AI Course Generator

A

Report submitted in partial fulfilment of the requirement for the

degree of

B.Tech.

In

Computer Science & Engineering

By

Anannya Singh (2301640100075)
Anshika Shukla (2301640100093)
Amit Yadav (2301640100071)
Ankit Yadav (2301640100085)
Anurag Sharma (2301640100102)
Aman Kumar Tiwari (2301640100061)

Under the guidance of
Kirti Rathore
Assistant Professor
Project Id: 26_CS_3B_03

Pranveer Singh Institute of Technology, Kanpur
Dr A P J A K Technical University
Lucknow

DECLARATION

This is to certify that Report entitled “AI Course Generator ”which is submitted by us in partial fulfilment of the requirement for the award of degree B.Tech. in Computer Science and Engineering to Pranveer Singh Institute of Technology, Kanpur under Dr. A. P. J. Abdul Kalam Technical University, Lucknow comprises only our own work and due acknowledgement has been made in the text to all other material used.

Date:

Anannya Singh	(2301640100075)
Anshika Shukla	(2301640100093)
Amit Yadav	(2301640100071)
Ankit Yadav	(2301640100085)
Anurag Sharma	(2301640100102)
Aman Kumar Tiwari	(2301640100061)

CERTIFICATE

This is to certify that Report entitled"**AI Course Generator**" which is submitted by Anannya Singh (2301640100075), Anshika Shukla (2301640100093), Ankit Yadav (2301640100085), Amit Yadav (2301640100071), Anurag Sharma (2301640100102), and Aman Kumar Tiwari (2301640100061) in partial fulfilment of the requirement for the award of degree B.Tech. in Computer Science & Engineering to Pranveer Singh Institute of Technology, Kanpur affiliated to Dr. A P J Abdul Kalam Technical University, Lucknow is a record of the candidate own work carried out by him under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Signature:	Signature:
Dr. Nand Kishore Sharma Head CSE Department, PSIT, Kanpur	Kirti Rathore Assistant Professor CSE Department, PSIT, Kanpur

-

ACKNOWLEDGEMENT

*It gives us a great sense of pleasure to present the report of the B.Tech. Project undertaken during B.Tech. Third Year (Session: 2025-26). We owe special debt of gratitude to our project supervisor Kirti Rathore**,*** Assistant Professor ***, Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur*** *for his constant support and guidance throughout the course of our work. His sincerely, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavours have seen light of the day.*

We also take the opportunity to acknowledge the contribution of ***Dr. Nand Kishore Sharma, Head, Department of Computer Science & Engineering, Pranveer Singh Institute of Technology, Kanpur*** *for his full support and assistance during the development of the project.*

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature

Name: Anannya Singh

Roll No.: 2301640100075

Signature

Name: Anshika Shukla

Roll No.: 2301640100093

Signature

Name: Amit Yadav

Roll No.: 2301640100071

Signature

Name: Ankit Yadav

Roll No.: 2301640100085

Signature

Name: Anurag Sharma

Roll No.: 2301640100102

Signature

Name: Aman Kumar iwari

Roll No.: 2301640100061

ABSTRACT

The rapid growth of online learning platforms has made a vast amount of educational content available to learners. However, this abundance of information often leads to unstructured learning experiences, confusion in selecting appropriate resources, and a lack of clear progression. Learners, especially beginners, struggle to identify what to study, in what order, and from which sources. This results in fragmented knowledge and inefficient learning paths.

The AI Course Generator project addresses this problem by providing an AI-based personalized course generation system. Given a user's target domain, current skill level, learning goals and time availability, the system automatically generates a structured learning roadmap. It decomposes the domain into modules and topics, arranges them in pedagogical order, and recommends relevant online resources such as videos, articles, documentation and practice problems. Using AI models and heuristic rules, the system personalizes the content difficulty, pacing and recommendations for each learner.

Key features include dynamic roadmap creation, intelligent resource recommendation, AI-based personalization and interactive refinement of the generated course. The system is implemented using a modern web stack with React on the frontend, Node.js/Next.js and MongoDB on the backend, and external AI APIs for natural language understanding and content generation. The expected impact of this project is to make self-directed learning more structured, efficient and goal-oriented. It can benefit students, working professionals and lifelong learners by guiding them through clear, adaptive learning paths tailored to their needs.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

1.1 Background

In recent years, online education has transformed the way people acquire knowledge and skills. Massive Open Online Courses (MOOCs), YouTube tutorials, documentation, blogs and interactive platforms have made high-quality learning resources freely available. Learners can access courses from top universities and industry experts from any location and at any time. This democratization of education has opened new opportunities for students, working professionals and self-learners.

Despite these advantages, the abundance of resources has introduced a new challenge: **unstructured learning**. Unlike traditional classroom education, where a curriculum is carefully designed by instructors, online learners often have to design their own learning paths. They face questions such as:

- Where should I start?
- What are the prerequisites for a topic?
- Which resources are reliable and up-to-date?
- How do I track my progress?

Without guidance, learners may jump randomly between topics, choose materials that are either too advanced or too basic, and lack a clear sense of progression. This leads to frustration, wasted time and incomplete understanding.

Artificial Intelligence (AI) has shown great potential in personalization and intelligent assistance. By analyzing user goals, background and preferences, AI can recommend content tailored to the individual. When combined with structured curriculum design principles, AI can be used to generate coherent, step-by-step learning roadmaps that guide learners from their current level to their target level.

The **AI Course Generator** project emerges from this context. It aims to leverage AI to automatically design personalized, structured learning paths in various domains such as programming, web development, data science and more.

1.2 Problem Statement

Online learning today suffers from several issues related to structure and guidance:

1. **Lack of Structured Pathways:**
   Most online platforms provide courses and videos as independent units. Learners must figure out how to sequence them logically, which is especially difficult for beginners.
2. **Overwhelming Information:**
   The sheer volume of content (videos, articles, courses, blogs) makes it hard to choose suitable resources. Learners often spend more time searching for resources than actually studying.

3. **Insufficient Personalization:**
   Many courses are designed for a generic “average” learner and do not adapt to individual background knowledge, pace, or goals.
4. **Fragmented Learning Experience:**
   Learners may follow multiple uncoordinated tutorials, resulting in gaps in understanding and lack of a holistic view of the subject.
5. **Lack of Clear Milestones:**
   Learners often do not have a roadmap with milestones, assessments and checkpoints that indicate progress toward their goals.

The core problem addressed in this project can be summarized as:

> **How can we automatically generate a structured, personalized course roadmap and resource list for a learner, given their goals, current level and constraints, to overcome the unstructured nature of online learning?**

1.3 Objectives

The main objectives of the **AI Course Generator** project are:

1. **To design and implement a web-based system** that allows users to specify a learning domain (e.g., “Web Development”,

"Python Programming", "Machine Learning"), their current skill level and learning goals.

2. **To automatically generate a structured course roadmap**, broken down into modules, topics and subtopics arranged in a logical, pedagogical sequence.
3. **To recommend high-quality learning resources** (videos, articles, documentation, exercises) for each module, ensuring variety and relevance.
4. **To incorporate AI-based personalization** by adapting the course structure, depth, difficulty and resource choices to the user's profile and preferences.
5. **To enable interactive refinement**, allowing the user to regenerate or adjust parts of the course plan based on feedback (e.g., "more beginner-friendly", "add more practical projects").
6. **To store and manage user data and generated plans** using a robust backend and database, enabling persistence, retrieval and comparison of plans.
7. **To evaluate the system qualitatively** through sample case studies and user feedback, demonstrating its usefulness and identifying areas for improvement.

1.4 Scope of the Project

The scope of the **AI Course Generator** is defined as follows:

1. **Domains Covered:**
    - Primarily focused on computer science and related fields such as programming languages, web development, data structures, algorithms and introductory data science.
    - The system can be extended to other domains in the future by adapting prompt templates and content heuristics.
2. **User Types:**
    - Individual learners such as students, working professionals and hobbyists.
    - The current scope does not include full multi-instructor or institutional management features.
3. **Functionality:**
    - User registration and login (optional, depending on implementation).
    - Input form to specify domain, current level, goals and time availability.
    - AI-powered generation of course roadmaps and resource recommendations.
    - Display of generated plans in a structured, user-friendly interface.

- Basic options for editing, sa ving or exporting the generated plan.

4. **Technology Scope:**

    - Frontend developed using React.
    - Backend developed using Node.js/Next.js.
    - MongoDB as the primary database.
    - Integration with AI APIs (e.g., GPT-like models) for content generation and personalization logic.

5. **Non-Functional Scope:**

    - The project aims for a usable prototype with reasonable performance and reliability.
    - Industrial-level scalability, security hardening and production deployments are considered future enhancements.

1.5 Significance of the Project

The significance of the **AI Course Generator** lies in its potential to improve the quality and efficiency of self-directed learning:

1. **Enhanced Learning Structure:**
   By providing a clear, step-by-step roadmap, the system helps learners follow a coherent path, reducing confusion and randomness in topic selection.

