

A Project Presentation on AI Powered E-learning platform.

Presented By

Priyanshu Raj (2100970130085)

Saurav Tiwari (2100970130101)

Shefali Singh (2100970130104)

Guided By

Ms. Sakshi Malhotra

(Assistant Professor)



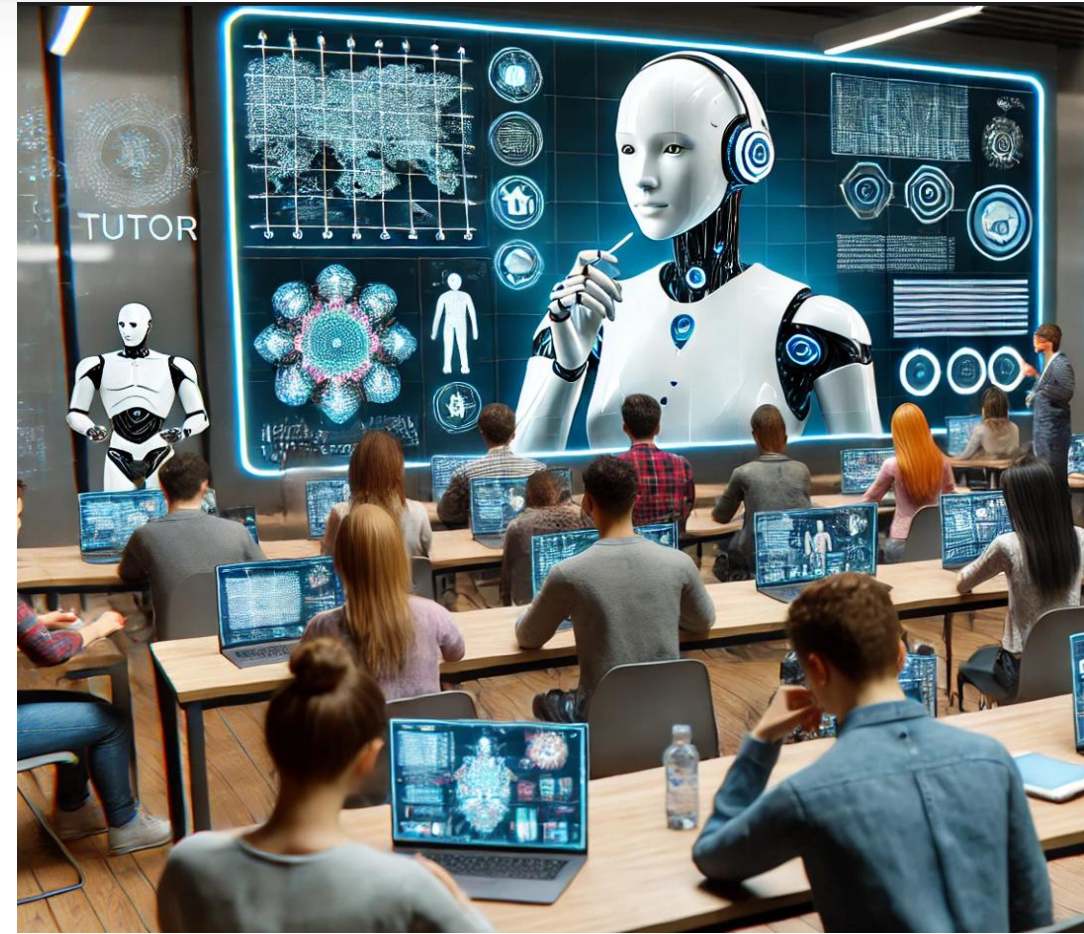
Department of Information Technology

Galgotias College of Engineering & Technology , Greater Noida

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh

Content

- **Introduction**
- **Implementation of Source Code**
- **Functionality of modules**
- **Testing of modules**
- **Module/ system Integration**
- **Detail Description of Concept and Technical Knowledge**



Introduction

1. Education is evolving, and traditional e-learning platforms often struggle with engagement, personalization, and real-time feedback.
2. Our AI-powered e-learning platform bridges this gap by offering adaptive learning experiences, AI-generated content, and instant assessments.
3. With the help of artificial intelligence, machine learning, and automation, we create dynamic, interactive, and customized courses tailored to each learner.
4. Unlike conventional platforms, our system analyzes student progress in real-time, adjusting course difficulty, explanations, and assessments to maximize understanding.
5. This innovation ensures that learning is no longer one-size-fits-all—instead, every student gets a personalized tutor-like experience powered by AI.

