Academic Planner — Project Documentation

Project Overview

The **Academic Planner** is a web application that helps students manage their courses, plan their schedules, and track academic progress effectively. It offers a smooth, responsive experience across devices with smart study suggestions and integrated AI chat support.

Features:

- ✓ **User Authentication** (Login & Signup with validation)
- ✓ **Course Management** (Add, update, delete, and track courses)
- ✓ **Progress Tracker** (Visual progress percentage)
- ✓ **Calendar & Timetable** (Plan upcoming study tasks/events)
- ✓ **Suggestion System** (Smart productivity tips)
- ✓ **AI Chat Support** (Assist users with academic queries)
- ✓ **Fully Responsive Design** (Mobile, tablet, desktop compatibility)
- ✓ **Error Handling** (Proper messages for invalid inputs)
- ✓ **API Testing Done** (For all major functionalities)

Technology Stack:

- ✓ **Frontend**: React.js, Tailwind CSS
- ✓ **Backend/API**: Node.js/Express.js (or Django, based on your backend)
- ✓ **Database**: MongoDB (or your configured database)
- ✓ **Authentication**: JWT (or relevant tokens)
- ✓ **Testing Tools**: Postman, Jest (optional for frontend testing)

Installation and setup

Prerequisites

Node.js and npm installed

MongoDB database connection (or your backend server)

Clone the Repository

git clone https://github.com/your-username/academic-planner.git

cd academic-planner

Install Dependencies

npm install

Set up Node.js Backend (backend):

cd backend

npm install

node index.js

Set up Python Server (pythonserver):

cd pythonserver

python -m venv venv

source venv/bin/activate # (On Windows: venv\Scripts\activate)

pip install -r requirements.txt

python app.py

Project Structure

```
Academic_Planner
|
├── academic_planner        # Frontend (React + Vite + TailwindCSS)
|   ├── public
|   ├── src
|   |   ├── assets
|   |   ├── components
|   |   ├── pages
|   |   └── main.jsx
|   └── index.html
|
|
├── backend                 # Backend (Node.js + Express + JWT)
|   ├── config
|   ├── controllers
|   ├── middleware
|   ├── model
|   ├── routes
|   └── index.js
|
└── pythonserver            # Python Server (Microservices / APIs)
    ├── app.py
    └── .env
```

API Documentation:

All API calls including login, signup, course management, and AI suggestions have been tested successfully. The login API returns the token and user data for valid inputs, and shows appropriate error messages for invalid submissions. Signup API correctly handles new user creation and duplicate email scenarios. Course APIs support adding, updating, deleting, and fetching courses with proper validations. AI chat and suggestion APIs also respond accurately for both valid and invalid inputs. Overall, all backend services are functioning smoothly and consistently.

Testing Report:

