

**Project Report on :**

**Problem Statement : 03**

**AI-Powered Personal Tutor: A Scalable, Adaptive Learning System for Enhanced Student Engagement**

---

**Table of Contents**

- 1. Introduction**
- 2. Team Structure & Contributions**
- 3. System Architecture**
- 4. Key Features**
- 5. Technical Implementations**
- 6. Highlights**
- 7. Project Working**
- 8. Libraries used**
- 9. Conclusion**

**Submitted by :**

Manish Kumar M (VI Sem BCA, National College, Basavangudi)

**Team:**

Manish Kumar M(Backend),  
Shreya Pandey(Frontend),  
Ritesh Prajapati(Content),  
Syeda Bushra (Testing/Docs)

# AI-Powered Tutor System - Comprehensive Project Report

## 1. Introduction :

The AI-Powered Personal Tutor System is an innovative, scalable learning platform designed to provide personalized education through artificial intelligence. Developed as a web-based application, it addresses critical gaps in traditional education by offering adaptive learning paths, real-time AI assistance, and secure user authentication.

### Problem Statement

Traditional education systems often fail to provide individualized attention, leading to:

**Fixed learning paces** that don't accommodate all students.

Limited feedback mechanisms, leaving learners uncertain about progress.

No real-time assistance for doubts outside classroom hours.

Our solution bridges these gaps with:

1. **Adaptive Learning** : Customized course recommendations based on performance.
2. **AI-Powered Tutoring** : Instant doubt resolution via chat.
3. **Progress Tracking** : Visual analytics for courses, assignments, and quizzes.
4. **Secure Access** : Encrypted authentication protecting user data.

## 2. Team Structure & Contributions

Team Member

**Manish Kumar M (Leader)** : Backend & AI Integration, Designed system architecture Integrated Ollama/Mistral AI based on training data and also Implemented Flask API endpoints

**Shreya Pandey** : Frontend Developer, Built responsive UI (HTML/CSS/JS) such as Theme toggling (dark/light mode) Dynamic course rendering and more.

**Ritesh Prajapati** : Course Content Developer, Created course materials (AI, ML, Data Science) Designed quizzes & assignments and also Integrated reference resources.

**Syeda Bushra** : Documentation & Testing, Prepared project reports and Recorded demo videos along with Conducting user test

## 3. System Architecture

### Frontend

Technologies : HTML5, CSS3, JavaScript (Bootstrap for responsiveness).

Key Pages:

Login/Register : Secure authentication with password hashing.

Dashboard : Progress overview, enrolled courses, and assignments.

Classroom : Interactive lessons with embedded AI chat and AI classmate.

Courses : Filterable catalog (AI, ML, Data Science).

### **Backend**

Framework : Flask (Python) for API endpoints.

AI Engine : Ollama with Mistral (7B parameter LLM) for:

Real-time Q&A (`/ask` endpoint) in selected course based on RAG.

Test/assignment generation.

Data Storage : localStorage for user data storage demonstration (simulated database), later used data base (sqlite) to save the records of user.

### **Data Flow**

1. User Input → Flask API → Trained Ollama AI → Streaming Response → Frontend.

## **4. Key Features**

### **Personalized Learning**

Adaptive Paths : Recommends courses based on quiz performance.

Self-Paced Progress : Users control learning speed.

Real-Time AI Tutor and AI classmate

### **Context-Aware Responses:**

```
def get_course_context_response(question, course):  
    prompt = f"Answer this question about {course}: {question}"  
    return chat_with_ollama(prompt)
```

Streaming Replies : Mimics human-like interaction.

### **Progress Tracking**

Visual Analytics :

```
function updateProgress(course, progress) {  
    const allProgress = JSON.parse(localStorage.getItem("courseProgress")) || {};  
    allProgress[course] = progress;  
    localStorage.setItem("courseProgress", JSON.stringify(allProgress));  
}
```

Assignment Status : Submitted/completed indicators.

### **Secure Authentication**

Password Hashing : SHA-512 with salt.

Session Management : JWT tokens (simulated via `localStorage`).