Source Code

1. Frontend - User Input Form (TypeScript)

User Input Form for Topic Name

Here, the **frontend** collects the topic name from the user using a form. It uses **React (with TypeScript)** to manage the state of the input and sends it to the backend via **Axios** for further processing.

```
labnine | Edit | Explain
const CourseGenerationForm: React.FC = () => {
  const [topic, setTopic] = useState<string>('');

  const handleSubmit = async (event: React.FormEvent) => {
    event.preventDefault();
    try {
      const response = await axios.post('http://localhost:5000/generate-course', {
        topic: topic,
      });
      console.log(response.data);
    } catch (error) {
      console.error("Error generating course:", error);
    }
  };
};
```

Source Code

2. Backend Code (JavaScript)

Backend: Input from User and Passing to AI Model

The **backend** receives the user's input (the topic name) and sends it to the AI model (via an API like OpenAI's GPT). The following is the **Node.js/Express code** that takes the user's input, processes it, and calls the model.

```
app.post("/generate-course", async (req, res) => {
  try {
    const { topic } = req.body;
    if (!topic) {
      return res.status(400).json({ error: "Topic is required" });
    }
    const prompt = `
      Generate a structured AI-powered course for the topic: "${topic}".
      The course should have 3 modules:
      1. **Module 1: Introduction & Fundamentals**
        - Provide an **introductory explanation** of the topic in a structured, easy-to-read format.
        - Cover the **key concepts** with real-world examples.
      2. **Module 2: Advanced Concepts & Study Material**
        - Explain **more in-depth details** of the topic with structured study material.
        - Provide a **step-by-step guide or important formulas, key takeaways, and reference links**.
      3. **Module 3: MCQ Assignment**
        - Generate **5 multiple-choice questions (MCQs)** based on the topic.
        - Each question should have **4 options** (A, B, C, D) with one correct answer.
        - At the end, provide an **answer key**.
      Format the response in structured JSON with keys: "module1", "module2", "module3".
    `;
  }
});
```

Source Code

3. Model Response Generation : Using Llama

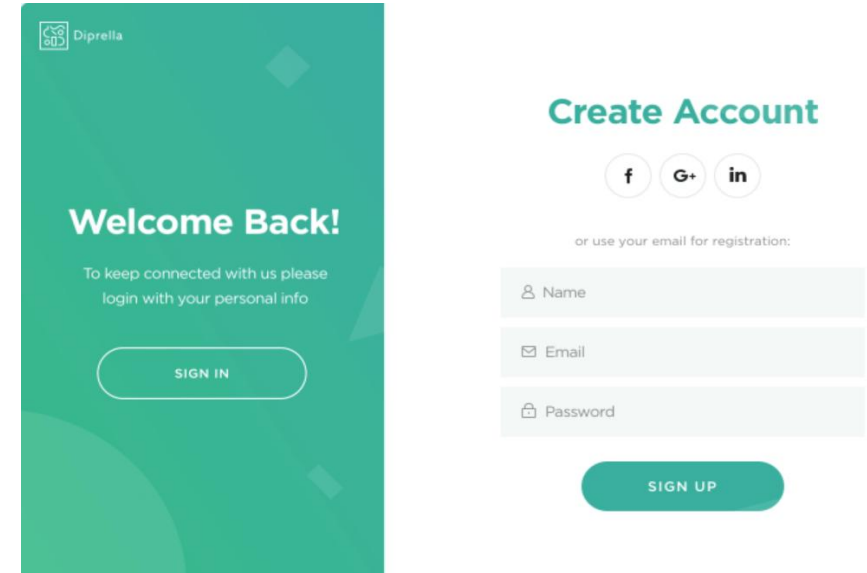
The AI model, implemented in Python (using OpenAI GPT), processes the request from the backend and generates the course content.

```
tabnine | Edit | Test | Explain | Document
def generate_course(prompt):
    response = openai.Completion.create(
        model="text-davinci-003",
        prompt=prompt,
        max_tokens=1500,
        temperature=0.7,
    )
    return response.choices[0].text.strip()
```