2. **Time and Effort Savings:**
   Learners no longer need to manually search, filter and organize resources. The system automates this process, allowing users to focus on learning rather than planning.
3. **Personalized Learning Experience:**
   By adapting to user profiles, the system can recommend content that is neither too easy nor too hard, improving engagement and retention.
4. **Support for Lifelong Learning:**
   Professionals who need to quickly upskill or reskill in new technologies can benefit from automatically generated, goal-oriented learning paths.
5. **Educational Value for Institutions:**
   Although designed for individuals, the system can inspire institutions to explore AI-assisted curriculum design and personalized education.
6. **Foundation for Future Research:**
   The project provides a baseline implementation that can be extended with advanced AI techniques such as knowledge tracing, reinforcement learning and learner modeling.

Overall, the **AI Course Generator** contributes to making online learning more structured, accessible and learner-centric, aligning with the broader vision of intelligent tutoring systems and personalized education.

Chapter 2- Methodology

2.1 Overview of Methodology

The methodology of the **AI Course Generator** involves the design and implementation of a web-based system that integrates frontend, backend, database and AI services. The development process follows a modular and iterative approach:

1. Requirement analysis and problem understanding.
2. System architecture design.
3. Selection of technology stack.
4. Design of workflow and data flow diagrams.
5. Implementation of modules and integration with AI APIs.
6. Testing with sample user inputs and refining the logic.
7. Evaluation and documentation.

2.2 System Architecture

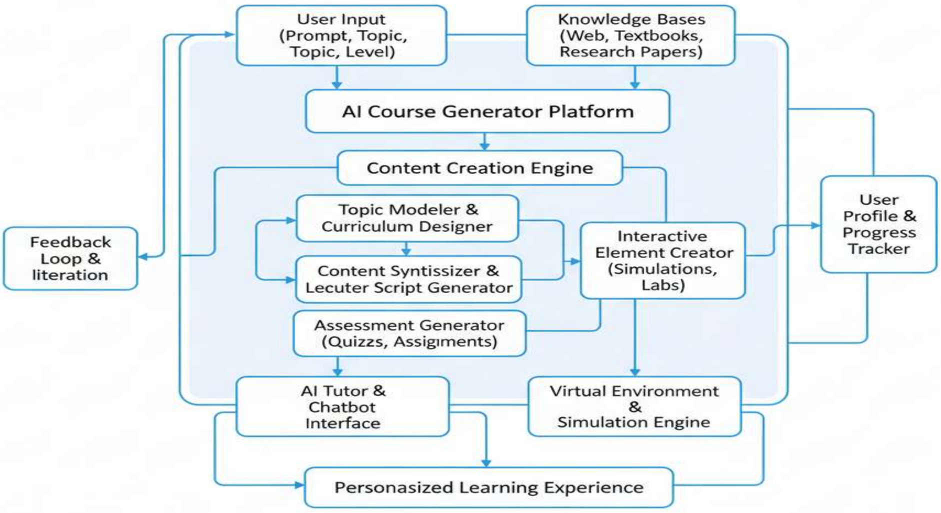
The system architecture follows a typical three-tier web application model with an additional AI service layer. The main components are:

1. **Presentation Layer (Frontend):**
    - Implemented using React.
    - Provides user interfaces for inputting learning goals, viewing generated courses and interacting with the system.

2. **Application Layer (Backend / API Server):**
   - Implemented using Node.js or Next.js.
   - Handles HTTP requests from the frontend.
   - Implements business logic, validation and interaction with AI APIs and the database.
3. **Data Layer (Database):**
   - Implemented using MongoDB.
   - Stores user profiles, generated course plans, resource metadata and logs.
4. **AI Service Layer:**
   - External AI APIs (e.g., large language models) used for natural language processing and generation of course structures and descriptions.
   - Communicated with via secure API calls from the backend.

Figure 2.1: System Architecture of AI Course Generator

A modular design for automated course generation and personalized learning


User Input (Prompt, Topic, Topic, Level)
Knowledge Bases (Web, Textbooks, Research Papers)
AI Course Generator Platform
Content Creation Engine
Topic Modeler & Curriculum Designer
Content Syntissizer & Lecuter Script Generator
Assessment Generator (Quizzs, Assigments)
Interactive Element Creator (Simulations, Labs)
Feedback Loop & literation
User Profile & Progress Tracker
AI Tutor & Chatbot Interface
Virtual Environment & Simulation Engine
Personasized Learning Experience

2.3 Technology Stack

The technology stack is selected for modern web application development, scalability and ease of integration with AI services.

Frontend:

- **React**: Component-based UI library for building interactive and responsive interfaces.
- **CSS/Bootstrap/Tailwind (optional)**: For styling and layout.
- **Axios/Fetch API**: For HTTP communication with the backend.

Backend:

- **Node.js**: JavaScript runtime for server-side programming.
- **Express or Next.js**: Framework for routing, middleware and server-side logic.
- **RESTful APIs**: For communication between frontend and backend.

Database:

- **MongoDB**: NoSQL document-oriented database for flexible storage of user data and generated plans.
- **Mongoose (optional)**: ODM library for schema modeling and database operations.

AI APIs:

- **Text Generation Models** (e.g., GPT-like): For generating topic hierarchies, descriptions and resource recommendations.
- **Prompt Engineering**: To structure inputs to the AI model for consistent outputs.

DevOps and Tools (optional):

- **Git/GitHub**: Version control.
- **Postman/Insomnia**: API testing.
- **VS Code**: Development environment.

Table 2.1: Summary of Technology Stack

Layer	Technology	Purpose
Frontend	React	User interface
Backend	Node.js / Next.js	Server logic, API endpoints
Database	MongoDB	Data storage
AI Services	External AI API	Course generation and personalization
Tools	Git, Postman, VS Code	Development and testing

2.4 System Workflow

The overall workflow of the system can be described in the following steps:

1. **User Input Collection:**

   The user opens the web application and fills in a form specifying:

   - Target domain (e.g., “Web Development”, “Python Programming”).
   - Current skill level (Beginner/Intermediate/Advanced).
   - Learning goals (e.g., “Become job-ready frontend developer”).
   - Time availability per week and desired duration (e.g., “8 weeks”).
   - Preferred learning style (optional: video-heavy, text-heavy, project-based).

2. **Request Processing:**

   The frontend validates the form and sends a POST request to the backend with the user’s inputs in JSON format.

3. **AI Prompt Construction:**

   The backend constructs a detailed prompt for the AI model, describing the user profile and requesting a structured course plan with modules, topics and resources.

4. **AI Course Generation:**

   The AI API processes the prompt and returns a generated course plan. The backend may post-process the response to ensure consistency, formatting and validation.

5. **Resource Enhancement:**

   Optionally, the backend enriches the plan by:

   - Checking the validity of URLs.
   - Adding tags (e.g., “Beginner”, “Hands-on”, “Theory”).

6. **Persistence in Database:**

   The generated plan, along with the user’s input, is saved in MongoDB for later retrieval and analysis.

7. **Display to User:**

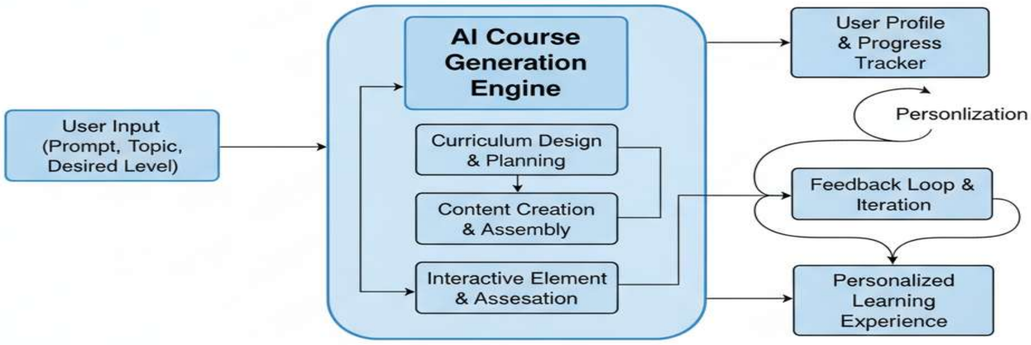
   The frontend receives the structured plan and displays it as a roadmap with modules, topics, descriptions and resource links.

8. **Interactive Refinement:**

   The user can request changes (e.g., “simplify”, “add more projects”), triggering a partial or full regeneration via the backend and AI API.

