

A Software Engineering Project On

STUDY BUDDY

as the partial fulfillment of

Semester - VI study

for the degree of

Bachelor of Science (Information Technology)

(Affiliated to Sarvajanik University)

during the academic year 2024-2025.

Developed By:

Hunny Parmar

Dhruvi Mehta

Stuti Shah

Princy Shah

External Guide:

Mr. Harmish Patel

Internal Guide:

Mrs. Charmy Patel



Sarvajanik University

Shree Ramkrishna Institute of Computer Education and Applied Sciences

Behind P.T Science College, M.T.B. College Campus, Athwalines, Surat-395 001



Shree Ramkrishna Institute of Computer Education and Applied Sciences
Behind P.T. Science College, M.T.B College Campus, Athwalines, Surat-395 001.



CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE

This is to certify that
Hunny Parmar

Exam no:
SR22BSIT129

has successfully completed her software engineering project work entitled

STUDY BUDDY

as the partial fulfillment of

Semester -VI study
for the degree of

Bachelor of Science (Information Technology)

(Affiliated to Sarvajani University)
during the academic year 2024-2025.

Internal Project Guide

HOD
Dept. of Computer Science

Date : _____

Place : Surat

University Examination:

Examination No. : _____

Signature of External Examiners : _____

Internship Provisional Certificate

Private and Confidential

Date: 22-04-2025

Hunny Parmar

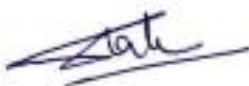
Enrollment Id: SR22BSFT129
Sarvajani University, Surat.

To whom it may concern

This certifies that Hunny Parmar is continuing with internship training in React Js Services in our organization under our Jr. Software Developer- Harmish Patel.

The provisional date for the Internship is 11-11-2024 to 22-04-2025. Her internship activities include Learning html, CSS, JavaScript, tailwind, ReactJS, typescript, Json, and Implementation of the same in Projects.

During her internship program with us, she has been exposed to different processes and was found diligent, hardworking, and inquisitive.



Vishal Shah
HR | Operations Head



INDEX

SR. NO	Contents	Page no.
1.	Introduction	1
	• Company Profile	2
	• Training Activities	3
	• Project Profile	8
2.	System Introduction	10
	• System Definition	11
	• System Objective	11
	• Modules/Functionality Description	13
3.	Requirement Analysis & Modelling	17
	• Use Cases, Activity Diagrams, Sequence Diagrams	18
4.	Design	24
	• Interface Design	25
	• System Architecture	37
	• System Flowchart	38
5.	Code Methodology	41
	• Code Rules/ Code guidelines/ Codestyles	42
6.	Testing	43

7.	Future Enhancement	46
8.	Bibliography & References	49

Introduction

❖ Company Profile

Name : Toshal Infotech

Toshal Infotech Pvt. Ltd., formerly known as "Idea To Working," is a privately held IT services and consulting company based in Surat, Gujarat, India. Established in 2011, the company specializes in delivering innovative software solutions tailored to meet diverse business needs.

Mission: To convert ideas into reality by providing cutting-edge technology solutions that drive business growth and efficiency.

☐ Services Offered:

- Custom Software Development: Tailored software solutions to meet specific business requirements.
- Web Development: Design and development of responsive and user-friendly websites.
- Mobile App Development: Creation of mobile applications for various platforms.
- E-commerce Solutions: Development of online shopping platforms with integrated payment systems.
- Digital Marketing: Services including SEO, social media marketing, and email campaigns to enhance online presence.
- UI/UX Design: Designing intuitive user interfaces and experiences.
- System Integration: Integrating various software systems to work seamlessly together.

☐ Technologies and Expertise:

- Programming Languages & Frameworks: ASP.NET, Vue.js, Laravel, Node.js, and more.
- Mobile Platforms: iOS and Android development.
- CMS Platforms: DotNetNuke and custom CMS solutions.
- Databases: SQL, PostgreSQL, and MongoDB.
- Cloud Services: AWS for scalable and secure cloud solutions.
- Others: Expertise in Zendesk UI customization, HTML5, CSS, JavaScript/jQuery, and ERP development.

☐ Locations:

- Surat Office: 203, 5th Floor, SNS Interio, Althan Bhatar Road, Surat, Gujarat, 395017, India.

☐ Contact Information:

- Phone: +91 7575062161
- Email: career@toshalinfotech.com
- Website: www.toshalinfotech.com

❖ Training Activities

➤ Internship Overview(Training Work) :

As a ReactJs Intern at Toshali Infotech, I worked on developing and optimizing Web Applications by implementing scalable Front-End Solutions, contributing to enhanced performance and User Interface. I learn many languages like Html,Css,JavaScript,ReactJs and complete many tasks During Internship Learning Phase.

➤ Key Responsibilities:

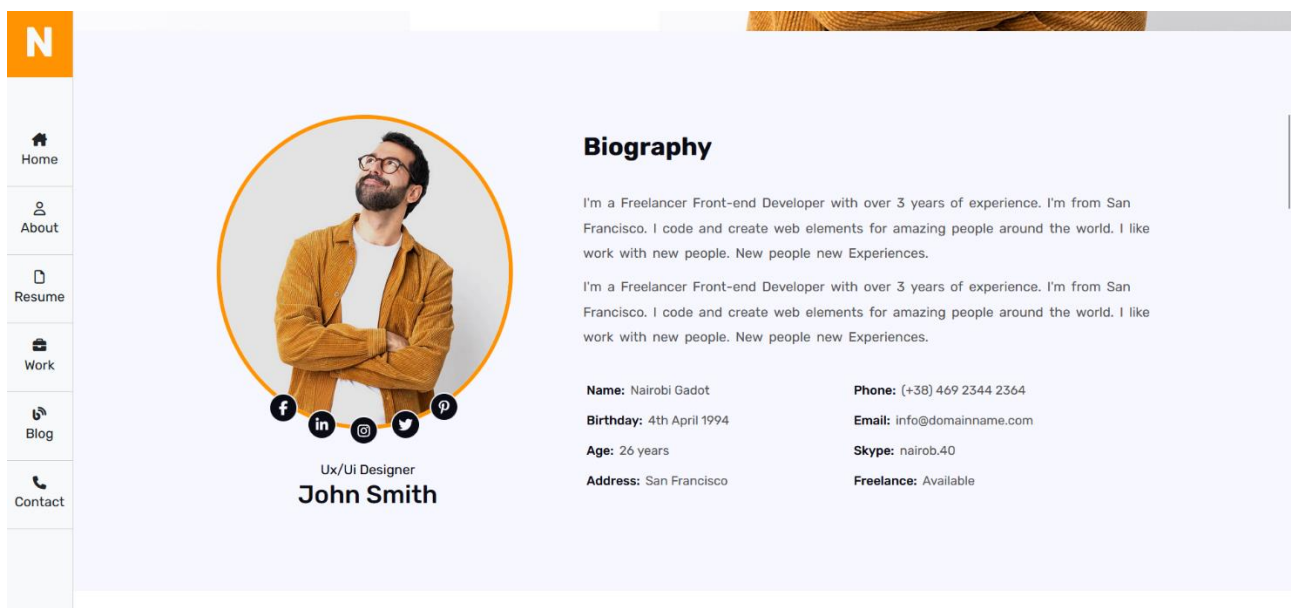
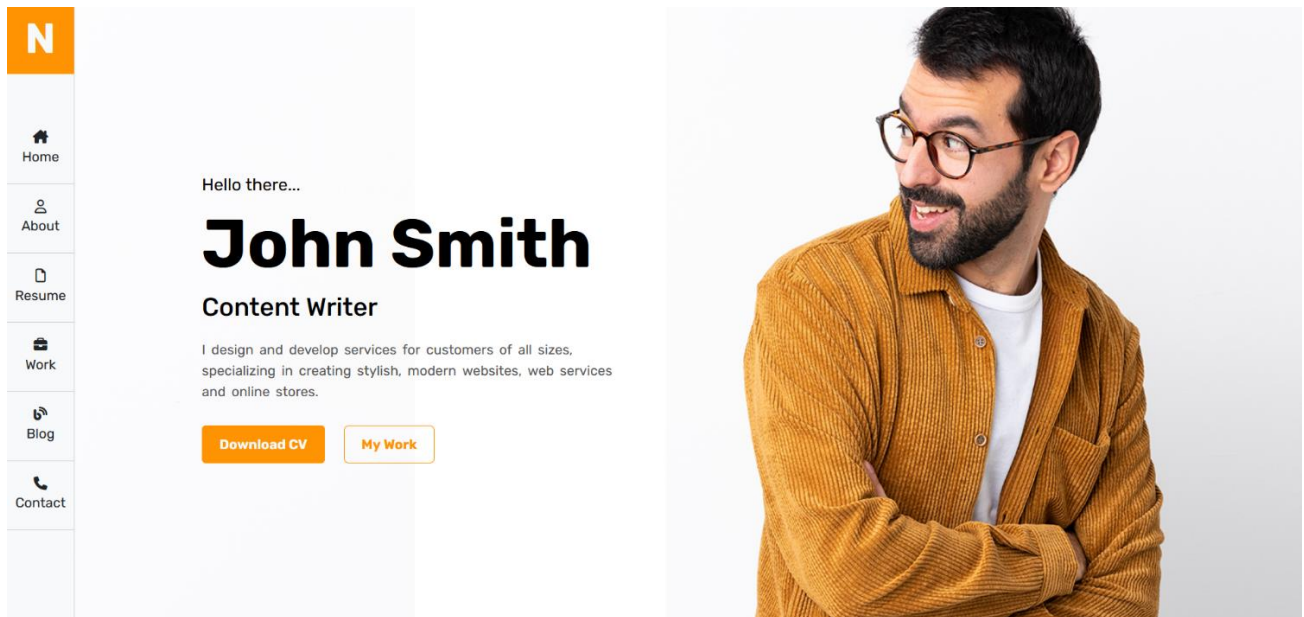
- Developed and maintained scalable front-end and components using modern web 2.1.2
Challenges Faced: technologies.
- Collaborated with cross-functional teams to design, develop, and deploy web applications.
Implemented responsive designs to enhance user experience.
- Integrated APIs and third-party services to improve application functionality.
- Participated in code reviews and contributed to improving development best practices.