Functionality of Modules

1. User Module

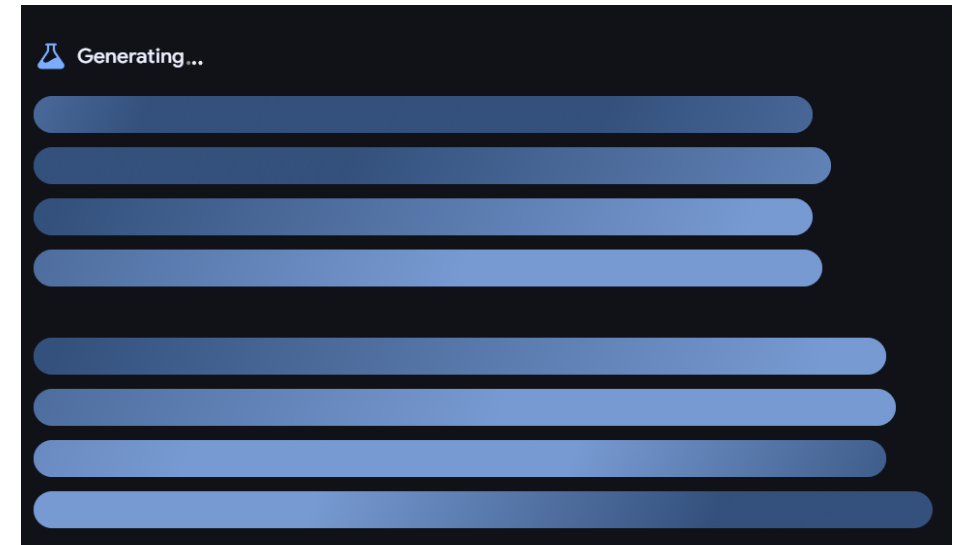
- Login screen or user registration form (illustrating user authentication).
- Secure login process (illustrating password hashing or JWT tokens).



The image displays two user interface screens for a module named 'Diprella'. The left screen is a login page with a teal background, featuring the text 'Welcome Back!' and a 'SIGN IN' button. The right screen is a 'Create Account' page with a white background, showing social login options (Facebook, Google+, LinkedIn) and a registration form with fields for Name, Email, and Password, followed by a 'SIGN UP' button.

2. AI-Powered Content Generation Module

- Takes a topic input from the user and generates a comprehensive course, including lectures, study materials, examples, and quizzes using GPT-4 or other AI models.
- The course is structured into modules that may include study material, interactive elements, and MCQ assignments.

