# 🚀 AI Study Planner + Productivity Tracker

**(MERN + Real AI Logic, not just chatbot)**

---

# 🎯 Core Problem

**Students:**

- Over-plan karte hain (unrealistic timetable)
- Ya bilkul plan nahi karte
- Consistency maintain nahi hoti
- Burnout ho jata hai

**Tumhara system solve karega:**

> **"Perfect plan nahi — sustainable plan."**

---

# 🧠 System Ka Core Flow

## Step 1: User Onboarding

**User enter kare:**

- Exam date
- Subjects
- Topics list
- Current level (Beginner / Intermediate / Advanced)
- Daily available hours
- Energy peak time (Morning / Night)
- Distraction level

**Yeh data MongoDB me store hoga.**

---

# 🤖 AI Personalized Planning Logic

**AI generate karega:**

- Daily subject distribution
- Revision cycles (spaced repetition)
- Break allocation
- Weekly rest day

**Example:**

Exam in 60 days
Math weak
Physics medium
**AI output:**

- Daily 2 hr Math
- Alternate day Physics
- Sunday revision only

---

# 📊 Adaptive Rescheduling (Most Important Feature)

**Yeh project ka REAL AI part yahan hai.**

**Agar:**

- User plan follow nahi karta
- Ya sirf 60% complete karta
- Ya 3 din skip karta

**AI automatically:**

- Schedule compress kare
- Weak subject ko priority de
- Realistic adjustment kare

Ye static timetable app nahi hona chahiye.

---

# 😵 Burnout Detection

**AI track kare:**

- Consecutive study hours
- Low completion %
- Late night sessions
- Mood input

**If burnout risk ↑**
→ Suggest lighter schedule
→ Add motivational prompt
→ Insert recovery day

Yeh feature tumhe differentiate karega.

---

# 🍅 Pomodoro AI Coach

**Normal timer nahi.**

**AI observe kare:**

- Kitne pomodoro sessions complete
- Kis subject me focus drop hota
- Kis time productivity best

**Phir suggest kare:**

- “You focus better at 8pm”
- “After 3 sessions, break increase needed”

---

# 📈 Focus Score Prediction

**Daily:**

- Completion %
- Pomodoro success rate
- Mood
- Distraction reports

**AI generate kare:**

Focus Score (0–100)

And graph show kare.

---

# 🗂️ MERN Architecture

## Frontend (React)

- Dashboard
- Calendar View
- Pomodoro timer
- Analytics charts
- Mood input modal

## Backend (Node + Express)

- Auth
- Plan generation API
- Progress tracking API
- AI logic layer

## Database (MongoDB)

**Collections:**

Users
Goals
DailyPlans
StudySessions
MoodLogs
Analytics

---

# 🤖 AI Implementation Options

### Option 1 (Basic):

**Use OpenAI / Gemini API for:**

- Plan generation
- Suggestions

### Option 2 (Advanced Hybrid):

- Rule-based engine for schedule math
- AI only for adjustment + suggestions

**Better approach = Hybrid.**

---

# 👥 Team Division

### You:

- React dashboard
- Timer logic
- Charts (Recharts)
- UX design

### Friend:

- Schedule algorithm
- AI prompts
- Burnout logic

- DB schema

---

# 🔥 Advanced Version 2 (If You Want Crazy Level)

**Add:**

- AI Voice Study Coach
- Exam prediction (based on consistency)
- Study group leaderboard
- Streak + gamification

# Website Looks As:

Ab hum **UI/Feature Architecture level** pe baat karte hain — kaun-kaun se sections banane chahiye taaki project professional lage, sirf student app nahi.

**Main tumhe 3 layers me breakdown karunga:**

1. 🌎 Public Pages (Without Login)
2. 🔐 Authenticated App Sections (Core System)
3. 🧠 Advanced AI Modules

---

# 🌎 1️⃣ Public Pages (Landing Side)

Ye resume + product presentation ke liye important hai.

## 🏠 Home Page

- Hero section (AI Study Planner intro)
- How it works (3 steps)
- Key features (AI scheduling, burnout detection, focus score)
- Demo screenshots
- CTA: “Start Planning Free”

---

## 📖 About Page

- Problem statement (students overplanning/burnout)
- Solution explanation
- AI ka role

- Vision (long-term roadmap)

---

## ✨ Features Page

**Detail breakdown:**

- Personalized schedule
- Adaptive rescheduling
- Burnout detection
- Pomodoro AI coach
- Focus score prediction

---

## 💰 Pricing Page (Even if fake for now)

- Free tier
- Pro tier (advanced analytics)
- Future roadmap

Product mindset dikhega.