➤ Challenges Faced:

- Handling Complex Debugging: Encountered issues with application performance and debugging intricate front-end requiring extensive troubleshooting and testing.
- Optimizing API Performance: Faced challenges in integrating and optimizing third-party APIs, ensuring seamless data flow and minimal latency.
- Version Control Conflicts: Managed and resolved Git merge conflicts while collaborating with development team

➤ Internship Tasks :

- Html,Css,JavaScript task : Learn Html,Css,JavaScript from Youtube tutorials and Documentation. Design Responsive demo site using Html,Css,JavaScript.



Experience.

Sr. Front-end Engineer

Google | Remote
Feb 2021 - Present

Full Time

Google LLC

Adipiscing Lorem ipsum dolor sit amet, consectetur elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor.

Visual Designer

Ridian | Remote
Mar 2020 - Present

Part Time

Ridian IT

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Animation Artist

IBThemes | Remote
Jan 2019 - Present

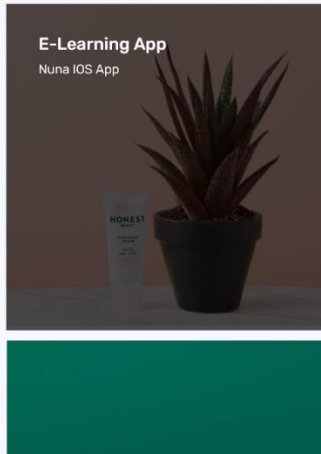
Full Time

IB Themes

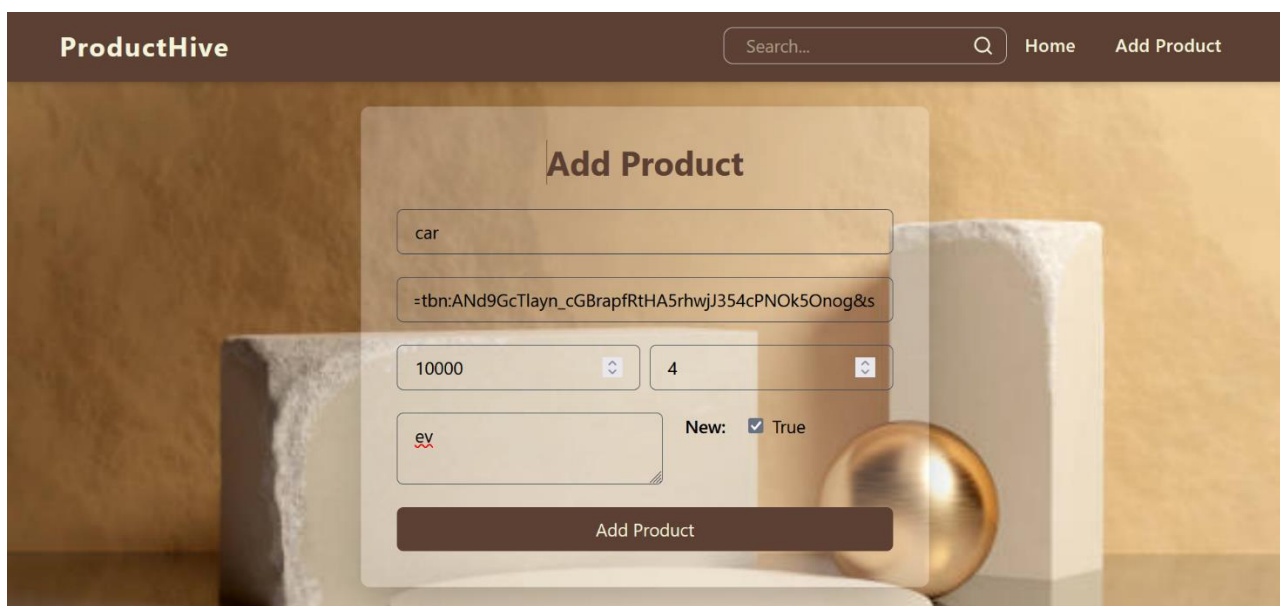
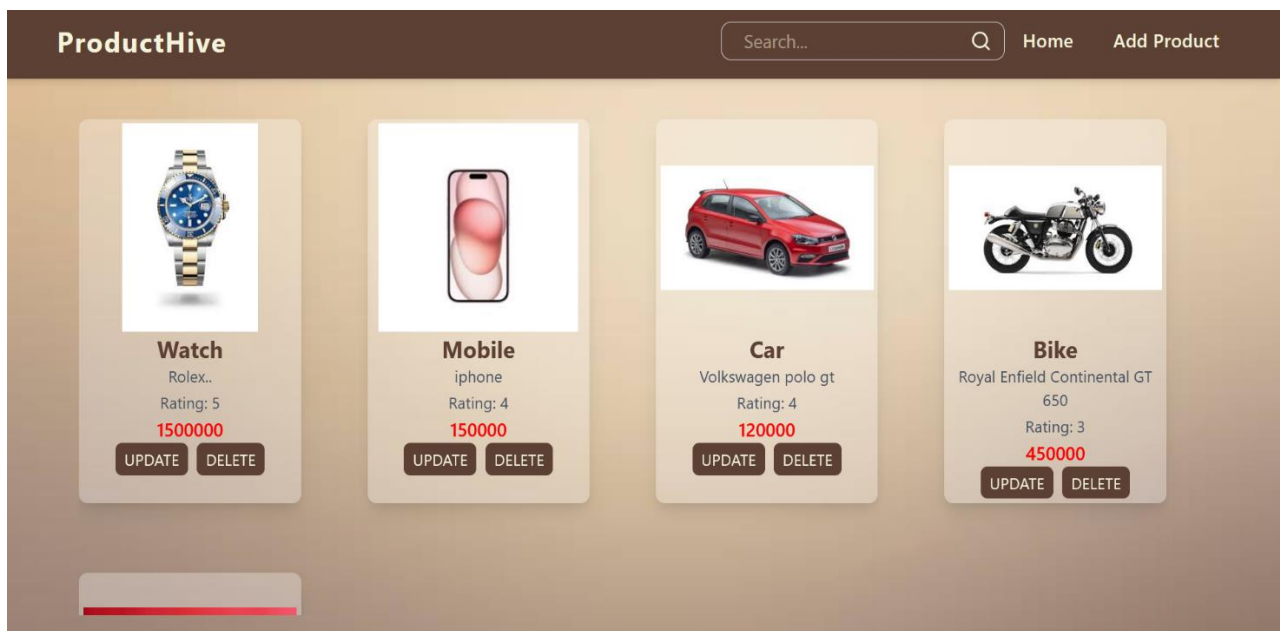
Consectetur adipiscing elit,lorem ipsum dolor sit amet, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor.

My Portfolio.

All Branding Photography Fashion Product



- ReactJs task : Learn ReactJs from Youtube tutorials and Documentation. CRUD operation with dummy api of products using ReactJs. Use redux store for state management. And also used tailwind CSS for UI.



Edit Product

shoes

<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9Gc>

10000

4

white color

New: ☒ True

Update Product



Project Profile

Project Title :	Study Buddy
Objective :	The objective of StudyBuddy – Your Academic Companion is to develop a smart, student-focused platform that enhances collaborative learning. The project aims to help students connect with suitable study partners, manage notes using flashcards, generate quizzes from study material, and communicate in real-time—all through a secure and user-friendly interface. It combines AI, chat systems, and productivity tools to support students in achieving academic success.
Frontend Tool :	React JS
Backend Tool :	Node JS
Database :	MongoDB
Tools & Technologies used:	Frontend: React.js, Tailwind CSS, Axios Backend: Node.js, Express.js Database: MongoDB, Mongoose Authentication & Security: JWT, bcryptjs, Helmet, CORS AI Integration: Gemini API (for quiz generation) Real-Time Chat: Socket.io Utilities & Testing: Postman, dotenv Deployment & Version Control: GitHub, Render/Vercel

Duration :	3 months & Ongoing
Team Members:	Hunny Parmar Dhruvi Mehta Stuti Shah Princy Shah

System Introduction

❖ Definition

StudyBuddy – Your Academic Companion is a comprehensive, web-based academic support platform designed specifically for students to improve their learning, revision, and collaboration experience. It serves as a centralized workspace where students can plan their academic tasks, prepare effectively using AI-powered tools, and engage with peers in real time.