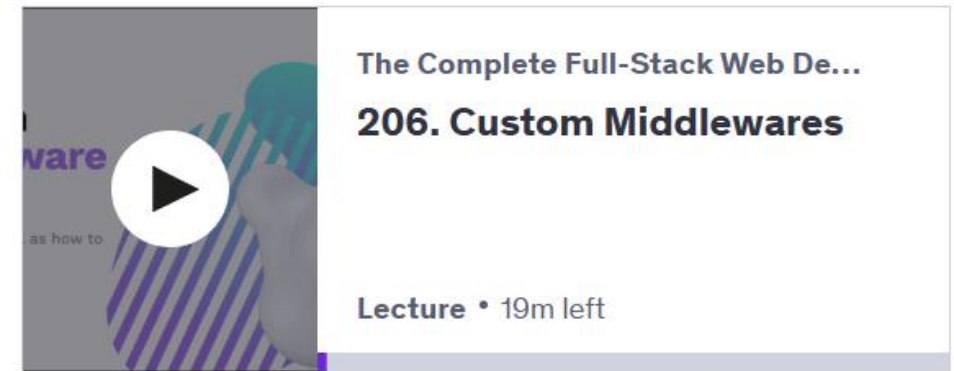


Functionality of Modules

Let's start learning

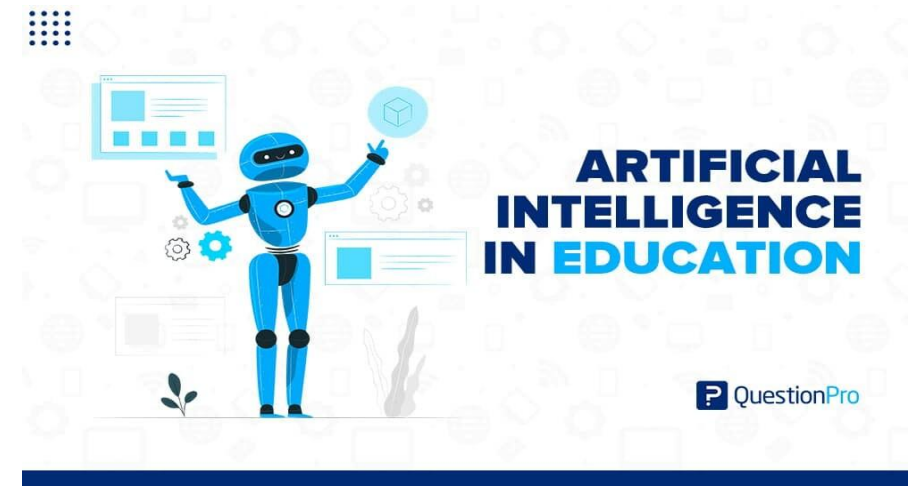
3. Course Management Module

- Instructors can create, manage, and update courses, which include lessons, quizzes, and study materials.
- Ability to upload multimedia content such as videos, images, and slides.



4. Adaptive Learning & AI Personalization Module

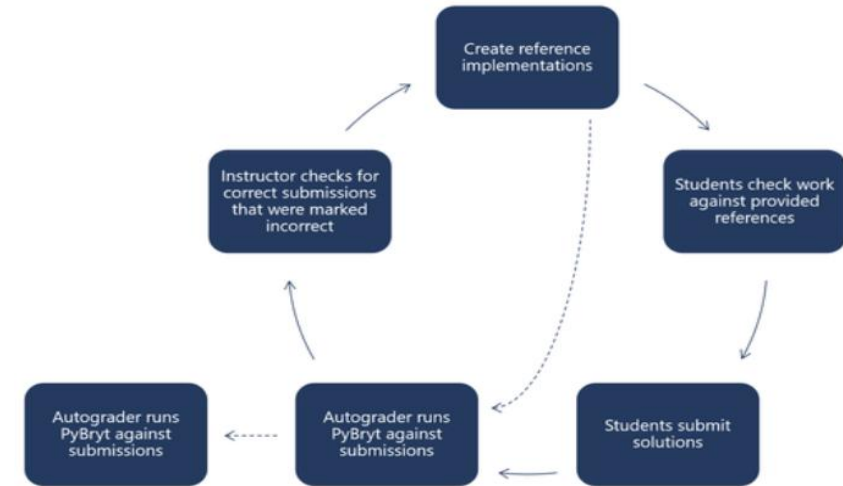
- AI adapts the course flow and content based on a learner's previous activity, quiz results, and performance.
- The course is structured into modules that may include study material, interactive elements, and MCQ assignments.