---

## 📞 Contact / Support

Simple form + FAQ.

---

# 🔐 2️⃣ After Login (Main Dashboard App)

Yaha se real project start hota hai.

---

# 🏠 Dashboard

**Show:**

- Today’s plan
- Progress %
- Focus score
- Upcoming deadlines
- Motivational AI message

---

# 🎯 Goals Section

**User:**

- New exam add kare
- Subject add kare
- Topic breakdown kare
- Available hours set kare

Yaha se AI scheduling trigger hoga.

---

# 📅 Study Planner (Calendar View)

- Daily tasks
- Drag & drop reschedule
- Completion checkbox
- AI auto-adjust button

---

# 🍅 Pomodoro Room

- Timer
- Subject select
- Session count
- Break suggestion
- Focus tracking

---

# 📊 Analytics Section

**Charts:**

- Weekly completion rate
- Focus trend
- Subject strength graph
- Burnout risk indicator

---

# 😵 Mood & Burnout Tracker

**User daily mood select kare:**

- 😄 Good
- 😐 Neutral
- 😞 Low energy

AI detect kare pattern.

---

# ⚙️ Settings

- Available study hours update
- Energy peak time
- Notification preferences
- Delete account

---

# 🧠 3️⃣ Advanced AI Sections (Optional but Powerful)

---

## 🧠 AI Suggestions Center

**AI weekly report de:**

- “Math weak ho raha hai”
- “Late night study affecting focus”
- “Add 1 rest day”

---

## 📈 Focus Score Breakdown Page

**Explain:**
**Score kaise calculate hua**

- Completion %
- Pomodoro consistency
- Mood
- Sleep timing (optional future)

---

## 🔁 Adaptive Reschedule Panel

**If user skip kare:**

- Auto adjust button
- Compare old vs new plan

---

# 🏗️ Final Structure Summary

**Public Side:**

- Home
- About
- Features
- Pricing
- Contact

**App Side:**

- Dashboard
- Goals
- Planner
- Pomodoro
- Analytics
- Mood Tracker
- AI Suggestions
- Settings

# Overview of Project:

# Project Folder Structure:

```
ai-study-planner/
│
├── client/              # Frontend (React)
├── server/              # Backend (Node + Express)
├── ai-engine/           # AI Planning + Analytics Logic
├── docs/                # Project Documentation
├── package.json         # Root scripts (concurrently run)
└── README.md
```

---

# Frontend Folder Structure:

**Tech: React + Vite + Tailwind CSS**

```
client/
│
├── public/
│
├── src/
│   ├── assets/
│   ├── components/
│   │   ├── common/
│   │   │   ├── Button.jsx
│   │   │   ├── Input.jsx
│   │   │   ├── Modal.jsx
│   │   │   └── Loader.jsx
│   │   │
│   │   ├── dashboard/
│   │   ├── planner/
│   │   ├── analytics/
│   │   └── ai/
│   │
│   ├── pages/
│   │   ├── public/
│   │   │   ├── Home.jsx
│   │   │   ├── Features.jsx
│   │   │   ├── Pricing.jsx
│   │   │   └── About.jsx
│   │   │
│   │   ├── auth/
│   │   │   ├── Login.jsx
│   │   │   ├── Register.jsx
│   │   │   └── ForgotPassword.jsx
│   │   │
│   │   ├── app/
│   │   │   ├── Dashboard.jsx
│   │   │   ├── StudyPlanner.jsx
│   │   │   ├── TaskManager.jsx
│   │   │   ├── Analytics.jsx
│   │   │   ├── AIInsights.jsx
│   │   │   └── Settings.jsx
│   │
│   ├── routes/
```

```
│   │   ├── PublicRoutes.jsx
│   │   └── PrivateRoutes.jsx
│   │
│   ├── context/
│   │   ├── AuthContext.jsx
│   │   ├── PlannerContext.jsx
│   │   └── ThemeContext.jsx
│   │
│   ├── hooks/
│   │   ├── useAuth.js
│   │   ├── usePlanner.js
│   │   └── useDebounce.js
│   │
│   ├── services/
│   │   ├── api.js
│   │   ├── authService.js
│   │   ├── plannerService.js
│   │   └── analyticsService.js
│   │
│   ├── utils/
│   │   ├── constants.js
│   │   ├── helpers.js
│   │   └── dateUtils.js
│   │
│   ├── styles/
│   │   └── global.css
│   │
│   ├── App.jsx
│   └── main.jsx
│
├── package.json
└── vite.config.js
```