The platform offers multiple integrated features:

- To-Do List to manage tasks and study plans
- Flashcards for quick revision and memory enhancement
- AI-Powered Quiz Generator that creates quizzes from uploaded PDFs or user-written notes using the Gemini API
- Smart Match System that pairs students with similar academic interests
- Real-Time Chat and Audio Calling features using Socket.io and WebRTC
- Leaderboard and Progress Tracking for gamified learning and performance monitoring
- StudyBuddy aims to create a collaborative and self-paced study environment, catering to both individual learners and study groups.

❖ Objective

- The primary objectives of the StudyBuddy platform are outlined below, focusing on both individual learning enhancement and collaborative academic engagement:
- To provide a centralized digital academic workspace where students can organize and manage their learning resources and tasks efficiently.
- To enable AI-powered quiz generation by allowing students to upload PDFs or notes, from which relevant questions and multiple-choice answers are auto-generated for practice and self-assessment.

- To offer real-time communication tools such as messaging and audio calls, helping students collaborate, discuss doubts, or engage in group study sessions.
- To track academic progress and engagement, offering insights into user activity, completed quizzes, and ranking students through a leaderboard to boost motivation.
- To implement a smart matching algorithm that connects students with similar subjects, goals, or learning styles, encouraging collaborative peer-to-peer learning

❖ Scope

The StudyBuddy system is designed with the following functionalities and user flows in mind:

User Authentication:

Students can register and securely log in using email and password.

Token-based authentication ensures data privacy and route protection.

Core Modules:

- **Quiz Module:** AI-generated quizzes from uploaded PDFs or manually entered notes.
- **Flashcard Module:** Create and manage digital flashcards for active recall.
- **To-Do Module:** Create, edit, and track study-related tasks.
- **Chat & Audio Call Module:** Real-time messaging with typing indicators and voice calling for enhanced collaboration.

AI Integration:

- Uses Gemini API to process user-uploaded content and generate relevant quiz questions.

- Quiz formats include MCQs, fill-in-the-blanks, and true/false.

Real-Time Communication:

- Uses Socket.io for live chat and WebRTC (via simple-peer) for peer-to-peer audio calls.
- Allows students to interact instantly, making learning more engaging.

Leaderboard & Performance Analytics:

- Tracks completed tasks, quiz scores, and activity levels.
- Displays a leaderboard to foster healthy academic competition among users.
- StudyBuddy is scalable and modular, with potential for mobile support and advanced analytics in future versions.

❖ Modules/Functionality Description

1. Registration

Allows users to create an account by providing personal, educational, and academic goal details.

Field Name	Data Type	Required	Default	Description
ProfilePicture	String	No	null	Stores the URL or path of the user's profile image.
FullName	String	Yes	—	Full name of the user.
UserName	String	Yes	—	Unique username used for identification and chat features.
Email	String	Yes	—	User's email address. Must be unique and valid (validated using validator).
Password	String	Yes	—	User's password (stored securely after hashing).
Country	String	Yes	—	The country where the user resides.
State	String	Yes	—	The state or region of the user.
EducationLevel	String	Yes	—	Represents the user's academic standing (e.g.,

				High School, BSc, MSc, etc.).
Subject	String	Yes	—	Primary subject the user is focused on or interested in.
StudyGoals	String	Yes	—	User's academic goals or areas of focus (e.g., exam prep, concept mastery).
createdAt	Date	Auto	—	Automatically managed timestamp when the record is created.
updatedAt	Date	Auto	—	Automatically managed timestamp when the record is last updated.

2. Login

Enables secure login using email and password. Includes OTP-based forgot password feature for account recovery.

3. To-Do List

Lets users manage academic tasks categorized by status (Backlog, ToDo, In Progress, Completed). Supports drag-and-drop feature for status updates.

Column Name	Data Type	Description
id	string	Unique task identifier
title	string	Title of the task
description	string	Description of the task
dueDate	date	Task due date
status	enum	Task status (Backlog, ToDo, etc.)
userId	string	User identifier

4. Flashcards

Users can create flashcards manually or from uploaded notes. Flashcards can be viewed, filtered, and searched.

Column Name	Data Type	Description
id	string	Unique flashcard identifier
question	string	Flashcard question
answer	string	Flashcard answer
userId	string	User identifier

5. Quiz

Users upload notes or PDFs; Gemini API generates quizzes. Includes scoring, answer tracking, and submission history.

Column Name	Data Type	Description
id	string	Unique quiz identifier
title	string	Title of the quiz
questions	array	List of questions and options
answers	array	List of correct answers
userId	string	User identifier

6. Live Chat

Real-time messaging system with support for text and image sharing. Online user tracking and audio calls enabled via WebRTC and Socket.io.

Column Name	Data Type	Description
id	string	Unique message identifier
senderId	string	Sender's user identifier
receiverId	string	Receiver's user identifier
content	string	The content of the message
timestamp	date	Timestamp of the message

7. Progress Tracking

Tracks user engagement such as number of quizzes taken, flashcards reviewed, and displays this on dashboard.

Derived from Quiz and Flashcard usage logs

8. Leaderboard

Ranks users weekly based on their performance (quiz scores, flashcard activity, etc.). Displays top 10 students.

Aggregated from Quiz and Progress Tracking statistics

9. Profile Page

Users can view and edit their personal info, educational background, and study preferences.

Editable fields: FullName, Email, Country, State, EducationLevel, Subject, StudyGoals, ProfilePicture

10. Logout

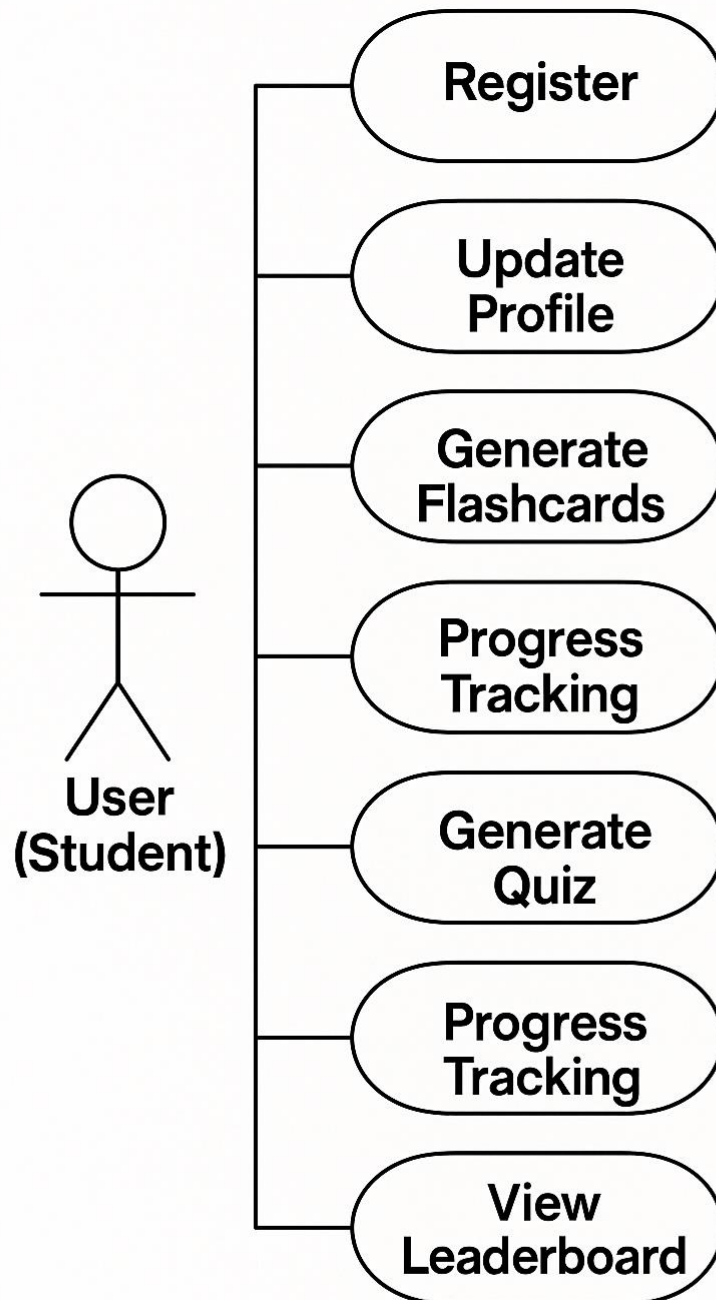
Securely logs out the current user. Accessible from the profile picture dropdown in the navigation bar.