Functionality of Modules

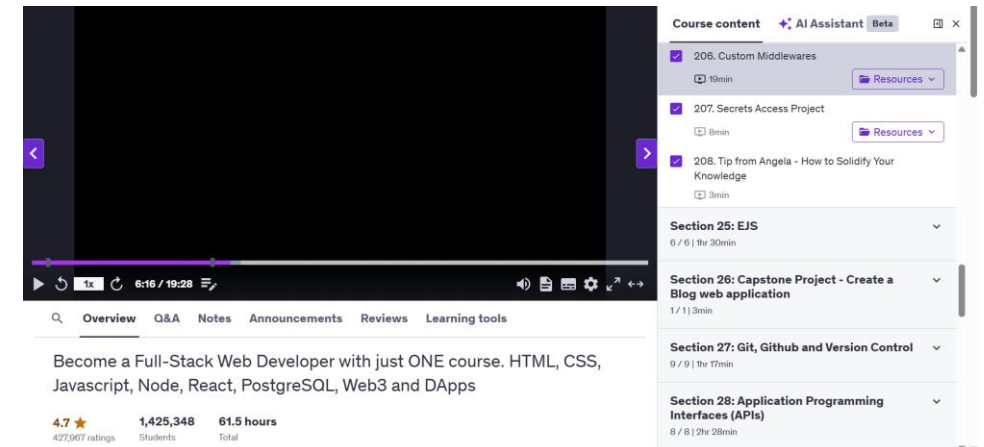
5. Assessment & Automated Grading Module

- The AI generates multiple-choice questions (MCQs) and short-answer questions for the student based on course content.
- The system automatically grades MCQs and provides instant feedback. For short-answer questions, the AI evaluates answers based on accuracy, relevance, and language.



6. Video Streaming & AI-Lecture Module

- AI generates video lectures with a virtual avatar teaching the course material.
- Voiceovers for AI-generated content are synchronized with animated visuals (diagrams, charts, and real-time annotations).



Functionality of Modules

7. Blockchain Certification

- After successful completion of course, user will be certified from our platform and this certification will be based on smart contract & blockchain.
- This certification provides value to course and gives a sense of validation to learner



Testing of Modules

1. Unit Testing:

- Unit tests are performed to validate individual functions, such as the logic for generating questions, the response of the AI chatbot, and user authentication.

2. Integration Testing:

- Integration testing is done to ensure that multiple systems (e.g., frontend, backend, AI models) work together seamlessly. For example, testing how data from the frontend is passed to the backend and then used by the AI model.

3. End-to-End Testing:

- End-to-end testing ensures that the entire user journey works as expected. For example, testing a user's journey from registering, selecting a course, interacting with the AI tutor, taking an assessment, and receiving feedback.

4. Performance Testing:

- This is done to ensure that the platform can handle high traffic, especially when AI models are used to generate content and assess learners in real time.

5. User Acceptance Testing (UAT):

- Real users test the platform to ensure that it meets their expectations and is user-friendly. Feedback is collected, and improvements are made accordingly.

Module/System Integration

1. Understanding the Modules

Each module in the platform provides a distinct functionality, but they need to be connected to ensure smooth flow and a **cohesive user experience**. Below are the key modules that need to be integrated:

Core Modules:

- **User Management Module:** Handles user registration, login, and profile management.
- **AI-Powered Content Generation Module:** Generates course content, study material, and quizzes using AI.
- **Course Management Module:** Enables instructors to create, manage, and organize courses.
- **Adaptive Learning & Personalization Module:** Adjusts course difficulty based on student progress.
- **Assessment & Automated Grading Module:** Creates and grades assessments automatically.
- **Video Streaming & AI-Lecture Module:** Manages video-based content delivery.
- **Admin & Analytics Module:** Provides admin dashboards and learning analytics.
- **Blockchain Certifications :** Certification that will add value to learners course

Module/System Integration

2. Key Steps in Module/System Integration

a. Data Flow between Modules

- User Management needs to interact with Course Management to store the learner's progress.

b. API Integration

- Api request to make connection live between Frontend, Backend and our model

c. Database Connection

- To store user data, course work, other necessary data we need database to store that.

d. Third Party Integration

- Video Hosting and we need AWS S3 for hosting our course Related videos
- Speech-to-Text & Text-to-Speech APIs: Integration with services like Google TTS/Whisper API for providing voiceover content and real-time speech recognition.