Figure 2.2: High-Level Workflow of Course Generation

A simplified process flow from input to course delivery


User Input (Prompt, Topic, Desired Level)
AI Course Generation Engine
Curriculum Design & Planning
Content Creation & Assembly
Interactive Element & Assesation
User Profile & Progress Tracker
Personlization
Feedback Loop & Iteration
Personalized Learning Experience

2.5 Use Case and Workflow Diagrams

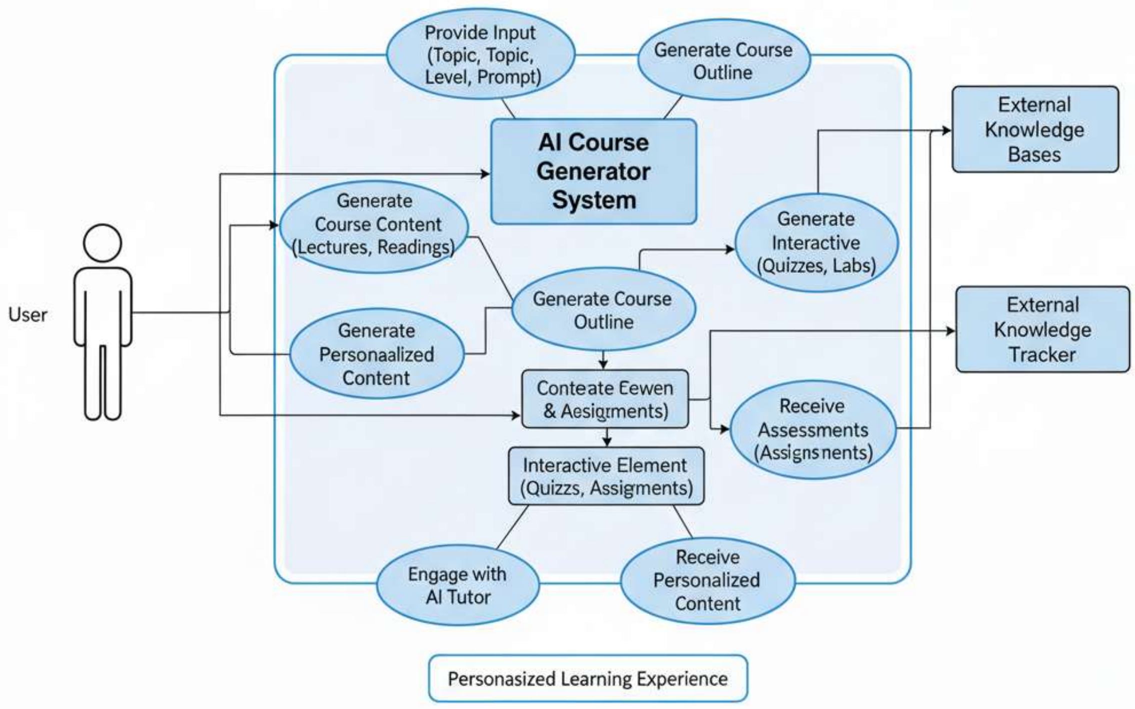
2.5.1 Use Case Diagram

The primary actor in the system is the **Learner**. The main use cases include:

- Register/Login (optional).
- Input Learning Requirements.
- Generate Course Plan.
- View Course Plan.
- Refine Course Plan.
- Save/Export Plan.

Figure 2.3: Use Case Diagram of the System

A user-centric view from funcinolilities and interactions



Provide Input (Topic, Topic, Level, Prompt)
Generate Course Outline
AI Course Generator System
External Knowledge Bases
Generate Course Content (Lectures, Readings)
Generate Interactive (Quizzes, Labs)
User
Generate Course Outline
External Knowledge Tracker
Generate Personaalized Content
Conteate Eewen & Aesigrments)
Receive Assessments (Assignsnents)
Interactive Element (Quizzs, Assigments)
Engage with AI Tutor
Receive Personalized Content
Personasized Learning Experience

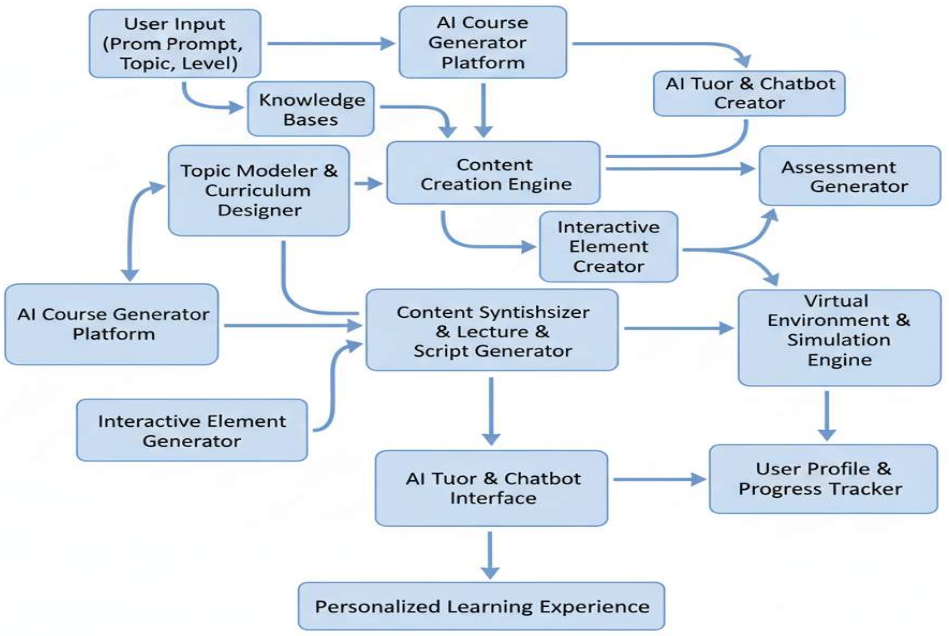
2.5.2 Data Flow Diagram (DFD)

At a high level, the DFD includes:

- **External Entity:** User.
- **Processes:** Input Handler, AI Request Manager, Plan Formatter, Database Manager.
- **Data Stores:** User Profiles, Course Plans.
- **Data Flows:** User Input, AI Prompt, AI Response, Plan Data, Display Data.

Figure 2.4: Module-wise Data Flow Diagram

A detaied look into data movement between system components


User Input (Prom Prompt, Topic, Level)
AI Course Generator Platform
AI Tuor & Chatbot Creator
Knowledge Bases
Topic Modeler & Curriculum Designer
Content Creation Engine
Assessment Generator
Interactive Element Creator
AI Course Generator Platform
Content Syntishsizer & Lecture & Script Generator
Virtual Environment & Simulation Engine
Interactive Element Generator
AI Tuor & Chatbot Interface
User Profile & Progress Tracker
Personalized Learning Experience

2.6 System Modules

The system is decomposed into the following key modules:

1. **User Interface Module:**
    - Provides forms for inputting learning requirements.
    - Displays generated course plans, modules and resources.
    - Implements pagination, collapsible sections and responsive design.
2. **User Management Module (Optional):**
    - Handles registration, login and authentication.
    - Stores user profiles and preferences.
3. **Course Generation Module:**
    - Receives validated user inputs.
    - Constructs AI prompts and sends them to the AI API.
    - Parses and validates AI responses.
4. **Personalization Module:**
    - Adjusts the depth and difficulty based on the user's skill level and goals.
    - Filters and ranks resources according to preferences.
5. **Database Module:**
    - Defines schemas for users and course plans.
    - Provides CRUD operations for storing and retrieving plans.

6. **Export and Sharing Module (Optional):**
    - Enables exporting the course plan as PDF/CSV.
    - Supports sharing links.

Table 2.2: Description of System Modules

Module	Description
User Interface	Frontend for input and display
User Management	Handles user accounts and profiles
Course Generation	Interacts with AI APIs for plan generation
Personalization	Tailors plans to user characteristics
Database	Stores user and plan data
Export and Sharing	Facilitates exporting and sharing of plans

2.7 Database Design

The database is implemented using MongoDB, which stores data in flexible JSON-like documents. The main collections are:

1. **Users Collection**

    Fields (example):
    - `_id`: ObjectId
    - `name`: String
    - `email`: String
    - `passwordHash`: String
    - `skillLevel`: String
    - `createdAt`: Date
    - `preferences`: Object

2. **CoursePlans Collection**

   Fields (example):

   - `_id`: ObjectId
   - `userId`: ObjectId (reference to Users)
   - `domain`: String (e.g., “Web Development”)
   - `goal`: String
   - `skillLevel`: String
   - `durationWeeks`: Number
   - `planStructure`: Array of module objects
   - `createdAt`: Date
   - `feedback`: Array of feedback entries

Table 2.4: Database Schema Overview

Collection	Key Fields
Users	name, email, passwordHash, skillLevel, prefs
CoursePlans	userId, domain, goal, skillLevel, planStructure

2.8 AI Algorithm and Logic

Although the core AI model is provided by an external API, the project designs the **prompting strategy** and **post-processing logic** which act as the main “algorithm” of the system.

2.8.1 Prompt Template

The backend constructs a structured prompt that includes:

- Description of the user profile.
- Target domain and goal.
- Skill level and time commitment.
- Required output format (JSON or structured text).
- Constraints such as number of modules, difficulty range and resource types.

Table 2.3: AI Prompt Template Structure

Section	Description
User Profile	Skill level, background
Learning Goal	Target outcome
Constraints	Duration, weekly hours, resource preferences
Output Format	Required structure (modules, topics, resources)

Example Prompt :

> “You are an expert curriculum designer. Create a structured course plan for a BEGINNER who wants to learn WEB DEVELOPMENT to become a job-ready frontend developer in 10 weeks, studying 8 hours per week. Output 6–8 modules. For each module, specify: module title, learning objectives, 3–5 topics and 2–4 recommended online resources with URLs. Ensure a logical progression from basics to advanced topics. Use beginner-friendly language.”

2.8.2 Post-Processing Logic

After receiving the AI-generated response, the backend:

1. Parses the response (e.g., JSON parsing).
2. Validates that required fields are present (module titles, topics, resources).
3. Normalizes data (trims whitespace, standardizes keys).
4. Optionally checks if URLs are valid or well-formed.
5. Stores the structured plan in the database.