## **5. Technical Implementation**

### **Frontend (Shreya Pandey)**

**Dynamic UI :**

- Theme toggling (CSS variables).
- Course cards with hover effects.
- learning tips with dynamic messages
- Homework section aligned with selected course
- Course completion and clear section

#### **Interactive Elements :**

- Quiz auto-grading.
- Assignment submission forms.

#### **Backend & AI (Manish Kumar M) :**

- Data preparation for RAG model
- Course meta data preparation in json formate
- Local server set up to route prompts and response from front end to backend

#### **- Flask API:**

```
@app.route("/ask", methods=["GET"])
def ask_ai():
    question = request.args.get("question")
    response = ollama.generate(question)
    return Response(stream_with_context(response))
```

- Prompt Engineering for fintued response
- AI Prompts : Customized for course context.

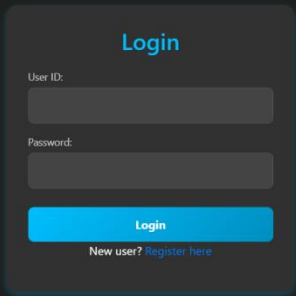
#### **Course Content (Ritesh Prajapati) :**

##### **- Structured Modules:**

- **Introduction to AI:** History, ethics, applications , advantages and types in genreal.
- **Machine Learning:** Algorithms, case studies.
- **Advance concepts of AI :** delving deep into Advance AI tiopics
- **Quizzes :** MCQs with instant feedback.
- **Assignments :** Dynamic assignment for selected courses

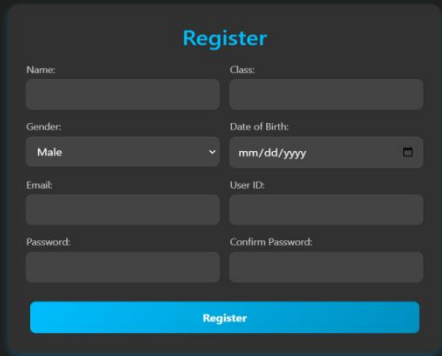
## 6. Highlights of our project and discription:

### - Login Pgae

A login form with a dark background. The title "Login" is in blue. Below it are two input fields: "User ID:" and "Password:". A blue "Login" button is at the bottom. Below the button is a link "New user? Register here" in blue.

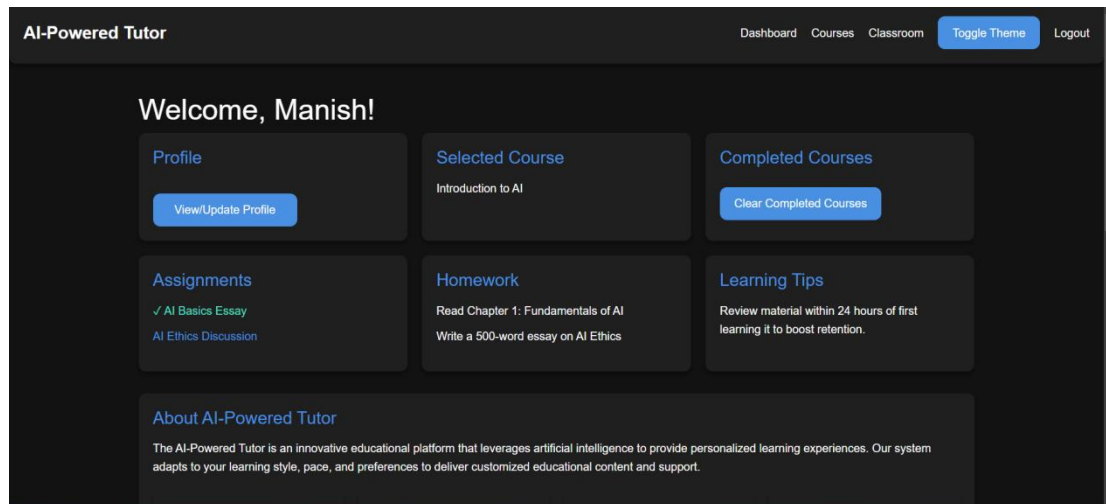
A basic and clean UI for secure User Login . For uninterrupted demonstration we have used Local storage but also we have another option with sqlite , which become a top notch

### - Registration Page

A registration form with a dark background. The title "Register" is in blue. Below it are several input fields: "Name:", "Class:", "Gender:" (with a dropdown menu showing "Male"), "Date of Birth:" (with a date picker showing "mm/dd/yyyy"), "Email:", "User ID:", "Password:", and "Confirm Password:". A blue "Register" button is at the bottom.