Session/token removal handled in frontend/backend middleware

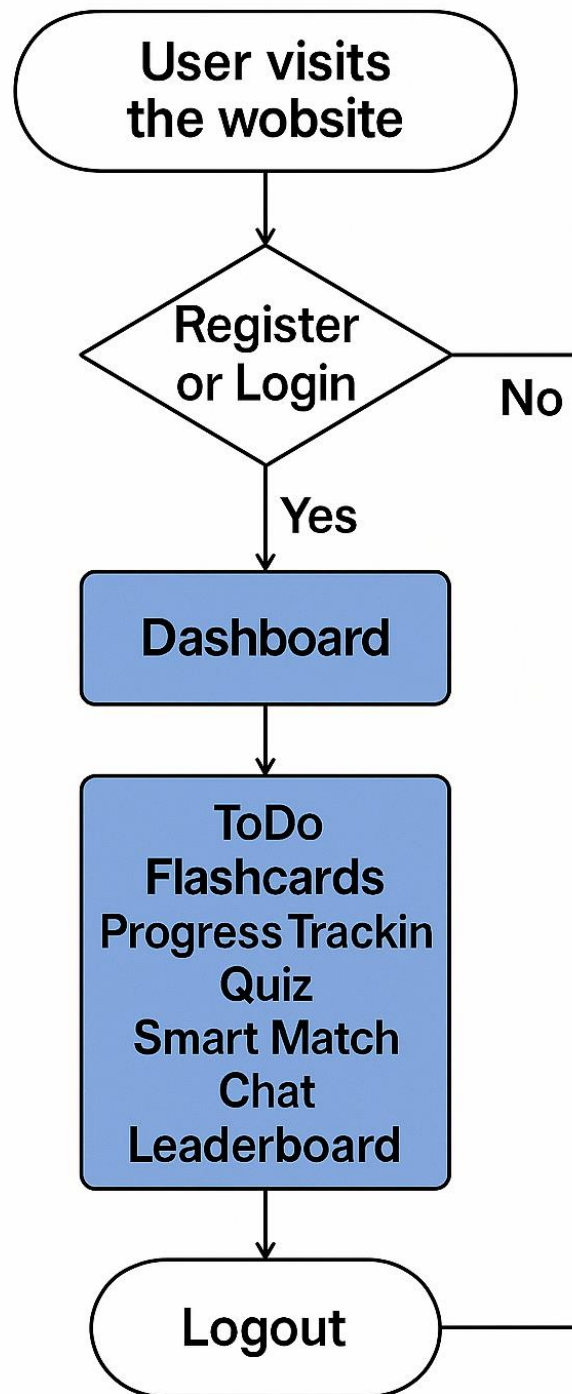
Requirement Analysis & Modelling

❖ Use Case Diagram

User (Student) Use Cases:

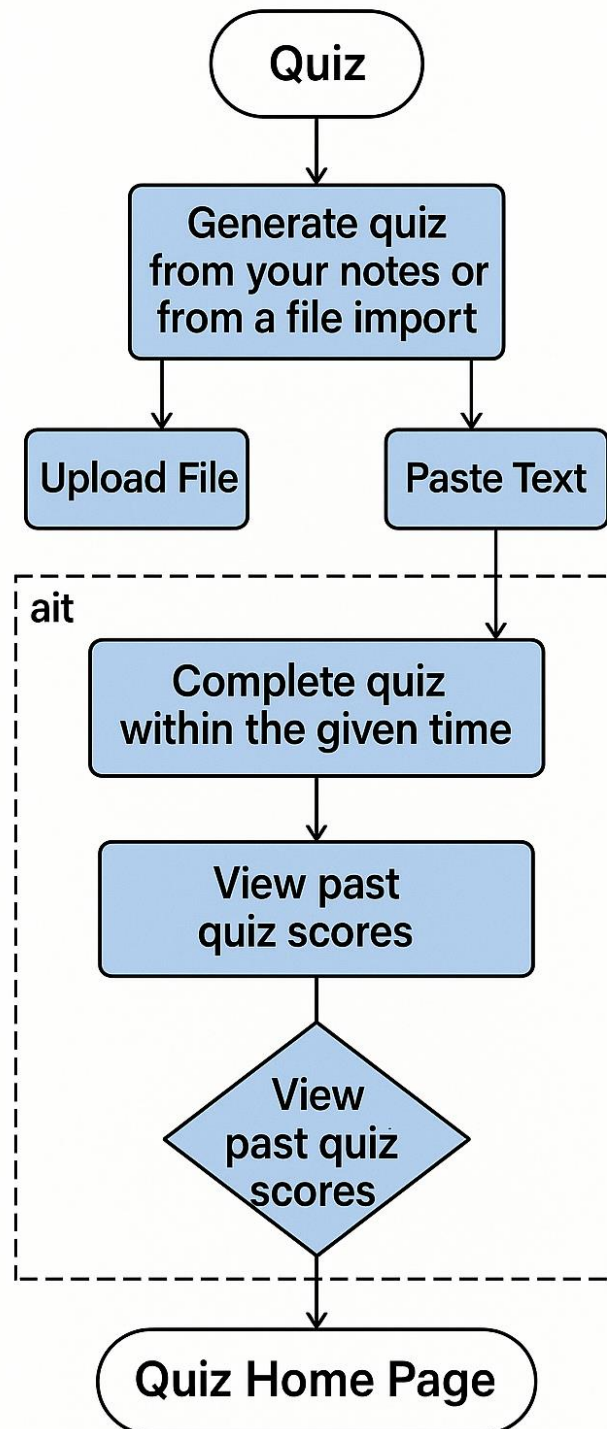


❖ Activity Diagram (User Flow)