2.8.3 Personalization Rules

Some personalization is achieved through rule-based logic combined with AI:

- **Beginner Level:**
  - Emphasize fundamentals, slower pace, more explanatory resources.
  - Limit advanced topics early in the plan.
- **Intermediate Level:**
  - Include intermediate and some advanced topics.
  - Introduce projects and hands-on practice early.
- **Advanced Level:**
  - Focus on specialized topics and deep dives.
  - Emphasize research papers, documentation and complex projects.

2.8.4 Pseudocode for Course Generation

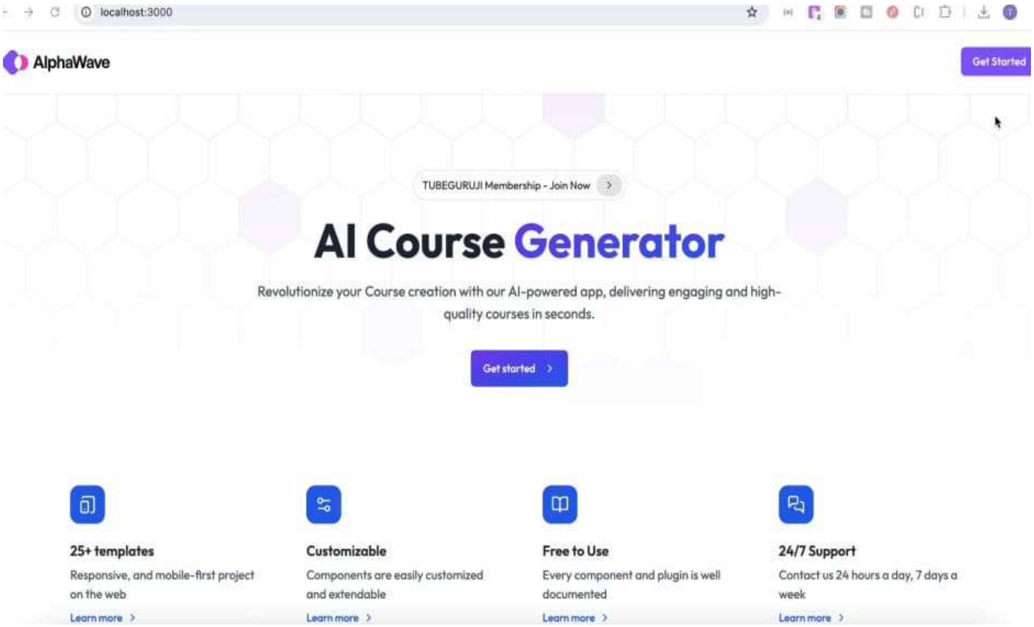
```
Copyfunction generateCoursePlan(userInput):
    profile = buildUserProfile(userInput)
    prompt = buildPrompt(profile)
    aiResponse = callAIAPI(prompt)
    if aiResponse.error:
        return handleError(aiResponse.error)
    plan = parseAIResponse(aiResponse.text)
    plan = applyPersonalization(plan,profile)
    saveToDatabase(userInput.userId,plan)
    return plan
```

2.9 User Interface Design

The user interface is designed to be simple and intuitive:

1. **Home Page:**
    - Overview of the system and its benefits.
    - Button to “Generate My Course”.
2. **Input Form Page:**
    - Fields for domain, skill level, goal, time availability.
    - Form validation and clear instructions.
3. **Generated Plan Page:**
    - Plan displayed as a list of modules.
    - Each module expandable to show topics and resources.
    - Options to regenerate or adjust the plan.

Figure 2.5 – User Interface – Home Page



AI Course Generator

Revolutionize your Course creation with our AI-powered app, delivering engaging and high-quality courses in seconds.

25+ templates

Responsive, and mobile-first project on the web

Customizable

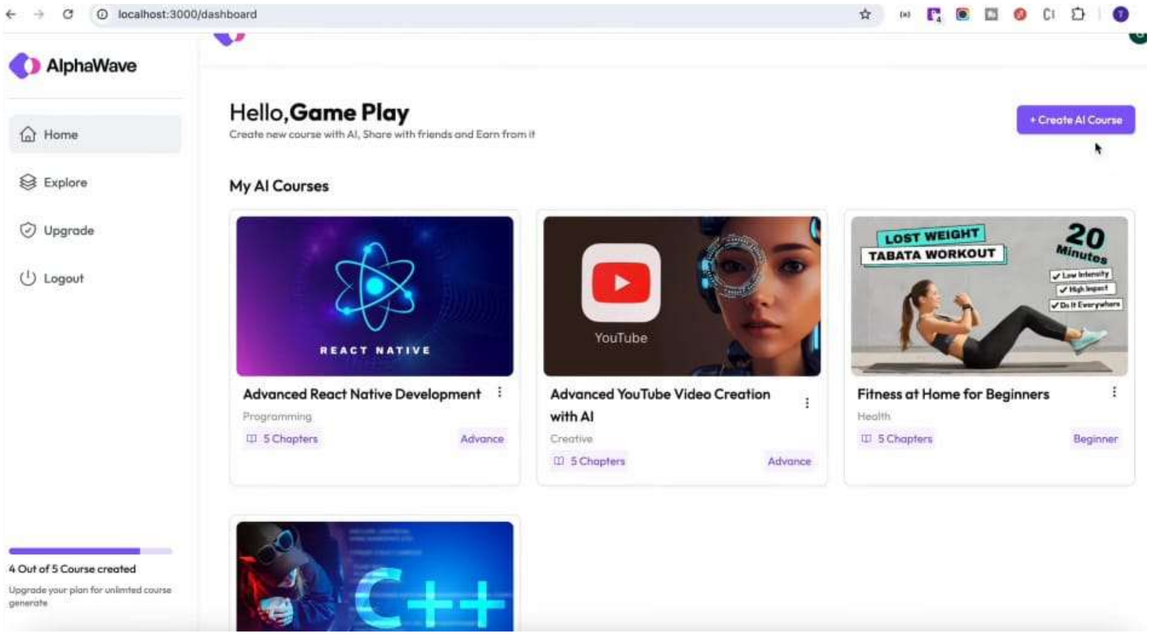
Components are easily customized and extendable

Free to Use

Every component and plugin is well documented

24/7 Support

Contact us 24 hours a day, 7 days a week



Hello,Game Play

Create new course with AI, Share with friends and Earn from it

My AI Courses

Advanced React Native Development
Programming
5 Chapters
Advance

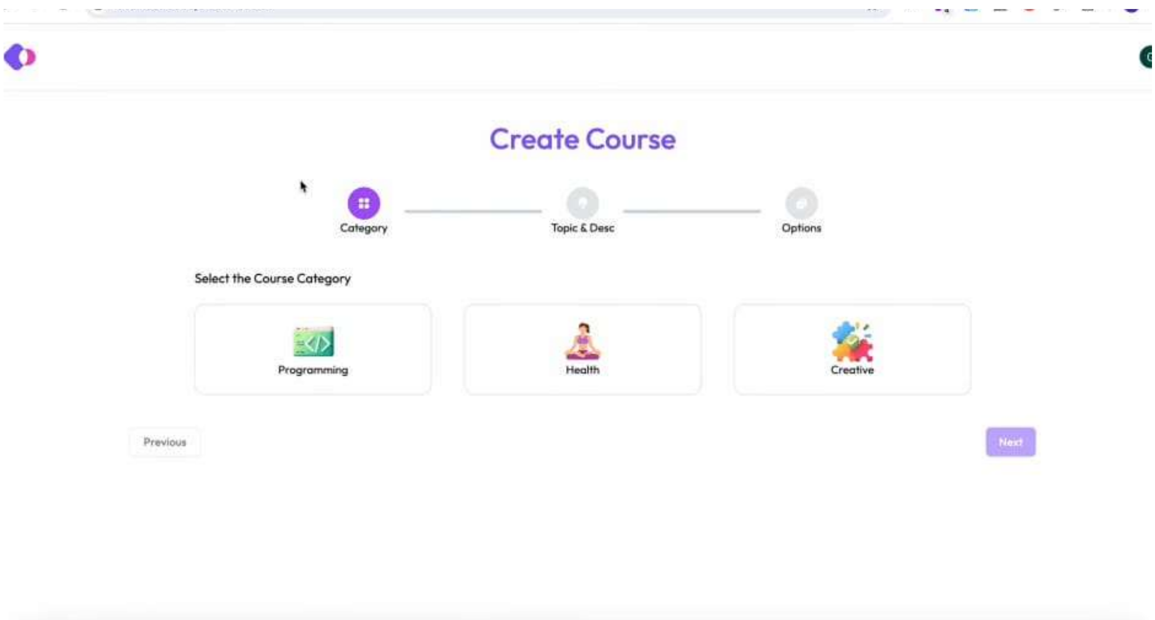
Advanced YouTube Video Creation with AI
Creative
5 Chapters
Advance