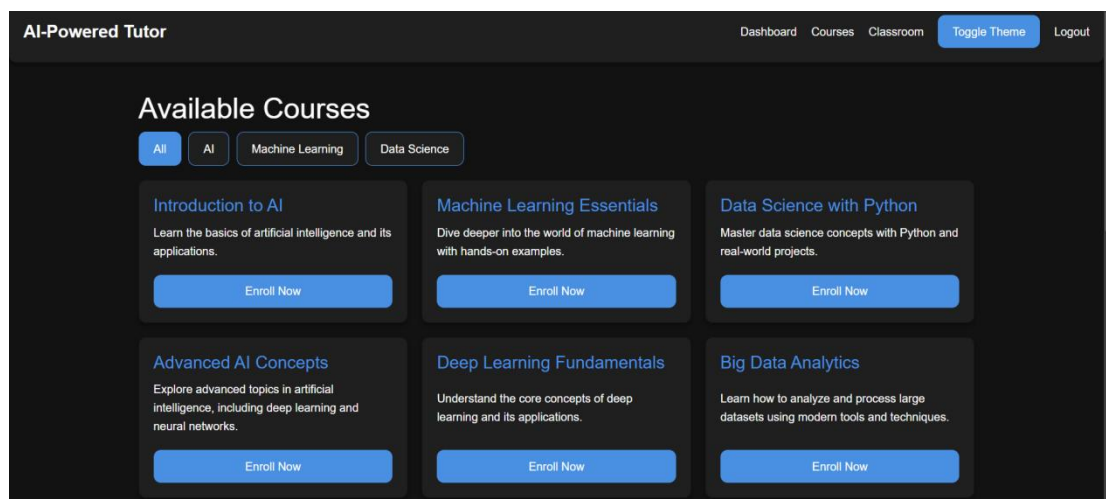
Another clean and elegant UI for user registration containing necessary fields like Name , Class Gender , DOB , email , Uid , and password. Here we have implemented logics to check if the user already exists , or check the duplication if Uid and email field.

## -User Home Page/ Dashboard



Dashboard greet with user name , and contains several dynamic sections such as profile for any updates , selected sourse section that display the current course , and once the course is completed it gets moved to completed course section. Cards of Assignment , home works changes dynamically acording to slection of course , and a helpful motivating card for tips to learners.

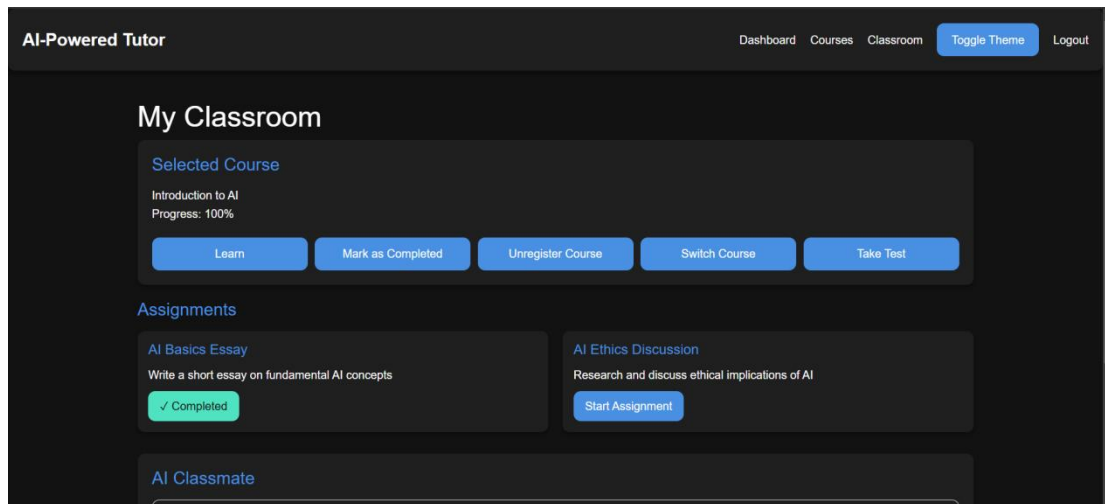
## -Course Page



This page has a buch of catalogs specifying the group of courses that we have added. Such as AI , Machine Learning and Data science for this project and demonstration.

