **For Students:** The student domain will focus on providing access to AI- generated questions, personalized feedback, and a repository of learning materials. Students will be able to post questions, interact with peers, and receive timely responses.
- **For Teachers:** The teacher domain will enable instructors to track student progress, review questions submitted by students, and provide detailed answers and resources. Teachers will also have the ability to create and manage assignments, track overall class performance, and offer personalized learning interventions.

challenges that are aligned with the student's current level of understanding.

- **Personalized Learning Experience:** AI will assess student responses and behaviors to create questions that target weak areas, reinforcing concepts that need further attention.
- **Automatic Feedback:** As questions are generated, AI will offer instant, tailored feedback, guiding the student through corrections and explanations.

This ensures that learning is continuous, even outside the classroom environment.

## 2. Enhancing Teacher-Student Interaction

A key objective is to foster more effective and meaningful teacher-student interactions. Teachers often face challenges in maintaining personalized communication with a large number of students. This platform will provide teachers with tools to efficiently manage interactions while ensuring each student receives personalized attention.

- **Direct Communication:** Teachers will be able to directly interact with students through the platform, offering personalized answers to questions, suggestions for improvement, and additional resources.

- **Progress Tracking:** The platform will allow teachers to track each student's academic progress, including the questions they ask, the quality of their answers, and areas where they need improvement. This allows for targeted intervention and support.

### 3. Psychological Safety and Inclusive Learning

One of the key components of the project is ensuring that the platform promotes psychological safety for students. Educational environments, especially online platforms, can often be intimidating or stressful for students, particularly those who struggle academically or face social anxieties. The project aims to create a space where students feel comfortable asking questions without fear of judgment or negative consequences.

- **Non-Judgmental Learning Environment:** The Q&A platform will be designed to encourage open and honest exchanges, ensuring that students can ask questions without feeling self-conscious. Feedback will be constructive and supportive, rather than punitive, allowing students to grow and learn without the pressure of competition.
- **Personalized Learning Paces:** The platform will allow students to engage with the material at their own pace, without feeling rushed or left behind. Students who need extra time or support will be provided with additional resources and personalized guidance from teachers.

# Tools, Hardware, and Software Used for the Website

This project utilizes a combination of frontend and backend tools, along with essential hardware infrastructure, to create a dynamic, secure, and scalable online learning platform. The website's architecture includes HTML, CSS, JavaScript for the frontend, and Django for the backend, ensuring smooth interaction between users (students and teachers), efficient data handling, and a responsive interface.

## Frontend Development Tools

The frontend is crucial as it is the user-facing part of the platform, where students and teachers interact. It must be engaging, responsive, and accessible across different devices. The frontend stack used includes:

### 1. HTML:

- HTML (Hypertext Markup Language) is the core language for creating the structure of web pages. HTML5 is used to define the basic structure of the website, including headings, paragraphs, links, forms, and multimedia elements.
- Key features of HTML5 include native support for audio, video, and interactive elements without the need for external plugins.

### 2. CSS:

- **CSS (Cascading Style Sheets)** is used to style the content defined by HTML. CSS3, the latest version, is used to implement advanced styles like animations, transitions, and responsive layouts.
- CSS frameworks like **Bootstrap** help create a mobile-first, responsive design, ensuring that the website adapts fluidly to different screen sizes—making it mobile-friendly for students and teachers.

### 3. JavaScript:

- JavaScript is the primary programming language used for creating dynamic and interactive elements on the website. JavaScript is essential for handling user input, form validation, real-time updates, and interactivity between users and the platform.
- Libraries and frameworks like **React.js** or **Vue.js** can be used for creating dynamic, single-page applications that ensure smooth and fast user experiences.

### 4. **React.js** (optional in case of dynamic content):

- **React.js** is a powerful JavaScript library for building user interfaces. It's particularly useful for creating single-page applications (SPA), where the page does not reload every time a user interacts with the platform. React.js allows for building reusable UI components, making the frontend both efficient and scalable.

