Full Stack Java Application with AI Integration and AWS Deployment

Mini Project 1

Module: Database Basics (Mysql)
Project Title: Course Catalog Database

Project Overview
Design a simple database to store course details (title, description, price, instructor, category, image). Students will practice creating tables, inserting records, and querying data.

Objective
To create and manage a database for storing course listings. Students will practice:

- Designing a relational schema for courses
- CRUD operations with SQL
- Querying by filters (e.g., category, price range)

Scope of the Project

- Create courses table with fields: id, title, description, price, instructor, category, image_url
- Insert sample course data
- Retrieve courses by category or price
- Update and delete records

Technical Requirements

- Language: SQL
- Database: Mysql

Mini Project 2

Module: Spring Boot Basics
Project Title: Course API

Project Overview
Build a REST API using Spring Boot to manage the course database. Implement routes for adding, updating, deleting, and fetching course details.

Objective
To create an API layer between the database and frontend. Students will practice:

- RESTful route design
- Service and Repository layers
- Error handling and validation

Scope of the Project
Routes:

- GET /courses
- GET /courses/{id}
- POST /courses
- PUT /courses/{id}
- DELETE /courses/{id}
- Data validation for title, price, and category
- Global exception handling

Technical Requirements

- Language: Java
- Framework: Spring Boot, Spring Data JPA
- Database: PostgreSQL
- Tools: Maven, Postman

Mini Project 3

Module: Authentication
Project Title: User Login & Roles

Project Overview
Implement user authentication so that only logged-in users can add/edit/delete courses. Admins have full access; normal users can view courses.

Objective

- User registration/login
- Role-based access control (ADMIN, USER)
- Password encryption

Scope of the Project
Endpoints:

- POST /auth/register
- POST /auth/login
- Protect course modification routes with roles
- Store hashed passwords

Technical Requirements

- Spring Security, JWT
- BCryptPasswordEncoder

Mini Project 4

Module: Course Search & Filters
Project Title: Search Functionality

Project Overview
Add search and filter options so users can find courses by keyword, category, or price range.

Objective

- Use query parameters for searching
- Implement filtering logic in repository queries

Scope of the Project

- GET /courses?category=Java&minPrice=1000&maxPrice=3000
- Full-text search on title/description

Technical Requirements

- Spring Data JPA query methods / @Query

Mini Project 5

Module: React Basics
Project Title: Course Dashboard Frontend

Project Overview
Develop a React-based interface to view available courses. Users can browse the course catalog, search by filters, and view detailed information about each course.

Objective
To build a clean, responsive frontend that fetches data from the Spring Boot backend API. Students will practice:

- Component-based architecture
- State and props management
- Fetching and displaying API data
- Event handling for search and filtering

Scope of the Project

- Display course listings in a grid/list view
- Search/filter courses by category, price, or keyword
- View individual course details including description, instructor, and price
- Responsive design for mobile and desktop users

Technical Requirements

- Language: JavaScript
- Framework: React.js (with React Router for navigation)
- Tools: VS Code, Browser DevTools, Axios/Fetch for API calls

Mini Project 6

Module: Full Spring Boot Integration
Project Title: Online Course Management Platform

Project Overview
Combine all previous mini projects into one complete Spring Boot full-stack application where users can sign up, browse available courses, enroll, and for admins — add, edit, or delete courses.

Objective
To integrate all modules into a working platform. Students will practice:

- Full-stack development with Spring Boot backend and a frontend React
- User authentication and authorization with JWT
- Connecting the frontend to Spring Boot REST APIs
- Handling form submissions and file uploads for course images/materials

Scope of the Project

- User authentication (JWT-based)
- Course creation with image upload (Admin only)
- Browsing, searching, and viewing courses (Public/Logged-in users)
- Editing and deleting courses by admin
- Enrollment management for users

Technical Requirements

- Language: Java (Backend), React
- Backend: Spring Boot, Spring Data JPA, Spring Security (JWT)
- Database: PostgreSQL or MySQL
- Frontend: React.js
- Tools: IntelliJ IDEA / VS Code, Postman, Docker (optional)

Mini Project 7

Module: AI Chatbot Integration
Project Title: AI-Powered Course Assistance Chatbot

Project Overview
Integrate an AI chatbot into the course management platform to answer student queries, provide course details, and guide users through the enrollment process — available 24/7. The chatbot will be connected to relevant data such as FAQs, course information, and platform policies.