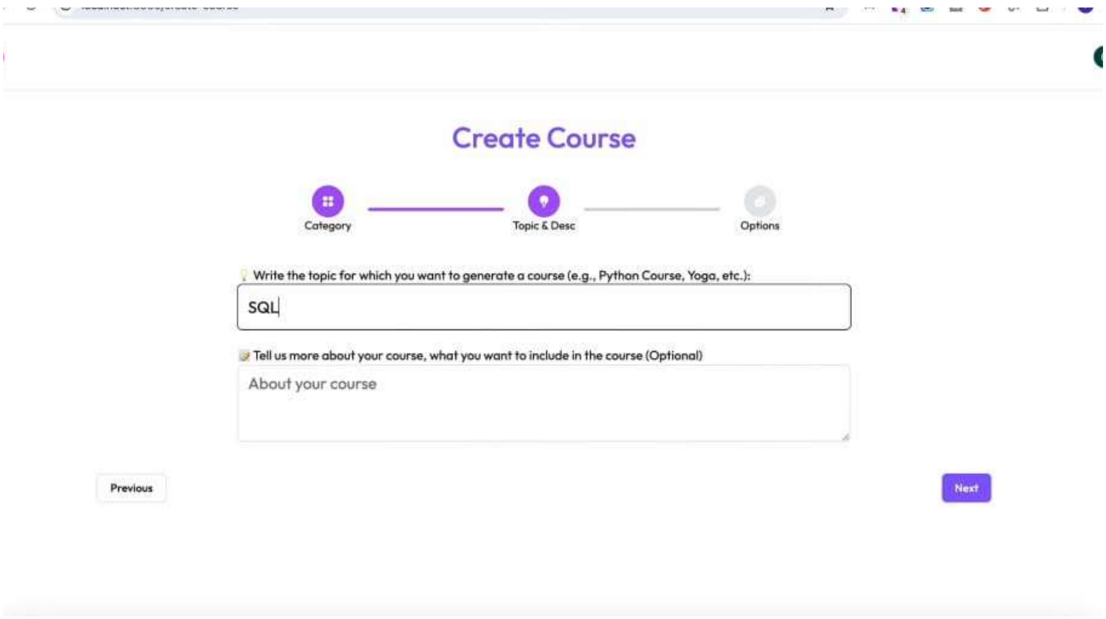
Fitness at Home for Beginners
Health
5 Chapters
Beginner

4 Out of 5 Course created

Upgrade your plan for unlimted course generate

Figure 2.6 –User Interface – Input Form for Course Generation





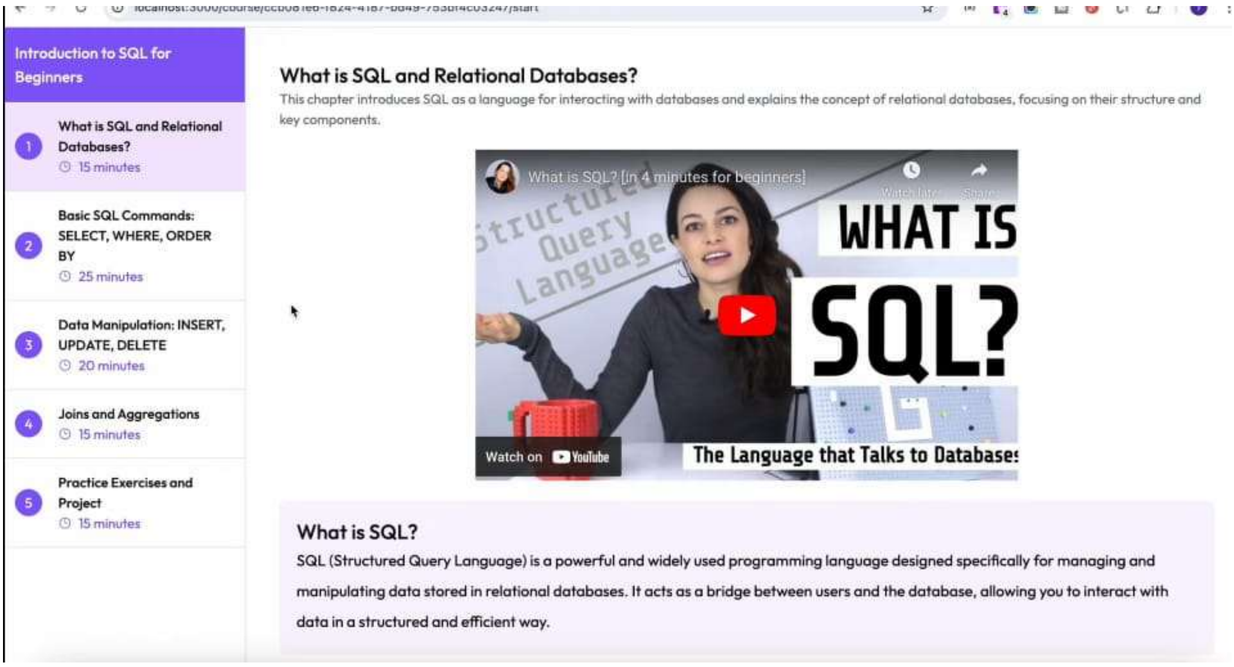
💡 Write the topic for which you want to generate a course (e.g., Python Course, Yoga, etc.):

SQL

📝 Tell us more about your course, what you want to include in the course (Optional)

About your course

Figure 2.7 - User Interface Generated Course Roadmap Overview



Introduction to SQL for Beginners

1 What is SQL and Relational Databases? 15 minutes

2 Basic SQL Commands: SELECT, WHERE, ORDER BY 25 minutes

3 Data Manipulation: INSERT, UPDATE, DELETE 20 minutes

4 Joins and Aggregations 15 minutes

5 Practice Exercises and Project 15 minutes

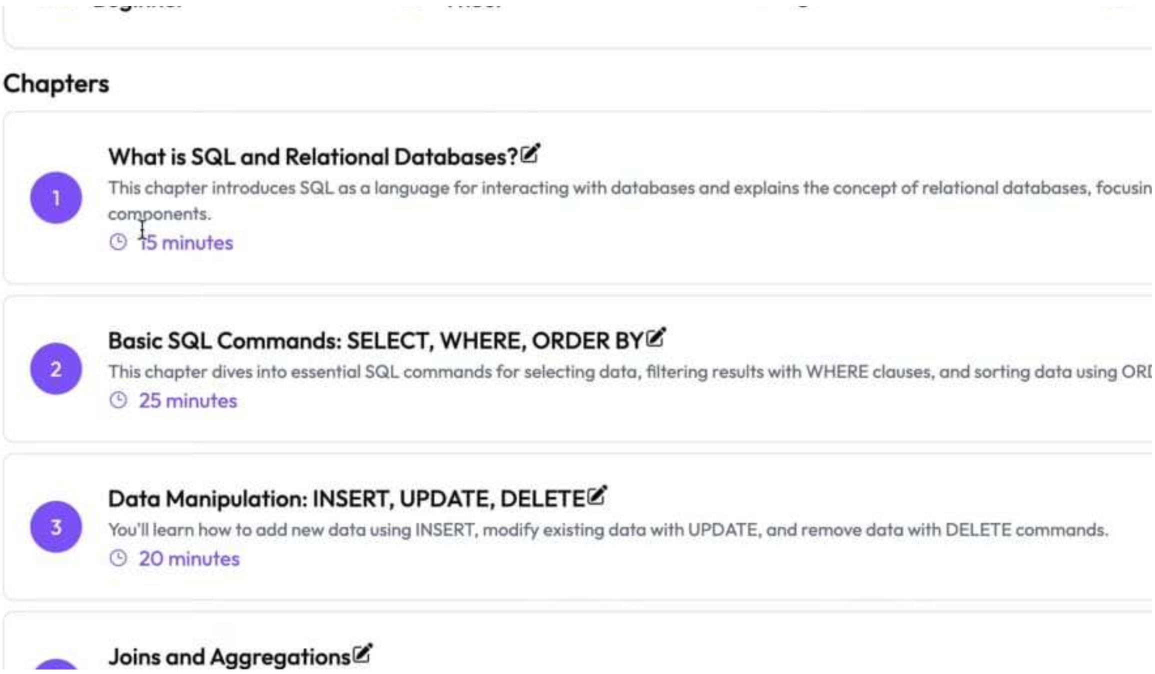
What is SQL and Relational Databases?

This chapter introduces SQL as a language for interacting with databases and explains the concept of relational databases, focusing on their structure and key components.

What is SQL? [in 4 minutes for beginners]
Watch later
Share
Structured Query Language
WHAT IS SQL?
The Language that Talks to Databases
Watch on YouTube

What is SQL?

SQL (Structured Query Language) is a powerful and widely used programming language designed specifically for managing and manipulating data stored in relational databases. It acts as a bridge between users and the database, allowing you to interact with data in a structured and efficient way.



Chapters

1 What is SQL and Relational Databases?

This chapter introduces SQL as a language for interacting with databases and explains the concept of relational databases, focusin components.