Since here only few courses are developed in detail and rest are in progress , so if user clicks on those course , we still manage that by displaying a **course in development** message to user

## -Classroom Page



This is the page where user learns and access to materials of selected course. We made sure that we don't miss out any basic and important functionalities that may hamper the learning and evaluating experience.

There are handy navigations at one place such as learn , completion , unregister course if not interested and then switch directly to course page from next button... and if course selected they might take the test and assignment at any time.

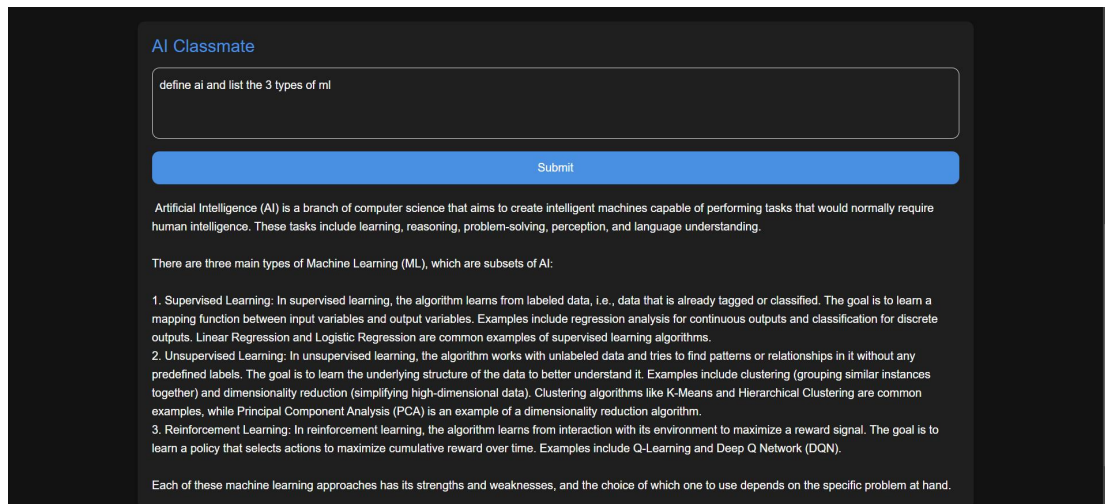
## -User course request

The screenshot shows a form titled 'Can't Find What You're Looking For?'. The subtitle is 'Request a new course or suggest a topic you'd like us to add to our curriculum.' The form has four main sections: 1. 'Course Topic/Title' with a text input field containing 'e.g., Computer Vision Fundamentals'. 2. 'Category' with a dropdown menu showing 'Select a category'. 3. 'Tell us more about what you're looking for' with a large text area containing the placeholder 'Describe the course content, learning objectives, or specific skills you want to acquire'. 4. 'Your Email (optional for updates)' with a text input field containing 'your@email.com'. At the bottom is a green 'Submit Request' button.

Also we made sure to add another functionality where if user doesn't find course of its interest , they still have the option to request a course to be developed.

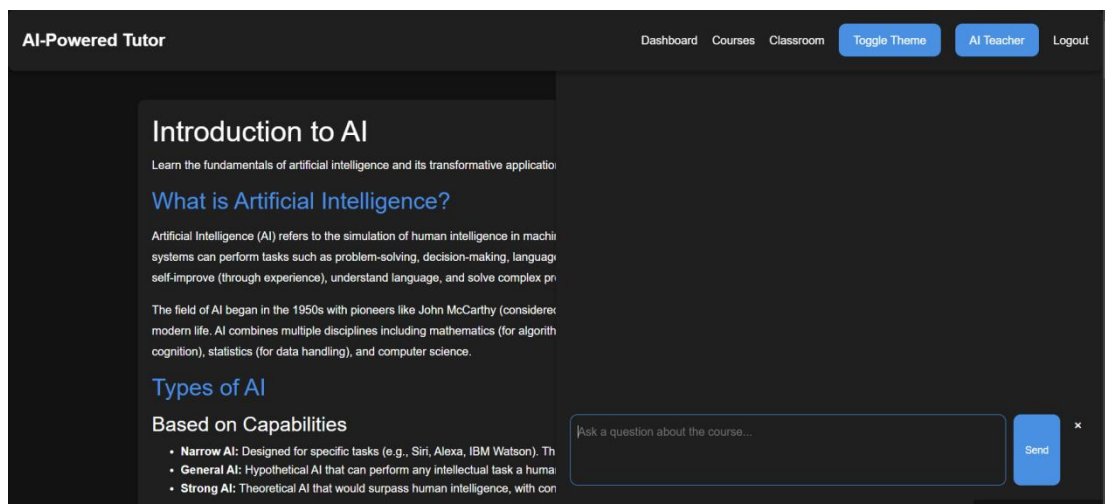
The field inside this card make sure to collect enough information regarding the requested course that may ease the process of course developers