Objective
To implement a conversational AI assistant that can:

- Answer general questions about the platform
- Provide details about specific courses from the database
- Assist instructors with adding courses
- Offer learners guidance on finding and enrolling in courses

Students will learn:
· Using AI APIs (e.g., OpenAI GPT API)
· Integrating chatbot UI in the frontend React
· Connecting chatbot to backend course data
· Handling real-time messaging

Scope of the Project

- Chatbot interface embedded in the frontend (floating widget)
- Backend API endpoint to handle chatbot queries
- Integration with AI API for natural language understanding and responses
- Contextual responses by querying the course database
- Fallback to FAQ database for unanswered queries

Technical Requirements

- Language: Java (Backend), JavaScript using React
- Backend: Spring Boot REST API
- Database: PostgreSQL/MySQL for course data, optional separate table for FAQs
- AI API: OpenAI GPT (or similar LLM API)

Tools:

- React Chatbot UI library
- Postman for testing
- API keys for AI provider
  · Concepts Used: API integration, NLP basics, secure data access

Mini Project 8

Module: Version Control with Git & GitHub
Project Title: Course Platform Source Code Management

Project Overview
Set up Git for the online course platform project and push code to GitHub. Students will learn the complete workflow for managing source code in teams, tracking changes, and collaborating effectively.

Objective
To introduce students to Git and GitHub for version control, ensuring:

- Code history tracking and rollback
- Collaboration through branches and pull requests
- Handling and resolving merge conflicts
- Maintaining a professional project repository structure

Scope of the Project

- Initialize a local Git repository for the Spring Boot project
- Create .gitignore to exclude target/, .env, and other sensitive files
- Commit and push code to GitHub repository
- Create feature branches (e.g., feature/course-upload)
- Merge changes via pull requests with proper code review
- Use GitHub Issues for tracking tasks and bugs

Technical Requirements

- Tools: Git, GitHub, IntelliJ IDEA / VS Code
- Concepts Used: Git basics, branching, merging, pull requests, issue tracking

Mini Project 9

Module: Cloud Deployment
Project Title: Deploying Online Course Platform to the Cloud

Project Overview
Deploy the complete **Spring Boot + AI Chatbot** course platform to a cloud platform, making it accessible to users worldwide. Students will configure environment variables, connect to a live relational database, and ensure production readiness.

Objective
To host the application on a cloud platform so users can:

- Access the course catalog via a public URL
- Use the backend API in a live environment
- Interact with the AI chatbot in real-time

Students will practice:

- Building the frontend for production
- Deploying Spring Boot backend with cloud hosting
- Connecting to a managed cloud database
- Managing secrets and environment variables securely

Scope of the Project

- Deploy React frontend (Vercel, Netlify, or similar) or serve via Spring Boot static resources
- Deploy Spring Boot backend (Render, Railway, AWS EC2)
- Connect backend to PostgreSQL/MySQL cloud database
- Configure API keys for AI chatbot securely in the environment
- Test live application using Postman and browser

Technical Requirements
Language: Java (Spring Boot backend), JavaScript (React frontend if used)
Tools:

- GitHub (for source repository)
- Vercel/Netlify (frontend)
- Render/Railway/AWS EC2 (backend)
- PostgreSQL/MySQL Cloud (database)
- AI API provider (OpenAI or similar)
- Concepts Used: CI/CD basics, environment configuration, cloud hosting