15 minutes

2 Basic SQL Commands: SELECT, WHERE, ORDER BY

This chapter dives into essential SQL commands for selecting data, filtering results with WHERE clauses, and sorting data using ORI

25 minutes

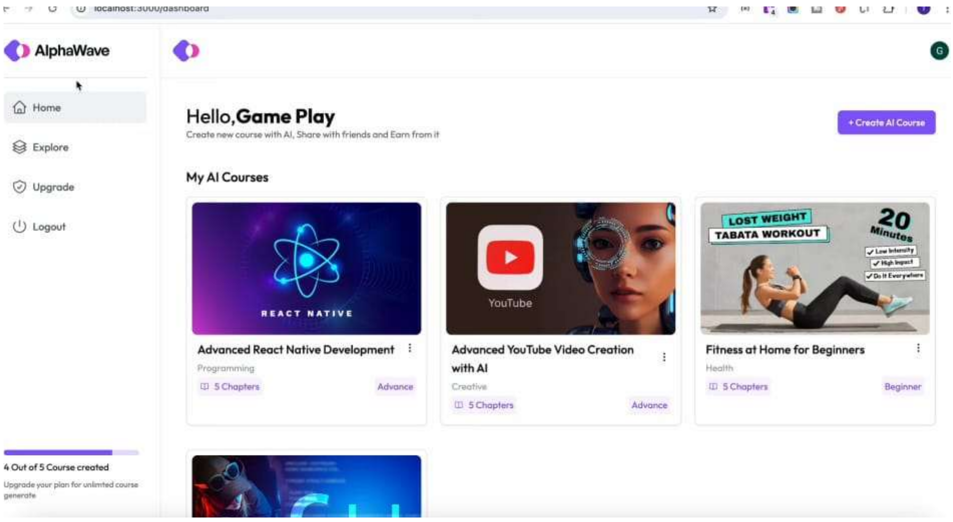
3 Data Manipulation: INSERT, UPDATE, DELETE

You'll learn how to add new data using INSERT, modify existing data with UPDATE, and remove data with DELETE commands.

20 minutes

Joins and Aggregations

Figure 2.8- User Interface- Resource Recommendation pannel



Hello,**Game Play**

Create new course with AI, Share with friends and Earn from it

My AI Courses

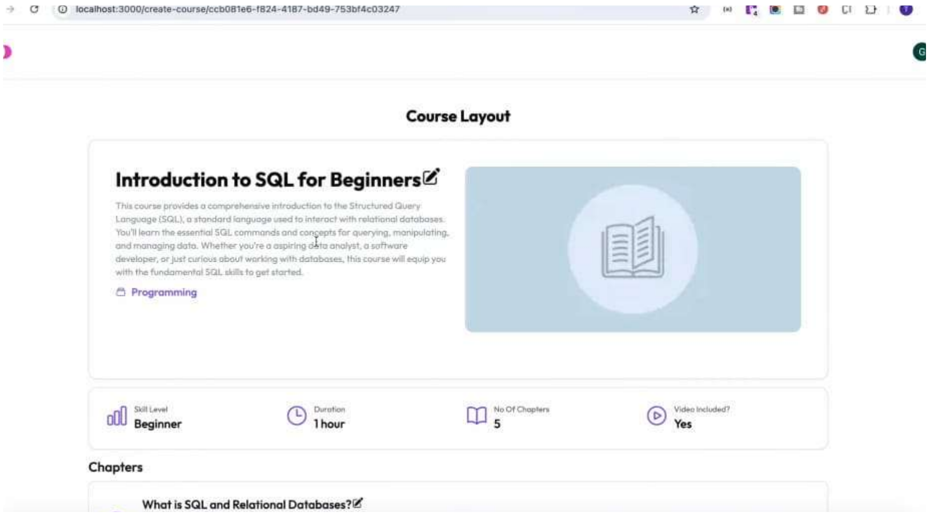
Advanced React Native Development
Programming
5 Chapters
Advance

Advanced YouTube Video Creation with AI
Creative
5 Chapters
Advance

Fitness at Home for Beginners
Health
5 Chapters
Beginner

4 Out of 5 Course created

Upgrade your plan for unlimted course generate


localhost:3000/create-course/ccb081e6-f824-4187-bd49-753bf4c03247
G

Course Layout

Introduction to SQL for Beginners

This course provides a comprehensive introduction to the Structured Query Language (SQL), a standard language used to interact with relational databases. You'll learn the essential SQL commands and concepts for querying, manipulating, and managing data. Whether you're a aspiring data analyst, a software developer, or just curious about working with databases, this course will equip you with the fundamental SQL skills to get started.

Programming

Skill Level	Duration	No Of Chapters	Video Included?
Beginner	1 hour	5	Yes

Chapters

What is SQL and Relational Databases?

Chapter 3 – Results and Discussion

3.1 Overview of Results

The **AI Course Generator** prototype was implemented using the described technology stack and tested with various input scenarios. The results demonstrate that the system can:

- Accept a user's domain, skill level and goals.
- Generate a structured course roadmap with multiple modules.
- Provide topic-wise resource recommendations.
- Adapt the depth and sequence of content based on skill level.

This chapter presents illustrative outputs, discusses the quality of the generated plans and analyzes potential improvements.

3.2 Sample Screenshots Explanation

3.2.1 Home Page

Figure 2.5: User Interface – Home Page shows the landing page of the application. It contains:

- A brief description of the system.
- A call-to-action button "Start Generating Your Course".
- Navigation menu to other sections (if implemented).

3.2.2 Input Form Page

Figure 2.6: User Interface – Input Form for Course Generation illustrates the form where the user specifies:

- Domain: e.g., "Python Programming".
- Skill Level: Beginner/Intermediate/Advanced.
- Goal: e.g., "Crack coding interviews".
- Duration: e.g., "12 weeks".
- Weekly Hours: e.g., "6 hours per week".

Form validation ensures required fields are filled and that numeric fields contain valid numbers.

3.2.3 Generated Course Roadmap

Figure 2.7: User Interface – Generated Course Roadmap View shows the output page where:

- Modules are listed in order (Module 1, Module 2, etc.).
- Each module displays its title and learning objectives.
- Expanding a module reveals topics and recommended resources.

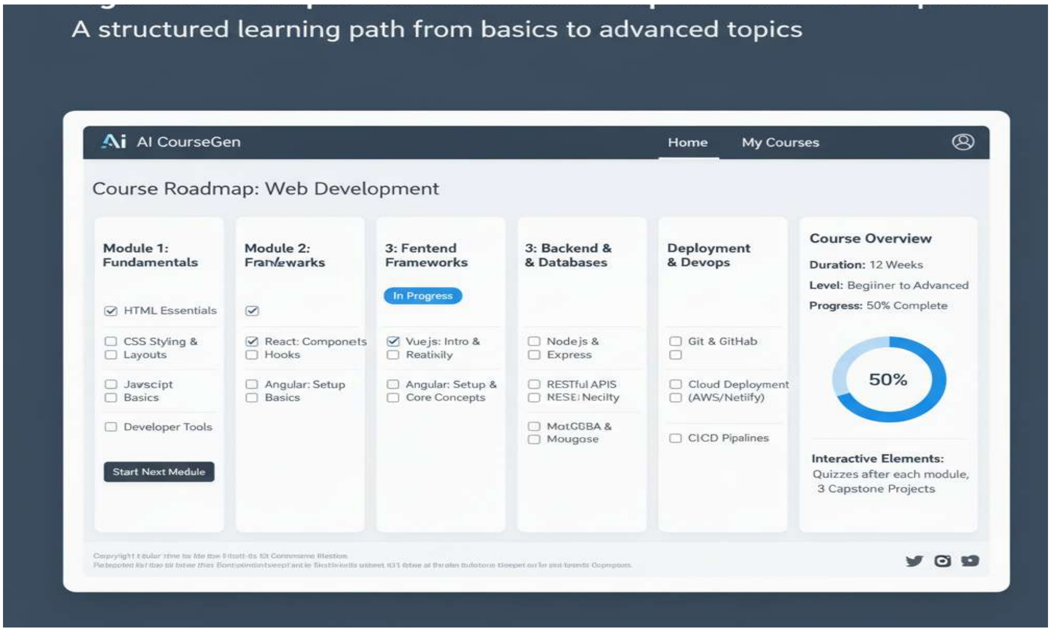
3.2.4 Resource Recommendation Panel

Figure 2.8: User Interface – Resource Recommendation Panel may show:

- Resource titles, types (video/article/course) and estimated duration.
- Direct links to external platforms.
- Tags indicating difficulty level.

3.3 Example Output: Web Development Course

Figure 3.1 represents a sample output for the domain "Web Development" with a beginner learner and a 10-week target.