Testing involved unit tests for components, forms, and APIs, API testing through Postman, and manual functional testing across various devices. Login, signup, course operations, calendar events, responsiveness, and AI chat interactions were thoroughly verified. Tools like Postman and manual browser/device testing were used. All functionalities performed as expected, APIs handled both valid and invalid inputs properly, and the UI maintained full responsiveness across mobile, tablet, and desktop views.
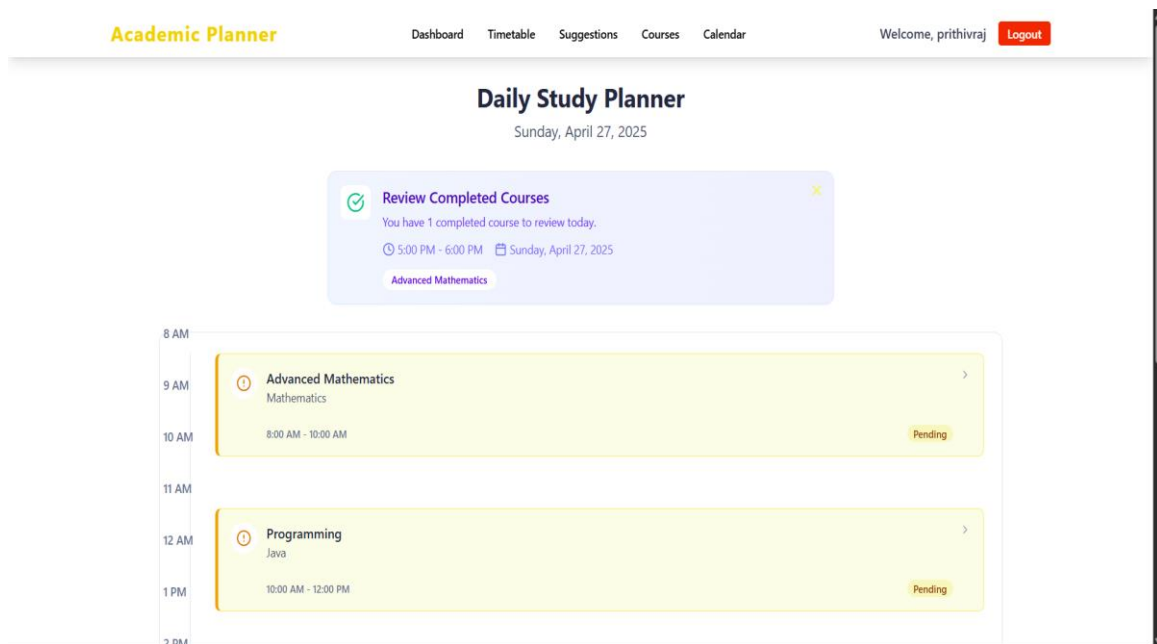
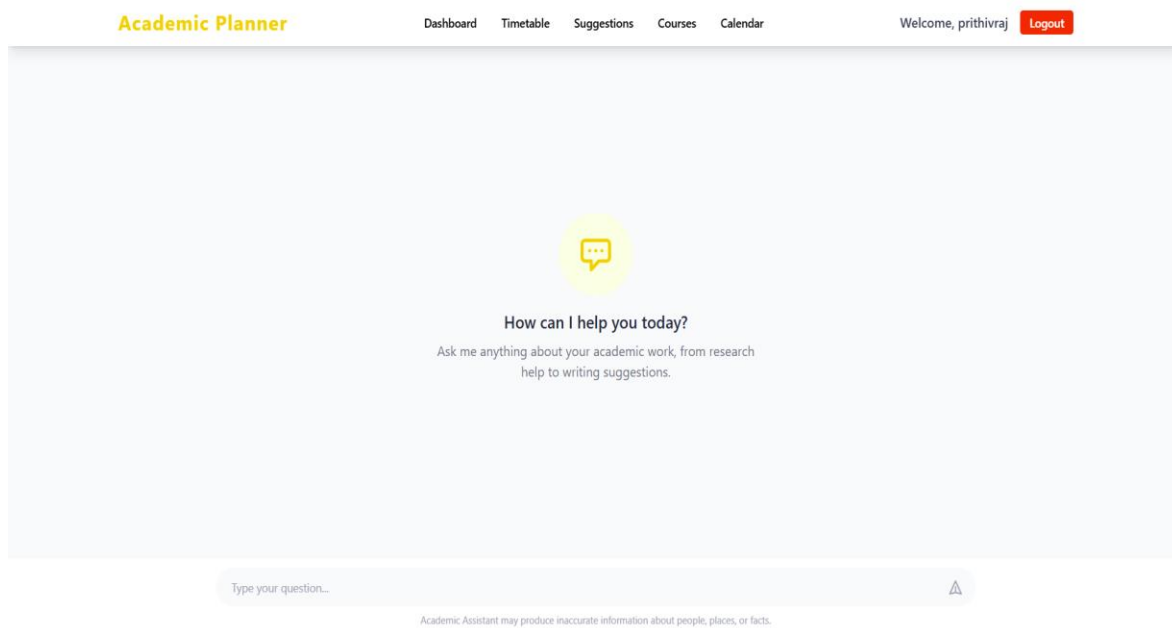## Output:
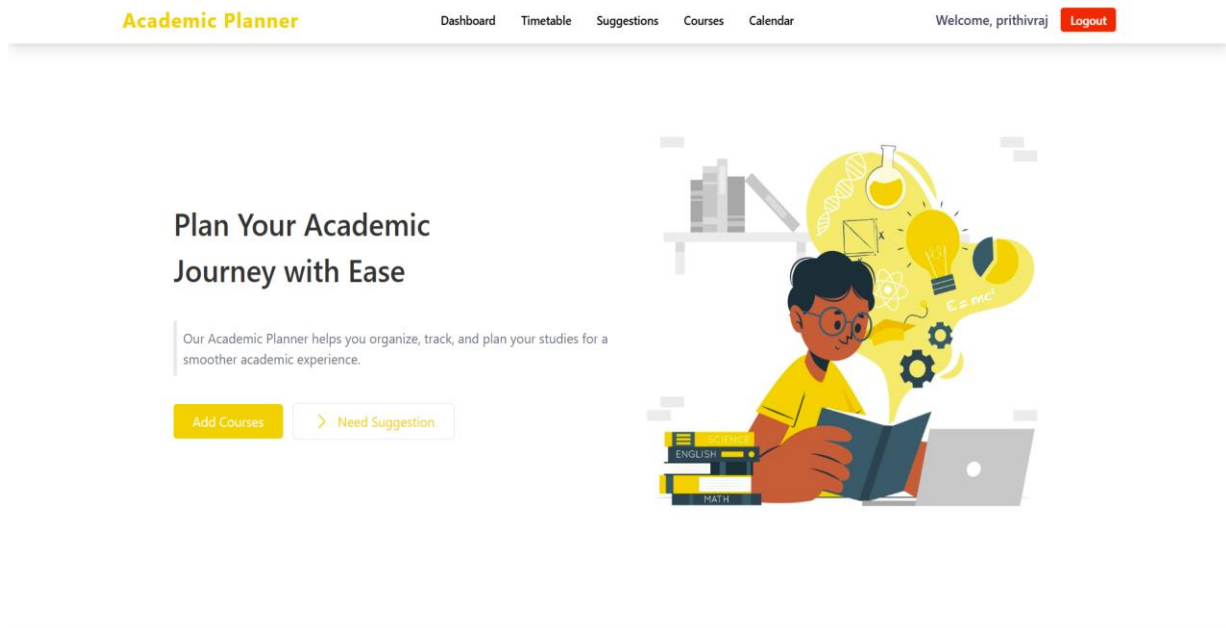
FIG.NO:1 – HOME PAGE