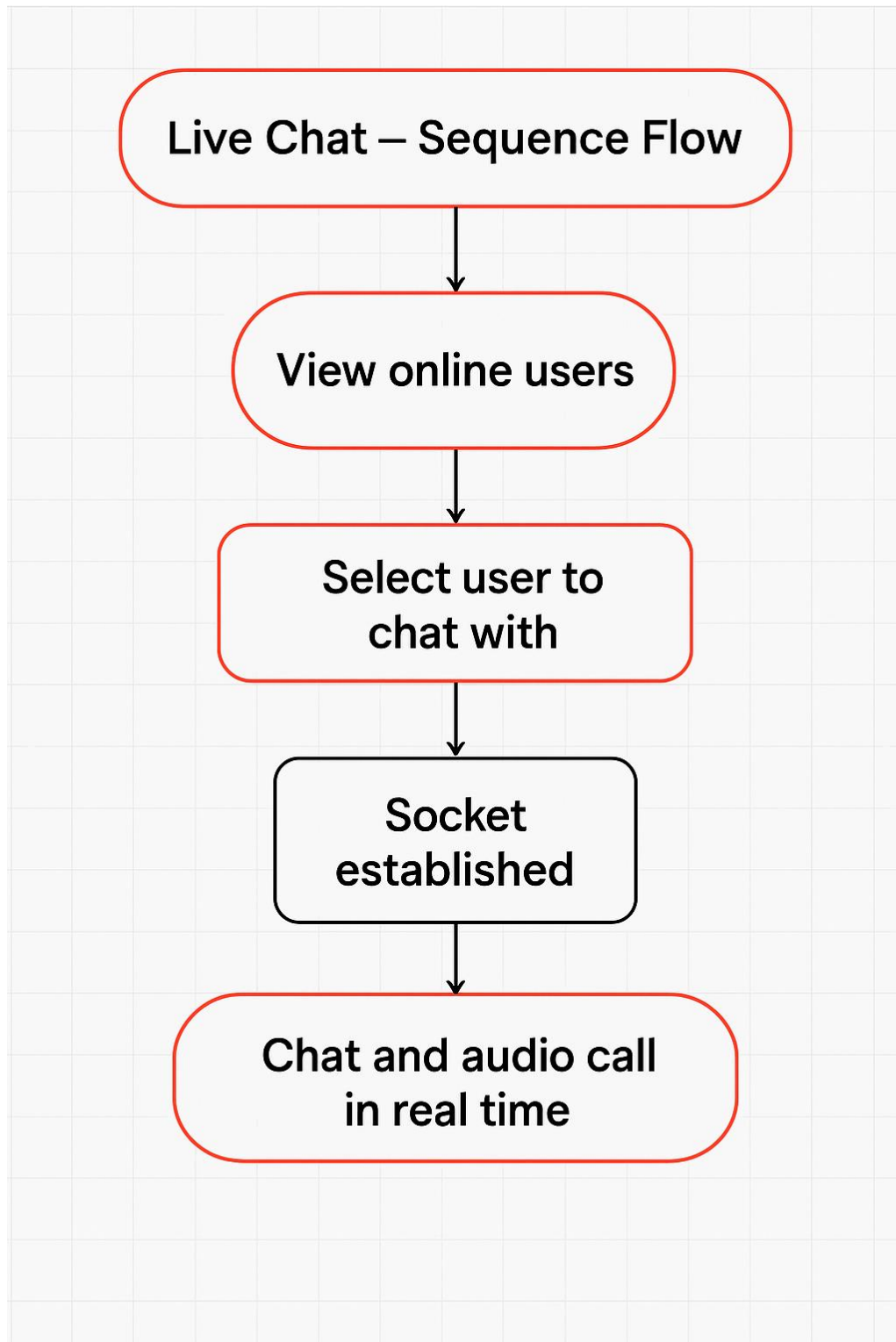


❖ Sequence Diagram

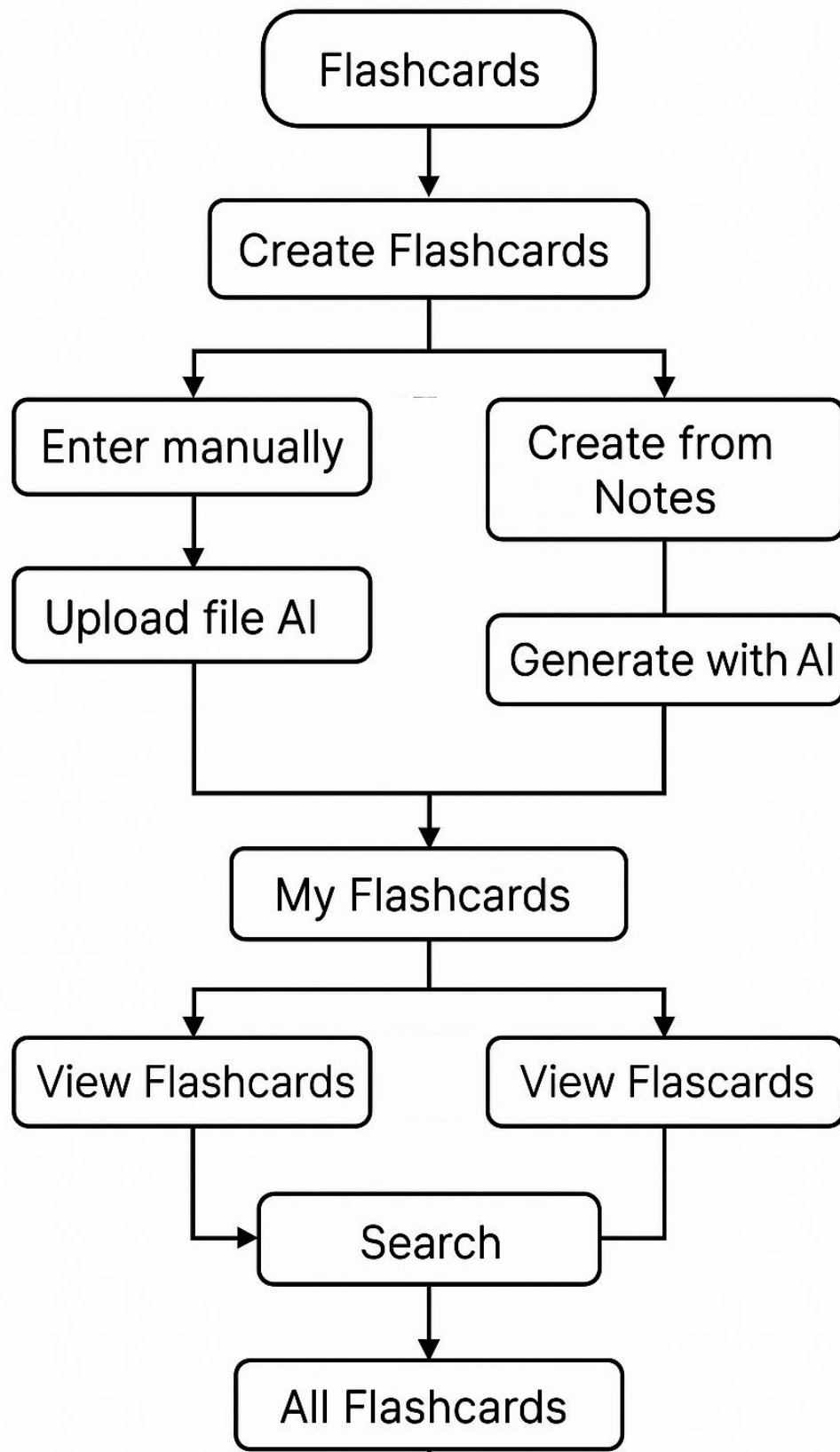
Quiz Feature :-



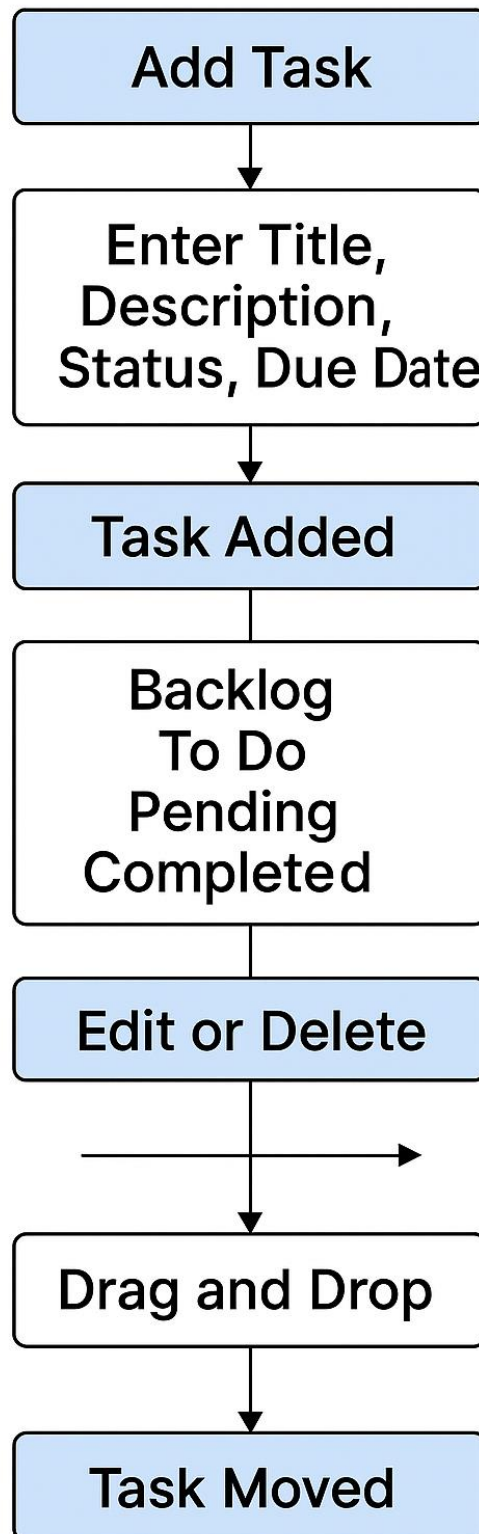
Live Chat Feature :-



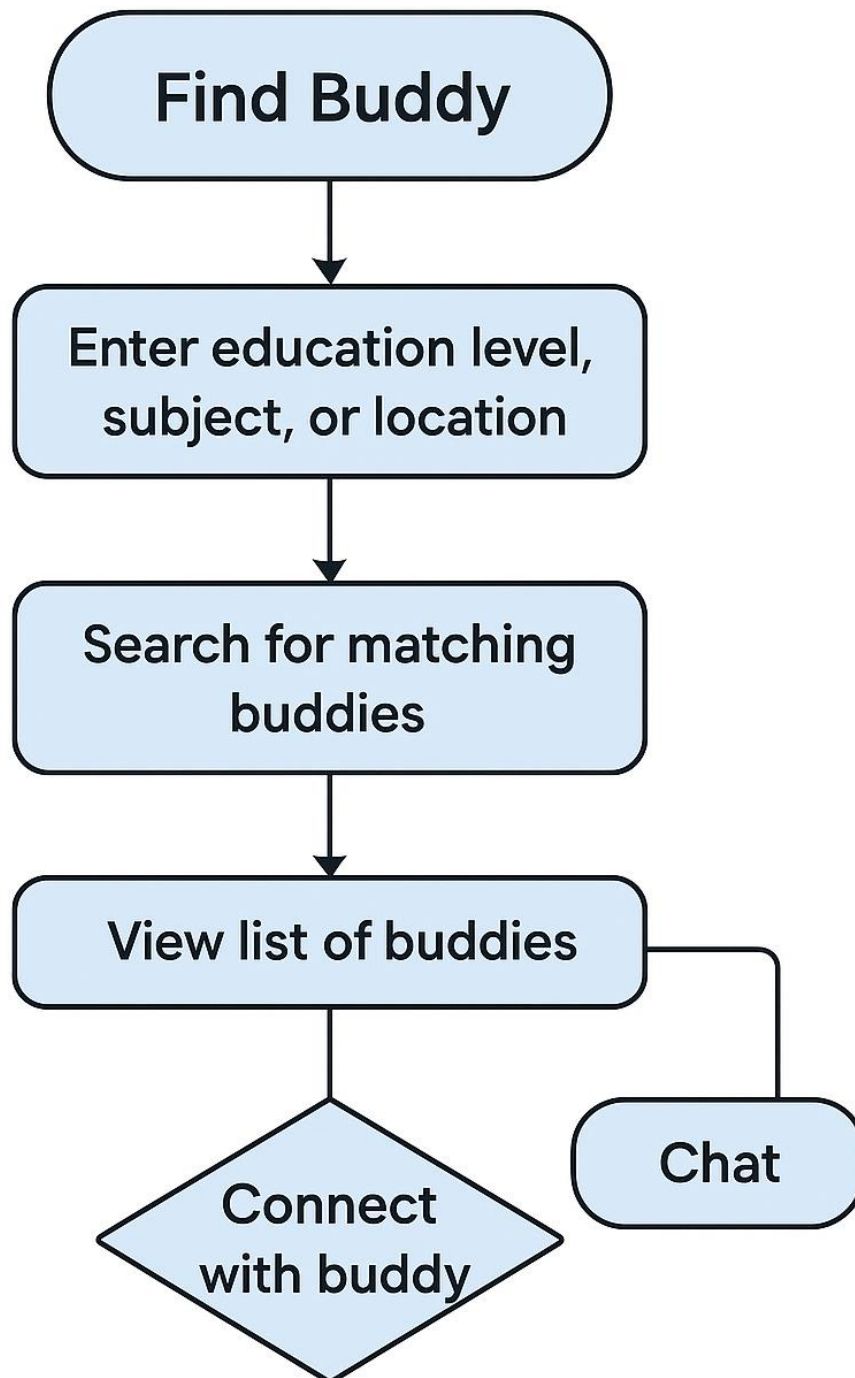
Flashcard Feature :-



ToDo List Feature :-



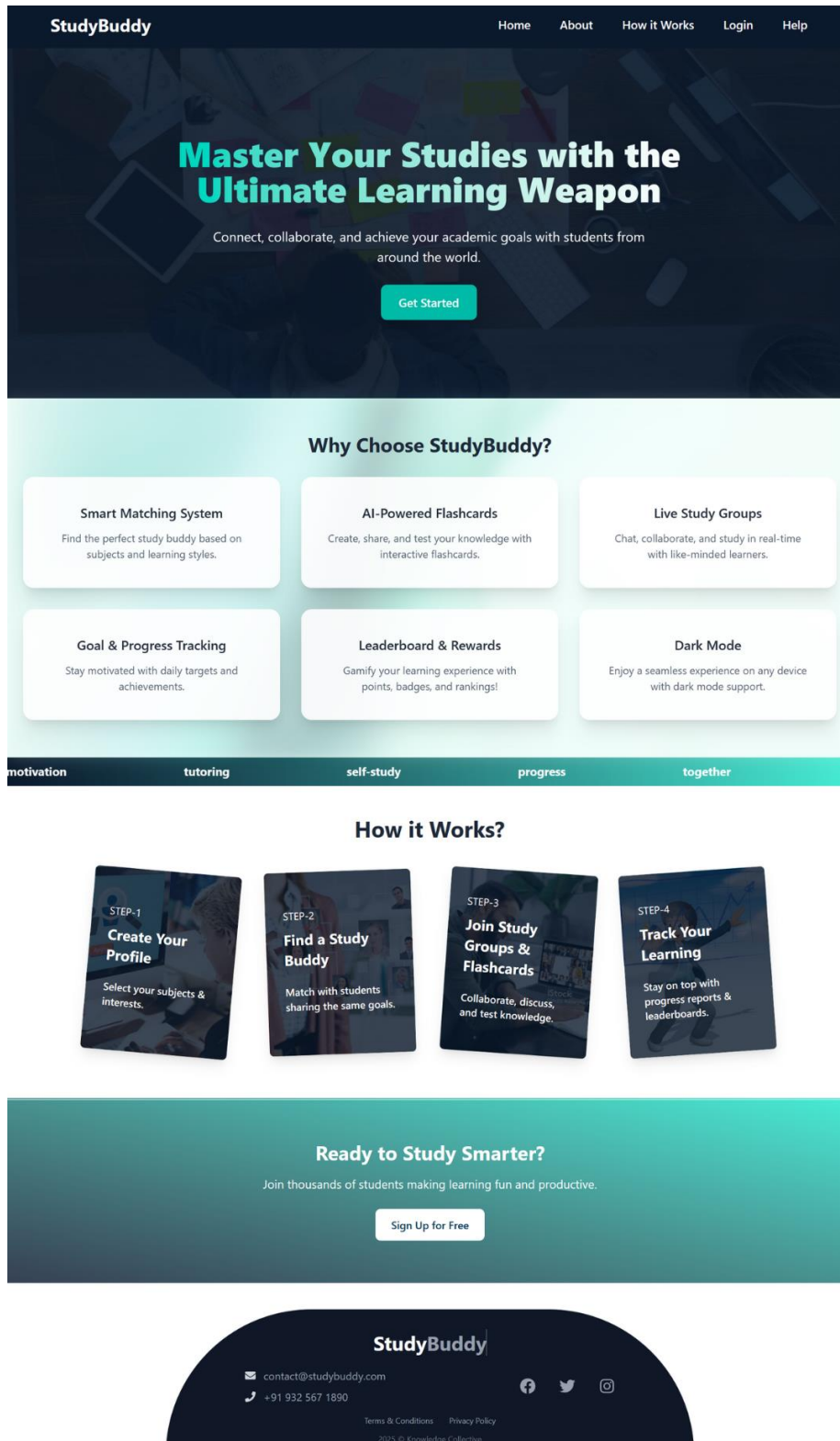
Smart Matching :-



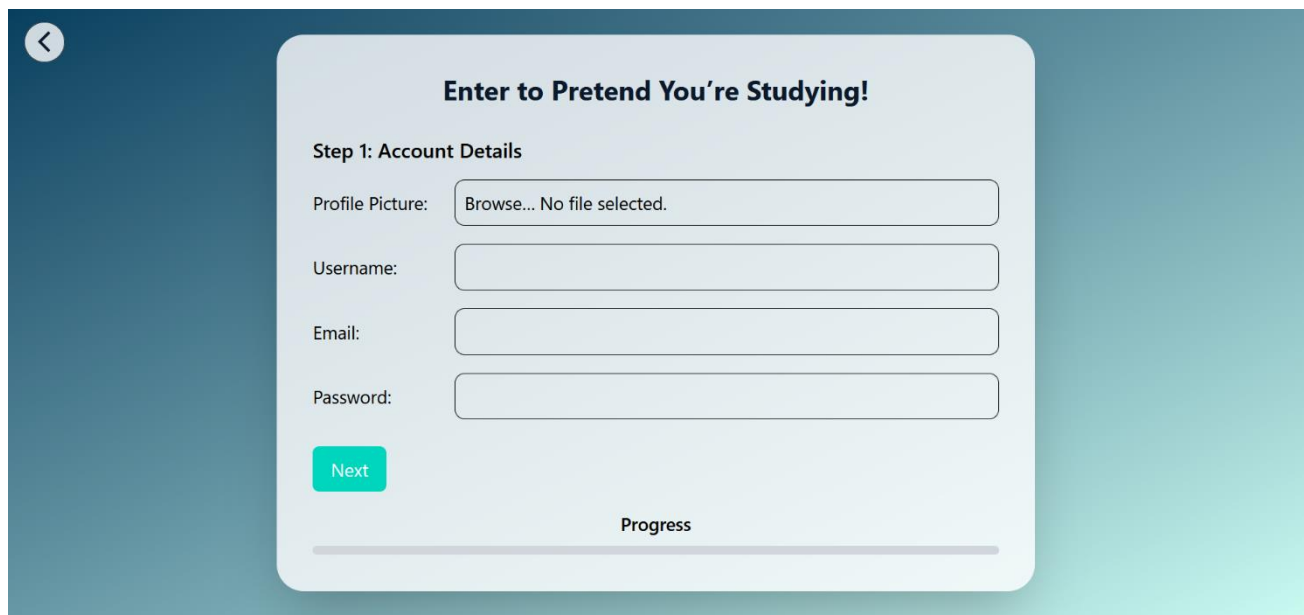
System Design

❖ Interface design

Landing Page :-



Signup Page :-



A screenshot of a mobile app's signup page. The page has a dark blue gradient background. In the top-left corner, there is a white circular button with a left-pointing arrow. The main content is a light blue rounded rectangle with a title "Enter to Pretend You're Studying!" in bold. Below the title is the heading "Step 1: Account Details". There are four input fields: "Profile Picture:" with a "Browse... No file selected." button, "Username:", "Email:", and "Password:". A green "Next" button is at the bottom left of the form. Below the form is a "Progress" indicator with a horizontal line.

Enter to Pretend You're Studying!