## Backend Development Tools

The backend of the platform is responsible for managing user authentication, data processing, serving requests, and interacting with the database. In this project, the backend is powered by **Django**, a high-level Python web framework. The backend stack used includes:

### 1. Django:

- **Django** is a robust Python web framework that follows the Model- View-Template (MVT) architecture. Django facilitates rapid web application development with a clean and pragmatic design.
- Django provides an out-of-the-box admin interface, database management tools, and security features, making it an ideal choice for building secure and scalable web applications.
- **Django ORM (Object-Relational Mapping)**: This feature allows developers to interact with the database using Python objects rather than writing SQL queries directly. It simplifies database operations such as storing user data, retrieving questions and answers, and managing student profiles.

### 2. Django REST Framework (DRF):

- **Django REST Framework** is a toolkit for building Web APIs in Django. It simplifies the process of creating RESTful APIs, enabling seamless communication between the frontend (e.g., React.js) and backend.
- It allows the backend to handle requests from the frontend, send and receive data, and dynamically generate responses based on the user's actions, such as posting questions or submitting answers.

### 3. Authentication and Authorization:

- Django has a built-in authentication system for handling user login, registration, and password management. **JWT (JSON Web Tokens)** is integrated for stateless authentication, ensuring that users can securely log in and perform actions on the platform.

### 4. Database: PostgreSQL or SQLite

For database management, **PostgreSQL** or **SQLite** is used. PostgreSQL is a powerful, open-source relational database that is well-suited for handling large datasets, ensuring data integrity, and supporting

complex queries. SQLite, being lightweight, can also be used for smaller-scale deployments or local development.

- **Django ORM** makes it easy to perform CRUD (Create, Read, Update, Delete) operations on the database without writing complex SQL queries.

## Machine Learning and AI Tools

To enhance the user experience with personalized learning, AI-driven features like automatic question generation and real-time feedback can be integrated into the platform. This can be achieved using:

### 1. TensorFlow.js:

- **TensorFlow.js** is an open-source library that runs machine learning models directly in the browser. It is used to process real-time data, such as student performance, and generate personalized feedback or recommend learning resources.

### 2. Natural Language Processing (NLP):

- Libraries such as **spaCy** or **NLTK** can be used for analyzing text data, processing student questions, and generating intelligent responses. NLP algorithms can enhance the ability of the platform to understand and provide relevant answers to student queries, especially in natural language.

## Deployment Tools

After development, the website needs to be deployed to a cloud infrastructure to ensure high availability and scalability:

### 1. Heroku:

**Heroku** is a platform-as-a-service (PaaS) that simplifies the deployment of web applications. It allows the platform to be hosted and scaled without the need for complex infrastructure management.

- Heroku also integrates easily with GitHub for continuous integration and deployment, making updates and maintenance easier.

### 2. AWS (Amazon Web Services):

- **AWS** provides reliable, scalable, and low-cost cloud computing services. The platform can use AWS to host the backend, database, and even machine learning models.



- Services like **Amazon EC2** for server hosting, **Amazon S3** for static file storage, and **Amazon RDS** for managed database hosting ensure that the platform can handle high traffic and scale as needed.

## **Hardware**

### **1. Cloud Servers:**

The platform is hosted on cloud servers provided by **AWS** or **Heroku**. Cloud-based hosting ensures that the platform is scalable, reliable, and can handle traffic from users across the globe.

- Cloud services provide a cost-effective and flexible infrastructure, allowing the platform to scale according to user demand.

### **2. End-User Devices:**

- The website is designed to be compatible with a wide range of devices, including desktops, laptops, tablets, and smartphones. This ensures that students and teachers can access the platform regardless of their device.

# Methodology for Developing the Helpo Platform

The methodology for developing the *Helpo* platform focuses on a systematic approach to creating a robust, user-friendly online question-and-answer platform for students and teachers. This process integrates modern software development practices, ensuring scalability, security, and usability. The platform adopts the Agile Development Methodology, emphasizing iterative development, user feedback, and continuous improvement.

## 1. Requirement Analysis

The first phase involves understanding and defining the project's scope. This includes identifying the core features required by both teachers and students and establishing system requirements.

