**Project Synopsis**

**AI-Chatbot for Student Queries**

# Problem Statement:

Traditional e-learning platforms often provide static content delivery without personalized guidance. Students may struggle to understand course material, especially when learning independently, as there is **no interactive tutor to clarify doubts**. Existing solutions lack an intelligent assistant that can dynamically explain uploaded notes, adapt to students’ queries, and provide a personalized learning experience. This creates a gap between static online content and interactive, classroom-style teaching.

# Proposed Solution:

The E-Learning Platform with AI Tutor is a **web-based learning management system (LMS)** enhanced with an **AI-powered chatbot** that acts as a virtual tutor. The backend, built with **Spring Boot**, handles course management, authentication, and content delivery. The **MySQL database** stores user information, courses, and uploaded notes. The frontend, developed in **React**, provides an intuitive UI for course browsing, lesson interaction, and chatbot communication. The AI Tutor, powered by GPT-based models, processes the uploaded notes and can **explain concepts, answer student queries, and provide contextual learning assistance**.

# Objectives:

* To create an **interactive, personalized learning experience** beyond traditional e-learning.
* To assist students in **understanding complex topics** through an AI-driven tutor chatbot.
* To provide educators with a platform to **upload, manage, and share course material** efficiently.
* To integrate **AI and NLP** into education for scalable, real-world impact.

# Methodology:

* **Backend (Spring Boot):**
  + Manage users (students/instructors), courses, modules, and uploaded content.
  + REST APIs for authentication, course retrieval, and AI chatbot queries.
  + Integration with MySQL for relational data handling.
* **AI Summarization Engine:**
  + Use retrained transformer models such as **BERT (extractive summarization)** or **Pegasus (abstractive summarization)**.
  + Implement  text preprocessing, tokenization, and summary generation pipelines.
  + Optimize model inference for scalability (via Hugging Face Transformers or TensorFlow).
* **Frontend (React):**
  + Responsive UI for students to browse courses, access notes, and interact with the chatbot.
  + Instructor dashboard for uploading notes, managing students, and monitoring progress.
* **AI Tutor Chatbot:**
  + Use GPT-based models (via OpenAI API or fine-tuned transformers).
  + Preprocess uploaded notes and enable chatbot to reference this content when answering.
  + Support contextual responses, topic explanations, and clarifications.
* **Database (MySQL)**:
  + Store users, roles, course structures, uploaded notes, and chat history.
  + Maintain strict relationships (e.g., Instructor → Courses → Students).

# Expected Outcome:

* **A functional e-learning platform with course management and interactive AI tutoring.**
* **Students can learn faster and better with personalized explanations of uploaded content.**
* Instructors can **digitally manage and distribute learning materials** with ease.
* A scalable solution that can later include features like **voice-enabled tutors, adaptive learning paths, and integration with existing LMS systems**.