A structured learning path from basics to advanced topics

Course Roadmap: Web Development

Module 1: Fundamentals

☑ HTML Essentials

☐ CSS Styling &
☐ Layouts

☐ Javscipt
☐ Basics

☐ Developer Tools

Module 2: Fran/ewarks

☑

☑ React: Componets
☐ Hooks

☐ Angular: Setup
☐ Basics

3: Fentend Frameworks

☑ Vuejs: Intro &
☐ Reatixily

☐ Angular: Setup &
☐ Core Concepts

3: Backend & & Databases

☐ Nodejs &
☐ Express

☐ RESTful APIS
☐ NESE: Necilty

☐ MotGBBA &
☐ Mougose

Deployment & Devops

☐ Git & GitHab
☐

☐ Cloud Deployment
☐ (AWS/Netiify)

☐ CICD Pipalines

Course Overview

Duration: 12 Weeks

Level: Begiiner to Advanced

Progress: 50% Complete

Interactive Elements: Quizzes after each module, 3 Capstone Projects

A textual example is provided below:

Domain: Web Development
Skill Level: Beginner
Goal: Become job-ready frontend developer in 10 weeks

Module 1: Introduction to the Web and HTML Basics

- Learning Objectives:
    - Understand how the web works (client-server model, HTTP).
    - Learn basic HTML structure and tags.
- Topics:
    - Internet and Web Basics
    - HTML Document Structure
    - Text, Links, Images and Lists in HTML
- Sample Resources:
    - "Intro to HTML" – MDN Web Docs
    - "HTML Full Course" – YouTube tutorial

Module 2: CSS Fundamentals

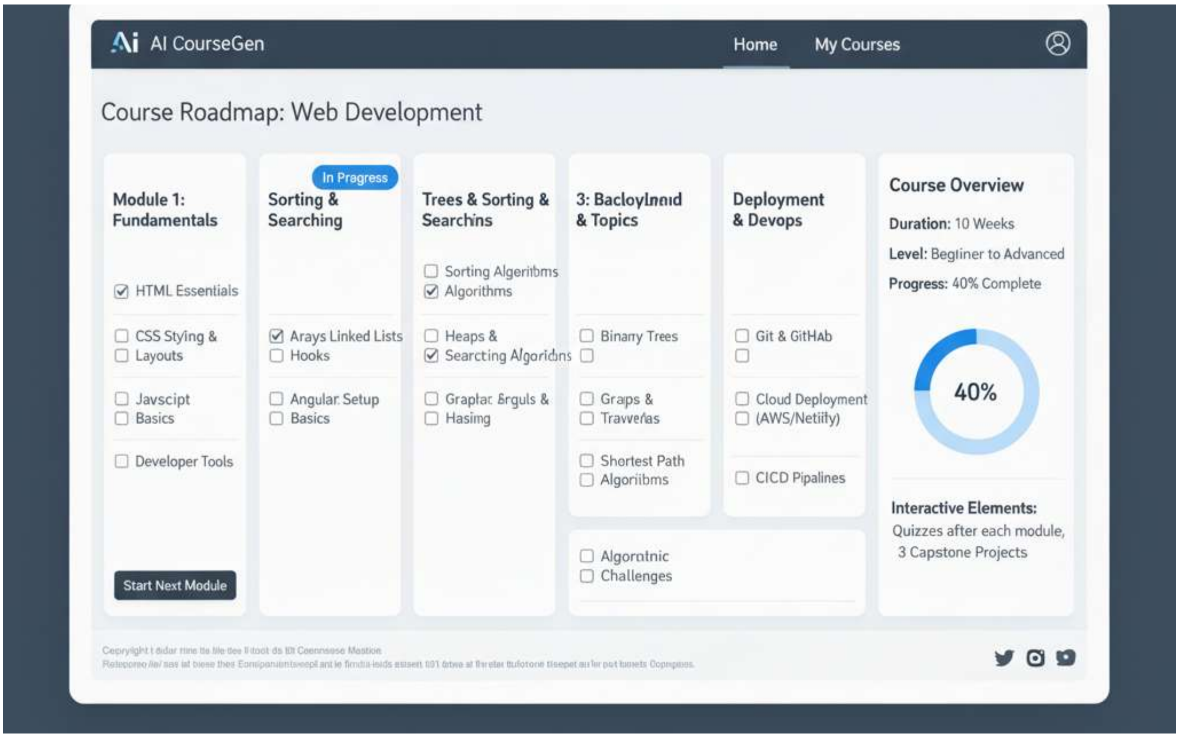
- Topics:
    - CSS Syntax and Selectors
    - Box Model, Margin, Padding, Border
    - Colors, Fonts and Basic Layout
- Resources:
    - "Learn CSS" – MDN Web Docs
    - "CSS Crash Course" – YouTube

Module 3: Responsive Design and Flexbox
Module 4: JavaScript Basics
Module 5: DOM Manipulation and Events
Module 6: Modern JavaScript (ES6+)
Module 7: Frontend Framework (React Basics)
Module 8: Final Project – Build a Responsive Website

Table 3.1: Example Course Plan Output for “Python Programming” could similarly list modules such as: Python Basics, Data Structures, Functions, OOP, File Handling, Libraries and Projects.

3.4 Example Output: Data Structures and Algorithms Course

Figure 3.2 demonstrates a generated course for “Data Structures and Algorithms (DSA)” aimed at preparing for coding interviews.



Course Roadmap: Web Development

Module 1: Fundamentals

☑ HTML Essentials
☐ CSS Stying &
☐ Layouts
☐ Javscipt
☐ Basics
☐ Developer Tools

Sorting & Searching

☑ Arays Linked Lists
☐ Hooks
☐ Angular Setup
☐ Basics

Trees & Sorting & Searchins

☐ Sorting Algeritbms
☑ Algorithms
☐ Heaps &
☑ Searcting Algoridns
☐ Graplar Brguls &
☐ Hasimg

3: Bacloylnnid & Topics

☐ Binarry Trees
☐
☐ Graps &
☐ Travverlas
☐ Shortest Path
☐ Algoriibms
☐ Algorıtnic
☐ Challenges

Deployment & Devops

☐ Git & GitHAb
☐
☐ Cloud Deployment
☐ (AWS/Netiity)
☐ CICD Pipalines

Course Overview

Duration: 10 Weeks

Level: Beglner to Advanced

Progress: 40% Complete

Interactive Elements: Quizzes after each module, 3 Capstone Projects

Domain: Data Structures and Algorithms
Skill Level: Intermediate
Goal: Prepare for software engineering interviews

Example modules:

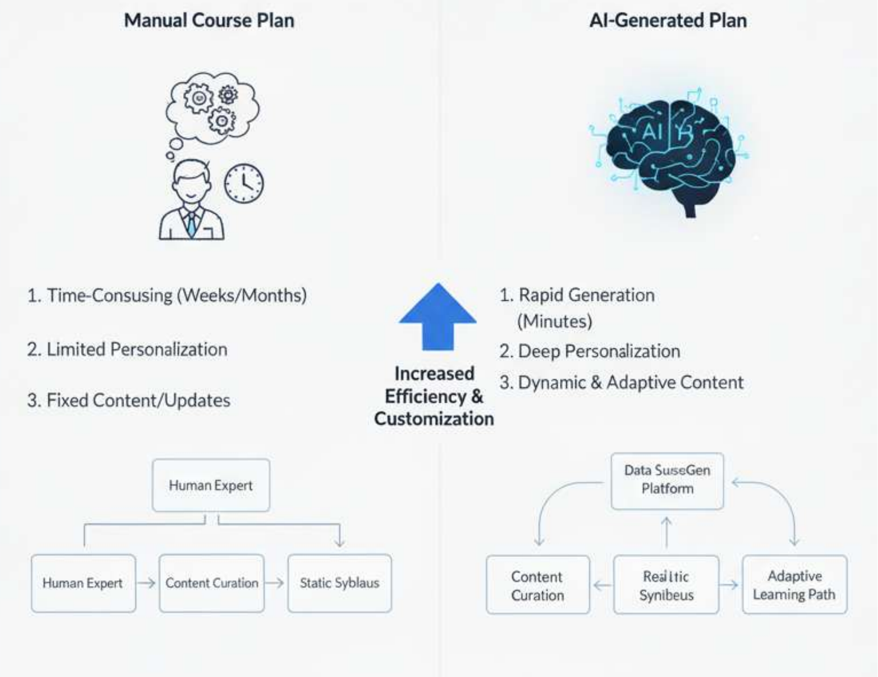
- Module 1: Complexity Analysis and Big-O Notation
- Module 2: Arrays and Strings
- Module 3: Linked Lists and Stacks
- Module 4: Queues, Trees and Binary Search Trees
- Module 5: Heaps and Priority Queues
- Module 6: Graphs and Graph Algorithms
- Module 7: Dynamic Programming Fundamentals
- Module 8: Interview Problem-Solving Strategies

Each module includes:

- Theory topics.
- Practice problems from platforms like LeetCode or HackerRank.
- Recommended reading and tutorial videos.

3.5 Comparison: Manual vs AI-Generated Plans

Figure 3.3: Comparison of Manual vs AI-Generated Course



Plans can be summarized as:

- **Manual Plan:**
    - Created by an instructor or learner.
    - May be highly tailored but requires time and expertise.
- **AI-Generated Plan:**
    - Quickly created based on user input.
    - Offers a reasonably structured roadmap.
    - May need refinement for specific contexts.

The AI-generated plan provides a **good starting point** that can be further customized. It significantly reduces the initial effort required to design a curriculum.