Key Activities:

1. Stakeholder Interviews: Collaborate with educators and students to gather insights into their needs, such as domain-based access for teachers and students, question-answering workflows, and user roles.
2. Functional Requirements:
  - User registration and login system.
  - Role-based access control.
  - Question creation, moderation, and answering.
  - Dashboard for students and teachers.
3. Non-functional Requirements:
  - Scalability to support large numbers of users.
  - Cross-platform compatibility for seamless access on any device.
  - Data security and user privacy.
4. Technology Stack Selection:
  - Frontend: HTML, CSS, JavaScript.
  - Backend: Django.
  - Database: PostgreSQL for scalable data handling.

## 2. System Design

This phase converts the requirements into an architecture that defines how the system will operate.

Key Components:

### 1. Frontend Design:

- Wireframes and mockups using tools like Figma or Adobe XD to visualize the user interface.
- Responsive design principles to ensure accessibility on all devices.

### 2. Backend Design:

- Using Django's Model-View-Template (MVT) framework to manage data flow between the frontend and database.
- Database Schema: Entity-Relationship (ER) diagrams are created to define tables for users, questions, answers, and feedback.

### 3. API Design:

- RESTful APIs using Django REST Framework for smooth communication between frontend and backend.

### 4. System Architecture:

- Layered architecture for modularity, including presentation, business logic, and data access layers.

Deliverables: Architectural diagrams, database schema, and API endpoints.

## 3. Implementation

The development process is divided into iterative sprints following Agile principles.

Frontend Development:

1. HTML: Building the structural layout of the platform.
2. CSS: Designing an intuitive and visually appealing user interface.

- 
- Using CSS libraries like Bootstrap for faster implementation of responsive elements.

### 3. JavaScript:

- Adding interactivity, such as form validation, dynamic question submission, and live search features.
- Optional integration with React.js for faster rendering and state management in single-page applications.

### Backend Development:

#### 1. Django Framework:

- Creating models for users, questions, and answers using Django's ORM.
- Implementing the business logic for role-based access and user management.

#### 2. Django REST Framework:

- Developing API endpoints to handle CRUD operations for questions and answers.
- Securing APIs with JWT Authentication to manage user sessions.

### Database Management:

- Configuring a PostgreSQL database to store user data, questions, and answers.
- Writing efficient queries using Django ORM to retrieve, update, and manage data seamlessly.

Deliverables: Completed modules for frontend, backend, and database integration.

## 4. Testing

Testing ensures the platform functions as intended and meets user expectations.

### Types of Testing:

#### 1. Unit Testing:

- Testing individual components like user registration, question submission, and API endpoints.

- - Tools: Pytest, Unittest.
2. Integration Testing:
    - Verifying the seamless interaction between the frontend, backend, and database.
  3. User Acceptance Testing (UAT):
    - Involving real users to test the platform and provide feedback.
  4. Performance Testing:
    - Ensuring the platform can handle high traffic, especially during peak usage times.

Deliverables: Test cases, reports, and a bug-tracking document.

## 5. Deployment

Once the testing phase is completed, the platform is deployed to a cloud-based hosting environment.

Steps:

1. Environment Setup:
  - Using Heroku or AWS for hosting the backend and database.
  - Configuring Nginx or Apache as web servers.
2. Domain Configuration:
  - Assigning separate domains for students and teachers for seamless access.
3. Static File Management:
  - Storing static files like images and CSS on Amazon S3 or a CDN for faster loading.

Deliverables: Deployed application with live domains.

## REFERENCE

Viswanathan.k.Iyer

The paper proposes an experiment in effectively teaching computer science subjects through a teacher-moderated online forum integrated with a university intranet portal. *Link:* <https://arxiv.org/abs/1701.02093>

Benzar Glen S. Grepon

Niño T. Baran, Kenn Migan Vincent C. Gumonan, Aldwin Lester M. Martinez, Mona LielE.Lacsa This paper discusses the design and implementation of an e-school system, highlighting an information systems approach to school management in a community college setting. *Link:* <https://arxiv.org/abs/2109.00198>

A.Angappan,V,RameshBabu

This paper discusses the creation of a web portal aimed at aggregating content and providing various services to a specific audience, enhancing the educational experience.

[https://www.researchgate.net/publication/228410513\\_Design\\_and\\_Development\\_of\\_an\\_Academic\\_Portal](https://www.researchgate.net/publication/228410513_Design_and_Development_of_an_Academic_Portal)