Step 1: Account Details

Profile Picture:

Username:

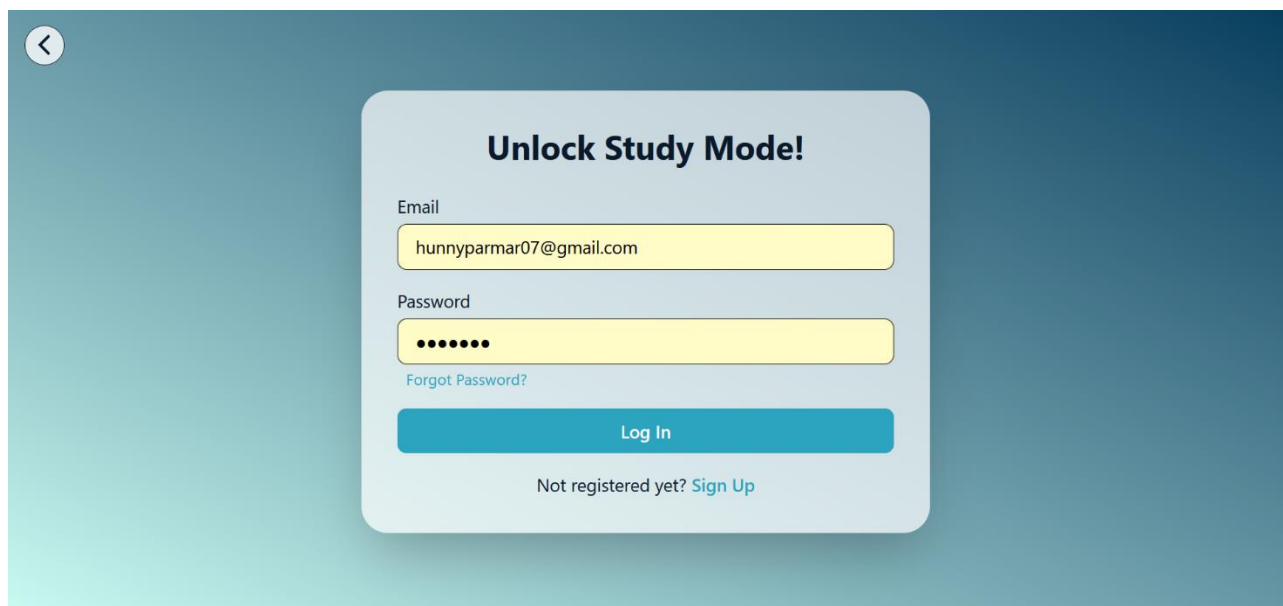
Email:

Password:

[Next](#)

Progress

Login Page :-



A screenshot of a mobile app's login page. The page has a dark blue gradient background. In the top-left corner, there is a white circular button with a left-pointing arrow. The main content is a light blue rounded rectangle with a title "Unlock Study Mode!" in bold. There are two input fields: "Email" with the value "hunnyparmar07@gmail.com" and "Password" with masked characters "••••••". Below the password field is a link "Forgot Password?". A blue "Log In" button is at the bottom of the form. Below the button is the text "Not registered yet? [Sign Up](#)".

Unlock Study Mode!

Email

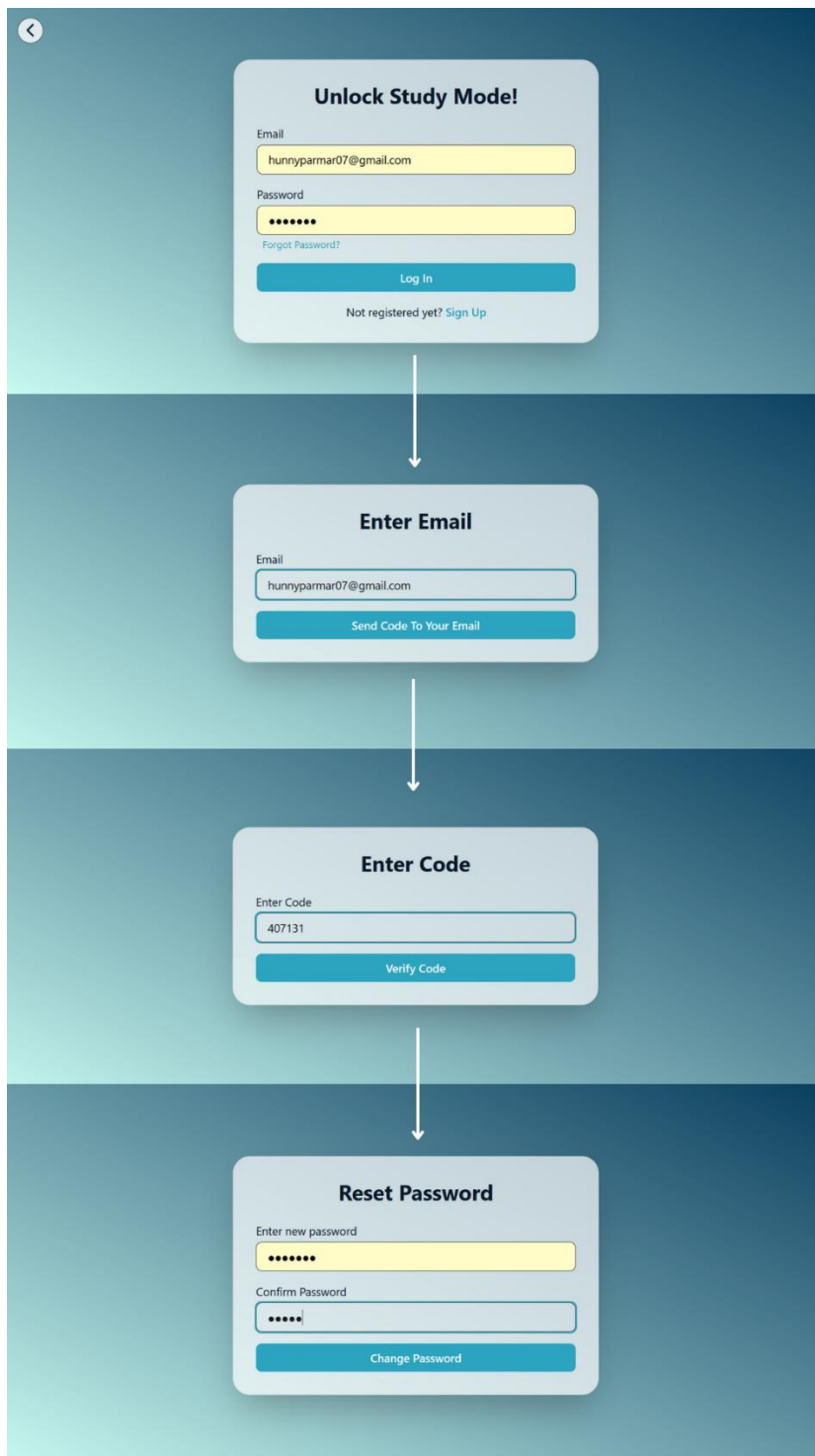
Password

[Forgot Password?](#)

[Log In](#)

Not registered yet? [Sign Up](#)

Forgot Password :-




Profile :-

StudyBuddy

- Home
- To-Do List
- FlashCard
- Find Buddy
- Quiz
- Notifications
- Leader Board

hunny2004 



[Change Picture](#)

Hunny Parmar
hunnyparmar07@gmail.com
Country: AU

Account Settings

[User Information](#)
[User Information](#)
[Location](#)
[Study Preferences](#)

Save Changes

Todo List Page :-

StudyBuddy

Home

To-Do List

FlashCard

Find Buddy

Quiz

Notifications

Leader Board

hunny2004

Task Management

Title

Due Date

Status

Task Title

dd / mm / yyyy

Backlog

Description

Task Description (optional)

Clear

Add Task

Backlog

ToDo

InProgress

Completed

Todo

change the ui

Due: 2025-04-15

Find Buddy Page :-

StudyBuddy

Home

To-Do List

FlashCard

Find Buddy

Quiz

Notifications

Leader Board

hunny2004

Find Study Buddy

Education Level

Subject

Select Country

S

Stuti Udaybhai Shah

@stuti_shah__

Education: 12th complete

Subject: science

Country: IN

D

Dhruvi Mehta

@dhruvi

Education: 12th pass

Subject: node.js

Country: IN

K

Khushi Mehta

@khushi

Education: 12th pass

Subject: node.js

Country: IN

S

S

@s

Education: bscit

Subject: "node"