FIG.NO:2 – TIME TABLE
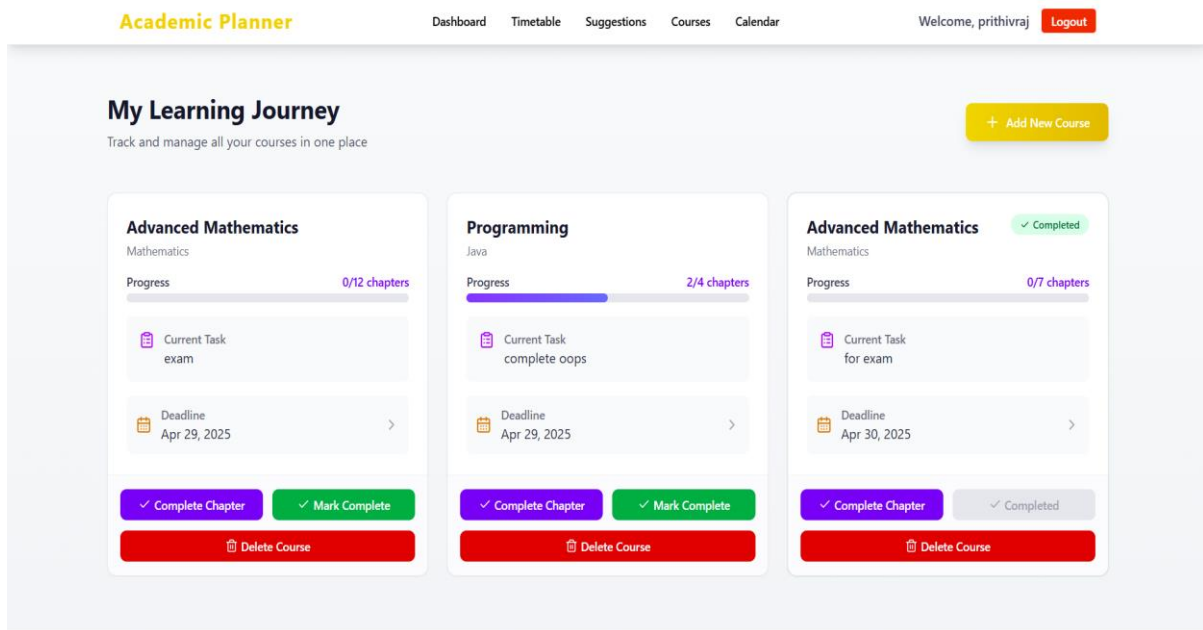

FIG.NO:3 – SUGGESTION PAGE
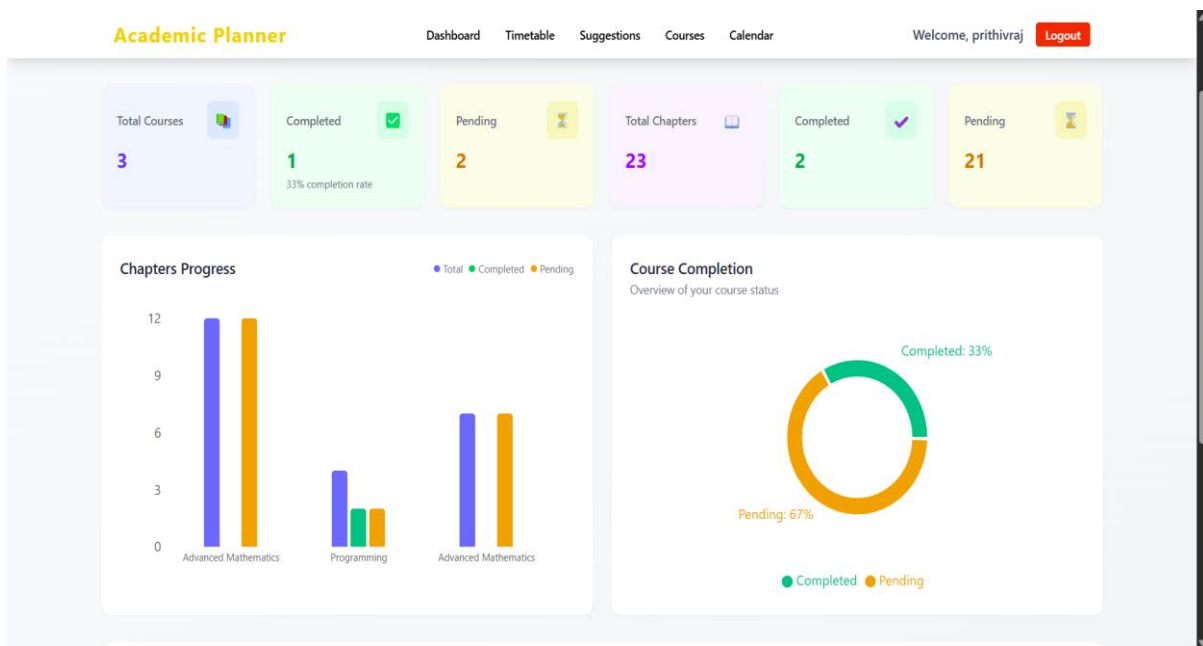
FIG.NO:4 – COURSE PAGE

FIG.NO:5 – DASHBOARD

Conclusion:

Academic Planner with AI offers a smart, intuitive, and efficient way for students to manage their academic life — powered by modern web technologies and cutting-edge AI!