---

# Backend Folder Structure:

**Tech: Node.js + Express + MongoDB**

```
server/
│
├── src/
│   ├── config/
│   │   ├── db.js
│   │   └── env.js
│   │
│   ├── controllers/
│   │   ├── auth.controller.js
│   │   ├── user.controller.js
│   │   ├── planner.controller.js
│   │   ├── task.controller.js
│   │   ├── analytics.controller.js
│   │   └── ai.controller.js
│   │
│   ├── models/
│   │   ├── User.js
│   │   ├── StudyPlan.js
```

```
│   │   ├── Task.js
│   │   ├── StudySession.js
│   │   └── Analytics.js
│   │
│   ├── routes/
│   │   ├── auth.routes.js
│   │   ├── user.routes.js
│   │   ├── planner.routes.js
│   │   ├── task.routes.js
│   │   ├── analytics.routes.js
│   │   └── ai.routes.js
│   │
│   ├── middlewares/
│   │   ├── auth.middleware.js
│   │   ├── error.middleware.js
│   │   └── validate.middleware.js
│   │
│   ├── services/
│   │   ├── planner.service.js
│   │   ├── analytics.service.js
│   │   └── ai.service.js
│   │
│   ├── utils/
│   │   ├── generateSchedule.js
│   │   ├── calculateProgress.js
│   │   └── sendEmail.js
│   │
│   ├── app.js
│   └── server.js
│
├── package.json
└── .env
```

---

# AI Engine Folder Structure:

**Future scalable microservice approach:**

```
ai-engine/
│
├── src/
│   ├── planners/
│   │   ├── smartPlanner.js
│   │   ├── adaptivePlanner.js
│   │   │
│   ├── analyzers/
│   │   ├── performanceAnalyzer.js
│   │   ├── weaknessDetector.js
│   │   │
│   ├── recommenders/
│   │   ├── studyRecommendation.js
│   │   └── resourceRecommendation.js
│   │
│   ├── prompts/
│   │   └── aiPrompts.js
│   │   │
```

```
│   └── index.js
│
└── package.json
```

---

# GitHub Collaboration Methods:

## ✅ Method 1: Add as Collaborators (Simple & Fast)

### ◇ Step 1: GitHub Repo Open karo

- Apne project repository me jao
- **Settings** → **Collaborators** → **Add people**

### ◇ Step 2: Friend ka GitHub username add karo

Unhe invite milega → wo accept karenge → done.

### ◇ Kaise kaam karega?

- Wo directly repo clone kar sakte hain
- New branch banayenge
- Code push karenge
- PR create karenge
- Tum review karke merge karoge

---

### 🔥 Professional Workflow (Highly Recommended)

Unhe bolo:

```
git checkout -b feature/login-system
git add .
git commit -m "Added login system"
git push origin feature/login-system
```

**Phir:**

- GitHub pe jaake **Pull Request create karenge**
- Tum review karoge
- Merge after approval

---

# ✅ Method 2 (More Professional – Recommended for Resume Projects)

### 🤓 Use Fork + Pull Request Model

**Yeh open-source style hota hai.**

### ◇ Step 1:

**Friends:**

- Tumhara repo fork karenge
- Apne account pe copy ban jayegi

### ◇ Step 2:

Wo apni forked repo me changes karenge

### ◇ Step 3:

Wo **Pull Request bhejenge original repo me**

---

### 🤔 Tumhare case me kaunsa best hai?

**Since:**

- Yeh team project hai
- Private collaboration hai
- Resume project hai

👉 **Collaborator + Feature Branch Workflow best hai**

---

# 🔥 Extra Professional Setup (Impress Interviewers)

### 1️⃣ Branch Protection Enable karo

**Repo → Settings → Branches → Add rule**

**Protect:**

```
main branch
```

**Enable:**

- Require pull request before merging
- Require review approval

**Issue:**
❌ Direct push to main allowed nahi hoga
✓ PR mandatory hoga

---

### 2️⃣ Proper Branch Strategy Use karo

```
main → production ready
develop → integration branch
feature/* → new features
bugfix/* → bug fixes
```

**Example:**

```
feature/ai-scheduler
feature/auth-system
bugfix/login-error
```

---

# 🔥 Tum ek kaam aur kar sakte ho

**Ek `CONTRIBUTING.md` file bana do repo me:**

```
# Contribution Guidelines

- Always create feature branch
- Do not push directly to main
- Write proper commit messages
- Create Pull Request with proper description.
```