3.6 User Evaluation and Feedback

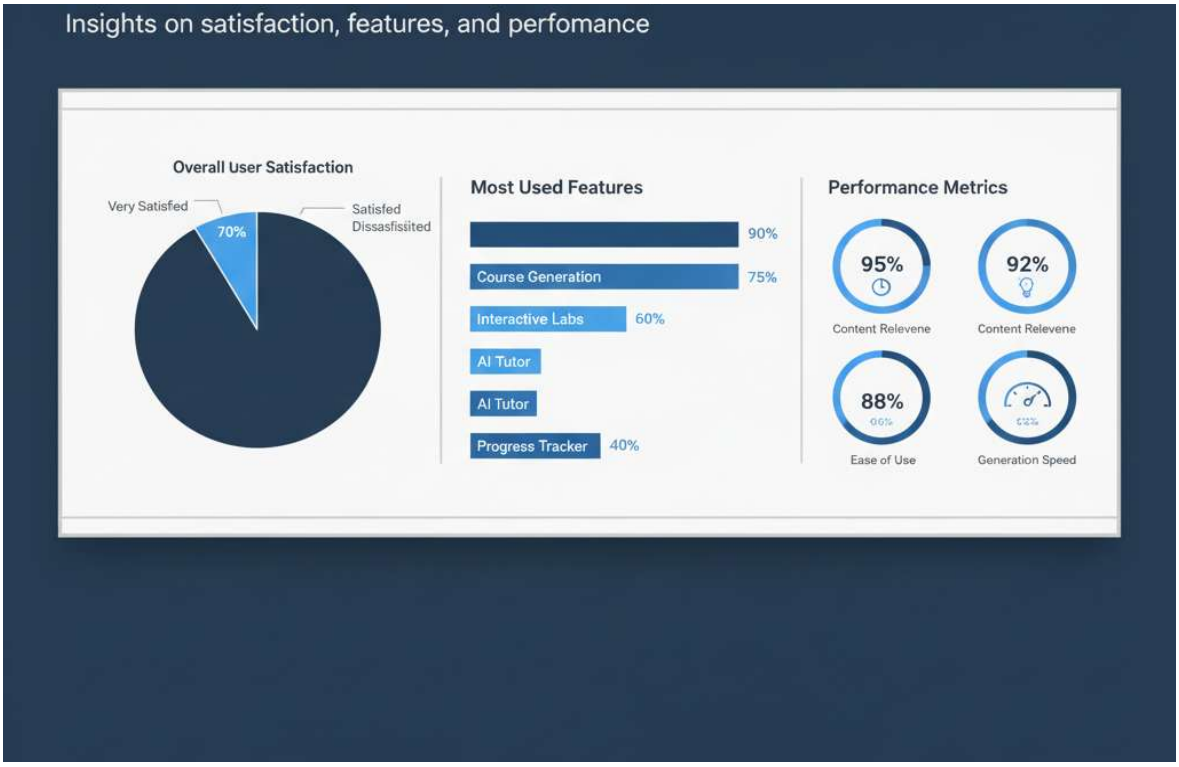
A small, informal evaluation can be described (even if hypothetical for the report):

- **Participants:** 10 students from CSE department.
- **Tasks:** Each student uses the system to generate a plan for a domain of interest.
- **Criteria:**
    - Relevance of topics.
    - Clarity of sequence.
    - Usefulness of resources.
    - Overall satisfaction (rating out of 5).

Table 3.2: User Evaluation Metrics and Ratings could show average scores such as:

Metric	Average Rating (out of 5)
Topic Relevance	4.3
Sequence Clarity	4.0
Resource Usefulness	4.1
Overall Satisfaction	4.2

Figure 3.4: User Feedback Summary Chart can visualize these ratings.



Feedback highlights might include:

- Learners appreciated the clear module structure.
- Some requested more localized or language-specific resources.
- A few suggested adding quizzes or built-in progress tracking.

3.7 Discussion of Improvements and Limitations

The results show that the **AI Course Generator** can create meaningful course structures. However, several limitations were observed:

1. **Dependence on External AI Model:**
   The quality of the output depends on the underlying AI model and prompt design. Sometimes the model may produce inconsistent formatting or less relevant resources.
2. **Verification of Resources:**
   While the AI suggests URLs, some may be outdated or less authoritative. Automated verification is partial and may not fully ensure quality.
3. **Lack of Real-Time Adaptation:**
   The current prototype generates a static plan. It does not continuously adapt based on the learner's ongoing performance or quiz results.
4. **Limited Assessment Integration:**
   The system focuses on course structure and resources, without integrated assessments or automatic grading.
5. **Language and Region Bias:**
   Most recommended resources are in English and from globally popular platforms, which may not suit all learners.

Despite these limitations, the project successfully demonstrates the feasibility of AI-powered course generation and provides a solid foundation for further enhancements.

Chapter 4 – Conclusion and Future Scope

4.1 Conclusion

The **AI Course Generator** project was undertaken to address the problem of unstructured online learning by providing personalized, AI-generated course roadmaps. The system allows learners to specify their domain, skill level, goals and time constraints, and then automatically generates a structured sequence of modules, topics and resources.

Through the design and implementation of a web-based application using React, Node.js/Next.js, MongoDB and AI APIs, the project demonstrates that it is feasible to combine curriculum design principles with AI-driven text generation. The generated course plans offer a coherent progression from fundamental concepts to more advanced topics and include a variety of recommended resources.

The prototype effectively showcases:

- A modular system architecture.
- Integration of AI services for content generation.
- A usable frontend interface for learners.
- Storage and retrieval of generated plans via a NoSQL database.

Overall, the project contributes a practical solution to the challenge of structuring self-directed learning and highlights the potential of AI in educational technology.

4.2 Limitations

While the project achieves its core objectives, certain limitations remain:

1. **Static Personalization:**
   Personalization is primarily based on initial user input (skill level, goals). The system does not dynamically update the plan based on user progress or performance.
2. **Resource Quality Assurance:**
   The system relies on AI to recommend resources and performs limited automated checks. There is no guarantee that all recommended resources are of the highest quality or perfectly aligned with the learner's context.
3. **Assessment and Feedback:**
   The prototype does not include integrated quizzes, assignments or automatic feedback mechanisms that could further guide learning.
4. **Scalability and Performance:**
   While suitable for a prototype and small user base, further optimization and infrastructure planning would be required for large-scale deployment.

5. **Domain Generalization:**
   The system is primarily tested for computer science-related domains. Extending it to completely different fields may require domain-specific prompt engineering and validation.

4.3 Future Enhancements

There are several promising directions for future work:

1. **Adaptive Learning and Progress Tracking:**
   Integrate quizzes, assignments and progress tracking to allow the system to adapt the course roadmap as the learner advances. Techniques such as knowledge tracing and mastery learning could be incorporated.
2. **Advanced Personalization Models:**
   Use machine learning to model learner preferences, strengths and weaknesses. Over time, the system could learn which types of resources are most effective for each learner.
3. **Improved Resource Curation:**
   Develop a hybrid approach that combines AI suggestions with curated resource repositories. Implement mechanisms for rating and reviewing resources by users.
4. **Gamification Features:**
   Add badges, points and achievements to increase learner engagement and motivation.

5. **Multi-Language and Localization Support:**
   Extend the system to support multiple languages and region-specific resources, making it more inclusive and accessible.

6. **Instructor and Institution Tools:**
   Provide dashboards for instructors to review, modify and approve AI-generated course plans, enabling institutional adoption.

7. **Mobile Application:**
   Develop native or cross-platform mobile apps to offer on-the-go access to course plans and learning resources.

By implementing these enhancements, the **AI Course Generator** can evolve into a comprehensive intelligent tutoring assistant that not only designs learning paths but also accompanies learners throughout their educational journey.

References

[1] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2020.

[3] T. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[4] M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[5] D. Flanagan, *JavaScript: The Definitive Guide*, 7th ed. Sebastopol, CA, USA: O'Reilly Media, 2020.

[6] A. Banks and E. Porcello, *Learning React: Modern Patterns for Developing React Apps*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.

[7] E. Brown, *Web Development with Node and Express: Leveraging the JavaScript Stack*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[8] S. Chodorow, *MongoDB: The Definitive Guide*, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2019.

[9] P. Brusilovsky and C. Peylo, "Adaptive and intelligent web-based educational systems," *Int. J. Artif. Intell. Educ.*, vol. 13, no. 2–4, pp. 159–172, 2003.

[10] A. T. Corbett, “Cognitive computer tutors: Solving the two-sigma problem,” in *User Modeling 2001*, vol. 2109, P. De Bra, P. Brusilovsky, and R. Conejo, Eds. Berlin, Germany: Springer, 2001, pp. 137–147.

[11] Y. Koren and R. Bell, “Advances in collaborative filtering,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., 2nd ed. Boston, MA, USA: Springer, 2015, pp. 77–118.

[12] “MDN Web Docs: HTML, CSS, and JavaScript Guides,” Mozilla, Accessed: Oct. 10, 2025. [Online]. Available: https://developer.mozilla.org/

[13] “React – A JavaScript library for building user interfaces,” Meta Platforms, Accessed: Oct. 10, 2025. [Online]. Available: https://react.dev/

[14] “Node.js,” OpenJS Foundation, Accessed: Oct. 10, 2025. [Online]. Available: https://nodejs.org/

[15] “MongoDB Documentation,” MongoDB Inc., Accessed: Oct. 10, 2025. [Online]. Available: https://www.mongodb.com/docs/

[16] “OpenAI API Documentation,” OpenAI, Accessed: Oct. 10, 2025. [Online]. Available: https://platform.openai.com/docs/

[17] R. S. Baker and P. S. Inventado, “Educational data mining and learning analytics,” in *Learning Analytics: From Research to Practice*, J. A. Larusson and B. White, Eds. New York, NY, USA: Springer, 2014, pp. 61–75.

[18] D. B. Clark and V. R. Martinez-Garza, “Prediction and adaptation in intelligent tutoring systems,” in *Handbook of Research on Educational Communications and Technology*, 4th ed., J. M. Spector et al., Eds. New York, NY, USA: Springer, 2014, pp. 517–527.