Module/System Integration

3. System Integration Workflow Example

1. User Input (Frontend) : Frontend sends the input to the Backend API.
2. Content Generation (Backend - AI Module): AI generates a structured course, including text content, video scripts, and MCQs.
3. Course Content & Personalization (Backend - Adaptive Learning):The module personalizes the course based on the learner's previous activity and learning pace.
4. Real-time Interaction (AI Chatbot & Virtual Tutor):The chatbot fetches relevant content from the course material database and responds in real-time.
5. Assessment & Feedback: The Assessment & Grading Module uses AI to grade and provide instant feedback.
6. Progress Tracking & Gamification : The Gamification & Progress Tracking Module monitors student progress and rewards achievements.
7. Blockchain Certifications : This certification provides value to course and gives a sense of validation to learner.

Detailed Description and Technical Knowledge

1. Concept Overview

AI-Driven Content Generation

- **Objective:** The main idea is to automate the creation of educational content based on the user's input (e.g., a topic name), allowing for instant course generation on virtually any subject.
- **Modules Involved:**
 - **Module 1 & 2 (Content Creation):** AI generates structured educational materials like lecture notes, study material, and summaries using large language models (LLMs) like OpenAI's GPT-4.
 - **Module 3 (Assessment Generation):** AI formulates multiple-choice questions (MCQs) and quiz assignments based on the generated content, creating a comprehensive learning experience.
 - **Module 4 (Interactive Learning):** AI personalizes content dynamically based on learner progress, adapting the complexity and providing real-time feedback.

Real-Time Personalization

- **Adaptive Learning Paths:** The AI continuously adapts to individual learner needs. Based on performance in tests, quizzes, and assessments, the AI adjusts the learning pace, difficulty, and even suggests different learning materials or alternative explanations.
- **Personalized Recommendations:** By analyzing data such as quiz scores, time spent on specific topics, and learning preferences, the platform recommends related content, enabling the learner to expand knowledge in a targeted way.

Detailed Description and Technical Knowledge

2. Technical Knowledge and Implementation

a. OpenAI GPT-4 (or GPT-3.5) for Content Generation

- **Content Generation API:**

- **Input:** The user provides a topic (e.g., “Basics of Thermodynamics”). The system sends a request to our AI model with a prompt asking for an AI-generated course on that topic.
- **Output:** The API generates structured content, including an introduction, key concepts, detailed explanations, examples, and more, organized into the required modules.

b. AI-Generated MCQs and Assessments

- **Training the AI for Assessment Generation:**

- The AI is trained to recognize key concepts from the content (e.g., “laws of thermodynamics”) and create **questions** with multiple-choice answers.
- The model ensures question diversity, generating questions on **concepts, definitions, formulas**, and **real-world applications**.
- **MCQ Example:** After reading about “thermodynamics,” the system could create the following multiple-choice question:

```
"question": "What is the First Law of  
Thermodynamics?",  
"options": {  
  "A": "Energy cannot be created or  
destroyed, only transferred or  
transformed.",  
  "B": "Entropy of a closed system  
always increases.",  
  "C": "Heat is a form of energy.",  
  "D": "Work is a form of energy."  
},  
"answer": "A"
```


Detailed Description and Technical Knowledge

2. Technical Knowledge and Implementation

c. Speech-to-Text & Text-to-Speech for Accessibility

- **Text-to-Speech (TTS)** technologies like **Google TTS** or **Coqui TTS** convert text-based educational content into **audio format**, making it accessible to **visually impaired learners** or those who prefer audio content.
- **Speech-to-Text (STT)** systems like **Whisper AI** convert **spoken questions** from learners into **text**, allowing the AI to respond through a **natural conversation interface**.

d. Interactive Video Generation

- **AI-generated Video Lectures:**
 - The platform uses tools like Synthesia AI or D-ID to create virtual avatars that act as instructors. These avatars can deliver the AI-generated course content via video, making the content more engaging.
 - The system uses scripts generated from AI and synchronizes them with visual content such as slides, diagrams, and animations to make learning more interactive and engaging.

e. Blockchain Certification :

- After successful completion of course, user will be certified from our platform and this certification will be based on smart contract & blockchain.
- This certification provides value to course and gives a sense of validation to learner