**STUDY BUDDY**

**Team members:**

Ameena Tabassum

Ayesha Masrath

A.Sadhwika

## Introduction

**Study Buddy** is a smart learning companion designed to help students organize, collaborate, and improve their academic journey. In today’s fast-paced education system, students often face challenges like managing study schedules, clarifying doubts, tracking progress, and finding the right resources. Study Buddy solves these problems by providing a **personalized digital assistant** that supports both individual and group learning.

It works as a **platform where students can create their study profiles, set academic goals, share resources, ask questions, and collaborate with peers**. With features like progress tracking, reminders, interactive discussions, and AI-based guidance, Study Buddy acts like a **virtual friend who motivates, guides, and helps students stay on the right path**.

By combining **technology and education**, Study Buddy enhances productivity, builds collaboration, and creates a supportive learning environment where students don’t feel alone in their academic journey.

**2. Problem Statement**

Students often struggle with:

* Managing study schedules & tasks.
* Finding reliable peer support or doubt clarification.
* Staying motivated during self-study.
* Difficulty in organizing notes, courses, and progress.

**Why it’s important:**

* With online learning and self-paced courses, students lack real-time guidance and accountability.
* Having a “study buddy” boosts productivity, collaboration, and motivation.

**Real-world example:**  
A university student preparing for exams spends hours searching for answers on forums. Instead, a “Study Buddy” app can instantly connect them with peers and AI-powered assistance.

## ****3.Solution / Idea****

**Study Buddy** is a web/mobile app designed to act as a smart companion for students.

* Provides a **personalized study profile**.
* Offers **peer-to-peer doubt solving**.
* Suggests **AI-powered learning assistance**.
* Helps track progress, tasks, and interests.

**What makes it unique?**

* Combines **AI + peer learning** in one platform.
* Focuses not only on academics but also on **motivation and accountability**.

# **4.How to Run the Program**

# Follow the steps below to set up and run the application using Visual Studio Code (VS Code).

# ---1. Open the Project in VS Code

# Launch Visual Studio Code.

# Open the project folder (e.g., VirtualStudyBuddyFinder) in VS Code.

# Open any file from the folder (this ensures VS Code recognizes the workspace).

# 2. Open Integrated Terminal

# Inside VS Code, go to the top menu:

# View → Terminal (or press `Ctrl + ``).

# This opens the integrated terminal at the bottom of VS Code.

# 3. Navigate to the Server Directory

# In the terminal, change the directory to the server folder:

# **cd server**

4. Install Dependencies (Backend)

Run the following command to install required packages:

npm install

5. Open a New Integrated Terminal for Frontend

Keep the backend terminal running.

Open another integrated terminal in VS Code (+ icon in the terminal tab).

In this new terminal, install dependencies for the frontend:

npm install

6. Start the Application

In the backend terminal, run the server:

npm run

In the frontend terminal, run the client:

npm run

7. Access the Application

Once the backend and frontend are running, VS Code will show the host link (e.g., http://localhost:3000).

Open this link in your browser.

The frontend will load, and index.html will open automatically.

# **5. **Working of Study Buddy****

### ****1. User Onboarding****

* A student signs up using email/Google account.
* Creates a **personal profile**: name, university, courses, interests.
* Profile acts as their “digital identity” in the platform.

### ****2. Study Profile & Dashboard****

* Student is redirected to their **dashboard**.
* Dashboard shows:
  + Ongoing courses
  + Study tasks / to-do list
  + Peer connections
  + Quick access to AI assistant

### ****3. Peer-to-Peer Interaction****

* Students can join **study groups** based on subjects or interests.
* Real-time **chat/ask questions** to peers.
* Option to upvote best answers (like StackOverflow but student-focused).

### ****4. AI Learning Assistant****

* Students can ask doubts to the AI bot (e.g., “Explain quicksort with an example”).
* AI provides **short, clear answers**, diagrams, or references.
* Acts as a **backup buddy** when no peer is available.

### ****5. Study Planning & Task Tracker****

* Students can add tasks like:
  + “Revise Unit 2 by tonight”
  + “Finish ML assignment by Friday”
* Tasks show deadlines, reminders, and progress.
* Gamification: Completing tasks earns points/badges.

### ****6. Real-time Collaboration (Optional Add-on)****

* Students can create **study rooms**.
* Each room can have:
  + Timer (Pomodoro-style study sessions).
  + Shared notes.
  + Whiteboard (future extension).

### ****7. Backend Processing****

* **Authentication:** Handled by Firebase Auth.
* **Data Storage:** Courses, chats, notes stored in Firebase/PostgreSQL.
* **AI Queries:** Sent to OpenAI API → response returned in real-time.
* **Notifications:** Push notifications for task reminders & replies.

### ****8. Output / Student Benefits****

* Students save time in searching for answers.
* They feel motivated by group accountability.
* Get **both AI + peer support**, making learning more effective.
* Personalized insights → shows their **progress over time**.

## ****6. Features****

**Must-Have Features:**

* Student Profile Creation (courses, interests).
* Doubt-Solving Community (chat/forums).
* Task & Study Schedule Tracker.
* AI Assistant for quick explanations.

**Nice-to-Have Features:**

* Gamification (rewards, badges).
* Smart Recommendations (study material, courses).
* Real-time group study rooms with timers.
* Integration with Google Calendar & Notes apps.

## ****7. Technical Details / Tech Stack****

**Frontend:** React.js (web), Flutter (mobile).  
**Backend:** Node.js with Express.  
**Database:** Firebase (real-time data), PostgreSQL for structured data.  
**Tools / Libraries:**

* OpenAI API (AI learning assistance).
* Firebase Auth (login/signup).
* TailwindCSS (styling).
* Socket.io (real-time chat).

## ****8. Demo / Screenshots****

* UI Screens: Login, Profile Setup, Dashboard, Chat Room.
* Short Demo Video Link: [Add YouTube/GDrive link]

## ****9. Results / Impact****

* Created a working MVP in the hackathon timeline.
* Successfully tested peer-to-peer chat & AI assistant.
* Positive feedback: “Helps save time & makes studying engaging.”

**Potential Impact:**

* Can be scaled into a **university-wide study platform**.
* Reduces academic stress & improves peer support.

## ****10. Challenges / Learnings****

**Challenges Faced:**

* Integrating real-time chat with Firebase.
* Managing AI response accuracy within limited API calls.
* Time constraints during hackathon.

**Learnings:**

* Importance of **team collaboration under deadlines**.
* Got hands-on with **real-time databases & API integration**.
* Improved knowledge of **UI/UX design for students**.

## ****11. Future Work / Next Steps****

* Add **video chat + collaborative whiteboard**.
* Improve AI with **personalized recommendations**.
* Scale for **cross-university collaborations**.
* Add **mobile-first optimization & offline mode**.

## ****12. References / Acknowledgments****

* **Libraries & APIs:** Perplexity, OpenAI API, Socket.io, TailwindCSS,React js .
* **Mentors/Guides:** Hackathon mentors, online open-source communities.