## - AI Classmate



This project also aims to develop a good interactive environment where learners must not get bored in the course completion journey, thus we have added a classmate powered by AI for user, where the learners can chat, learn and stay updated with new trends and tech with their conversation and queries.

## -AI Teacher :



While we have added an AI powered Classmate for general conversation, but when user is in learning page, we made sure that to another AI assistant teacher which is basically trained on that particular course, leading to more catered responses on course-related queries. This model is trained using models of RAG and Mistral AI for more appropriate responses.



## 7. How the Project Works

The **AI-Powered Personal Tutor System** is a comprehensive web-based learning platform designed to deliver personalized education through artificial intelligence. The system begins with a **secure authentication** process, where users register or log in with encrypted credentials, ensuring data privacy. Upon accessing the **dashboard**, students view their enrolled courses, progress metrics, and upcoming assignments. The **course selection** feature allows users to browse and filter subjects (AI, ML, Data Science) and enroll in self-paced learning paths.

Within the **classroom interface**, students engage with structured course content, including interactive lessons, quizzes, and reference materials. The **AI Teacher**, powered by **Retrieval-Augmented Generation (RAG)**, provides real-time, context-aware assistance by retrieving answers from curated course materials, ensuring accuracy. Meanwhile, the **AI Classmate** offers general study support, such as motivation tips and productivity advice, without relying on course-specific data.

Progress is tracked via **visual indicators**, including completion percentages and achievement badges. The **assignment system** lets users submit tasks, receive AI-generated feedback, and monitor grading status. Additionally, the platform recommends **adaptive learning resources** (videos, articles, quizzes) based on performance.

By combining **personalized learning**, **AI-driven tutoring**, and **secure data management**, the system creates a dynamic, student-centric educational experience that adapts to individual needs.

## 8. Libraries and Technologies used :

**Bootstrap 4.5** : For responsive UI  
**JavaScript ES6+** : For core logic and dynamic implementation  
**Flask** : for creating local api end points  
**Requests** : For handling API call requests  
**JSON** : Used for data seialization for API responses  
**Ollama** : Inorder to host Mistral model  
**FAISS** : Used in building rag model for vector database initialisation  
**Crypto-JS** : For password hashing  
**Web Storage API** : for Local storages  
And more...

## 9. Conclusion :

The **AI-Powered Tutor System** successfully demonstrates:

✓ **Secure authentication** by sqlite and local storage(client side)

- ✓ **Personalized learning** through adaptive courses.
- ✓ **Instant AI assistance** via Mistral LLM and RAG based model.
- ✓ **Robust progress tracking** with visual analytics.

**Impact :** This prototype lays the foundation for scalable, AI-driven education tools that can democratize access to personalized learning.

**Appendix :**

- **GitHub Repo** : [[https://github.com/Manish-909/AI\\_Powered\\_tutor.git](https://github.com/Manish-909/AI_Powered_tutor.git)]
- **Demo Video** : [[https://github.com/Manish-909/AI\\_Powered\\_tutor/blob/4a1b9808c63cb47af49c102d887f5110256eecfe/demo\\_\\_vedio.mp4](https://github.com/Manish-909/AI_Powered_tutor/blob/4a1b9808c63cb47af49c102d887f5110256eecfe/demo__vedio.mp4)]

**Submitted on :** 30|March|2025