B

BHAVESH TELI

@Bhavesh

Education: Graduation

Subject: Computer Science

P

Princy Shah

@princy

Education: 12th pass

Subject: node.js

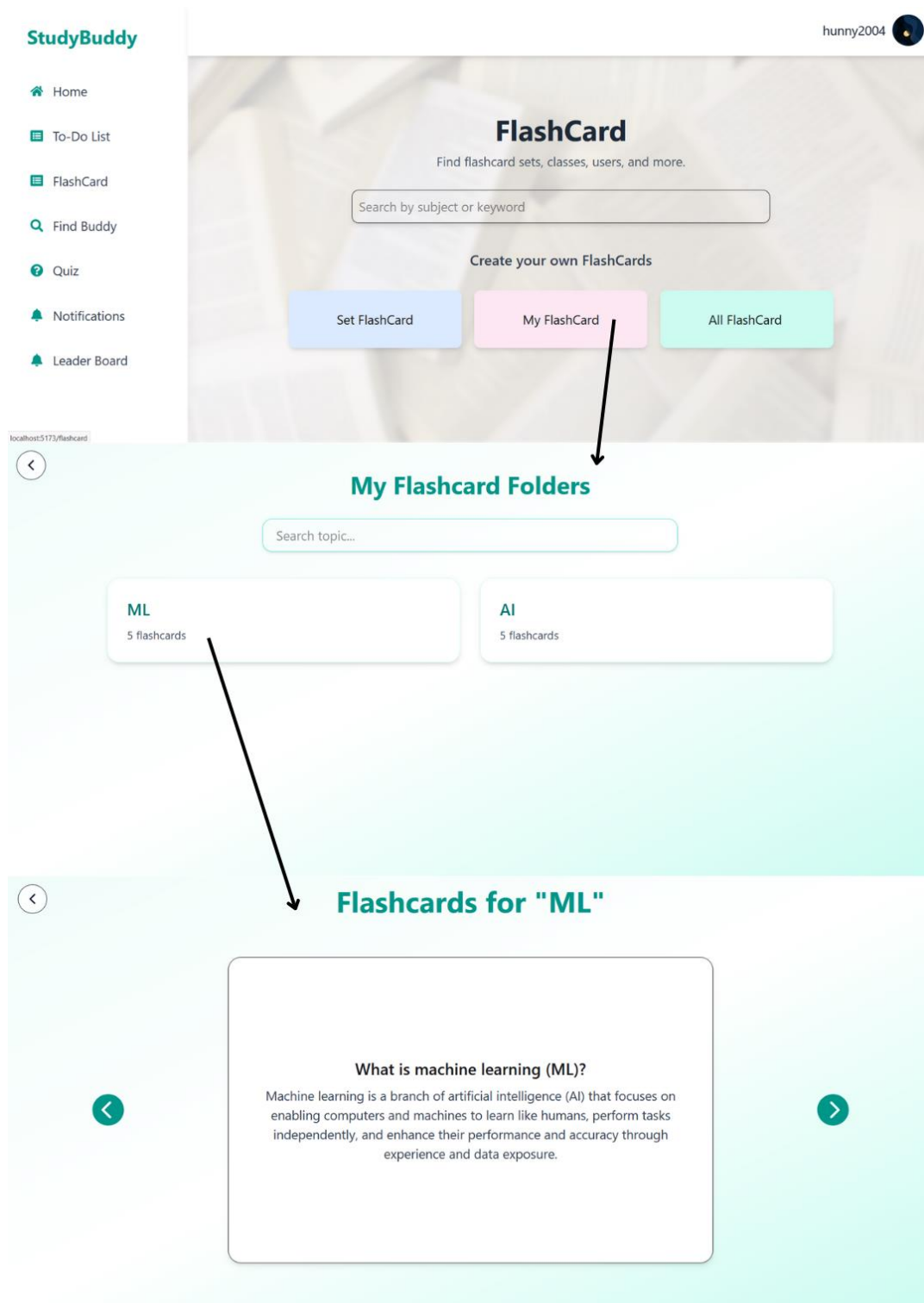
FlashCard :-

Set FlashCard :-

The diagram illustrates the workflow for setting a flashcard on the StudyBuddy platform:

- StudyBuddy Home Page:** The 'Set FlashCard' button is highlighted with an arrow pointing to the next step.
- Modal: How do you want to create your flashcard?** This modal offers two options:
 - Generate from an upload:** This path leads to the 'GENERATE FLASHCARD SET' screen.
 - Create them yourself:** This path leads to the 'CREATE A NEW FLASHCARD SET' screen.
- GENERATE FLASHCARD SET:** This screen allows users to paste text or upload files. It includes a character count (0/100,000) and a 'Generate' button.
- CREATE A NEW FLASHCARD SET:** This screen provides a form to enter a topic, add notes, and create terms and definitions. It includes an 'Import' button and a 'Generate from File' button.

My FlashCard :-



All FlashCards :-

The image shows a two-part screenshot of the StudyBuddy application. The top part is the main FlashCard interface, and the bottom part is a detailed view of a specific flashcard.

Top Screenshot: FlashCard Interface

- Header:** StudyBuddy logo on the left, user 'hunny2004' on the right.
- Left Sidebar:** Home, To-Do List, FlashCard, Find Buddy, Quiz, Notifications, Leader Board.
- Main Content:**
 - FlashCard** title with subtitle 'Find flashcard sets, classes, users, and more.'
 - Search bar: 'Search by subject or keyword'
 - Create your own FlashCards** section with three buttons: 'Set FlashCard' (blue), 'My FlashCard' (pink), and 'All FlashCard' (teal).

An arrow points from the 'All FlashCard' button to the bottom screenshot.

Bottom Screenshot: All Flashcard Folders

- Header:** 'All Flashcard Folders' title with a back arrow.
- Search:** 'Search topic...' bar.
- Folders:** Four cards showing flashcard counts:
 - AI:** 10 flashcards
 - ML:** 5 flashcards
 - Node.js:** 5 flashcards (highlighted with a teal border)
 - React.js:** 5 flashcards

An arrow points from the 'Node.js' folder to the next screenshot.

Bottom Screenshot: Flashcards for "Node.js"

- Header:** 'Flashcards for "Node.js"' title with a back arrow.
- Flashcard Content:**
 - Question:** 'What is Node.js?'
 - Answer:** 'Node.js is a JavaScript runtime environment that allows developers to run JavaScript on the server side.'
- Navigation:** Left and right arrows for navigating between flashcards.

Quiz :-

Upload Files :-

StudyBuddy

hunny2004

Generate a Practice Test

Choose or upload materials to generate practice questions designed for you

Upload files Paste texts View Past Quizzes

Drag and drop notes, readings, lecture slides, etc.
Supported file types are .docx, .pdf, .pptx

Browse files

Generate

Quiz

Time Left: 29:55

1. What type of library is Framer Motion?

A game development library

A data visualization library

A state management library

A declarative animation library

2. What language is Framer Motion specifically designed for use with?

React

Java

Python

JavaScript

3. What company developed Framer Motion?

Microsoft

Facebook

Google

Framer

Submit

GSAP React Spring

Anime.js JQuery Animate

Correct Answer: React Spring

19. Which library is generally preferred for complex animation sequencing and timelines?

GSAP

React Spring

Correct Answer: GSAP

20. What is a primary benefit of using Variants in Framer Motion?

Smaller bundle size Reusability of animation states


Improved performance Simplified event handling


Correct Answer: Reusability of animation states


Score: 4 / 20
Time Taken: 31s


Paste texts :-


StudyBuddy


User 


 Home


 To-Do List

 FlashCard

 Find Buddy

 Quiz

 Notifications

 Leader Board

Generate a Practice Test

Choose or upload materials to generate practice questions designed for you

Upload files **Paste texts** View Past Quizzes


Paste Your Notes


Paste text here


Generate


View Past Quizzes :-


StudyBuddy


hunny2004 


 Home


 To-Do List

 FlashCard

 Find Buddy

 Quiz




 Notifications

 Leader Board

Generate a Practice Test

Choose or upload materials to generate practice questions designed for you

Upload files Paste texts **View Past Quizzes**




Drag and drop notes, readings, lecture slides, etc.

Supported file types are .docx, .pdf, .pptx

Browse files

Generate

Your Quiz History

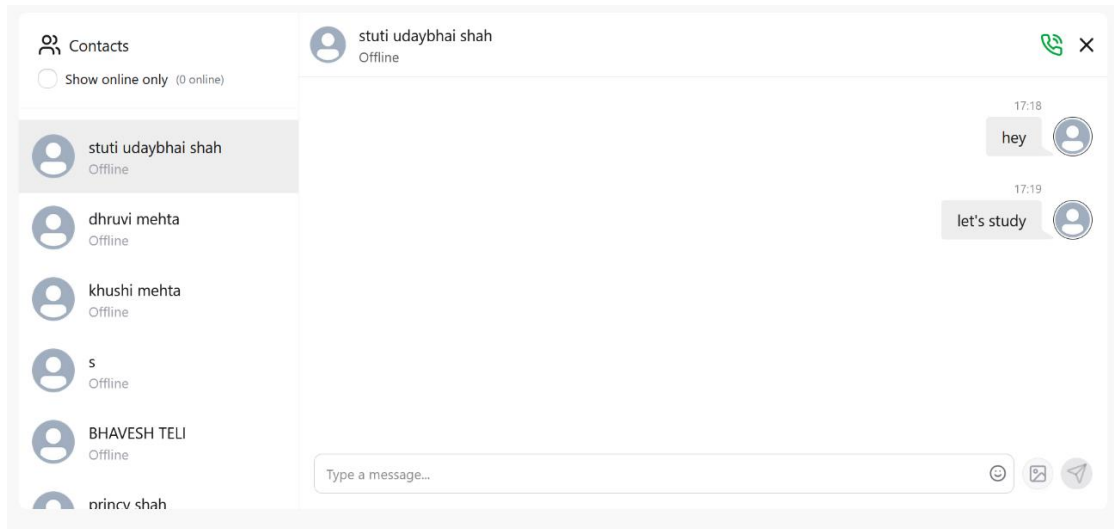
 **Total Score: 4 / 20**
Each correct answer earns 1 point.

Quiz 1

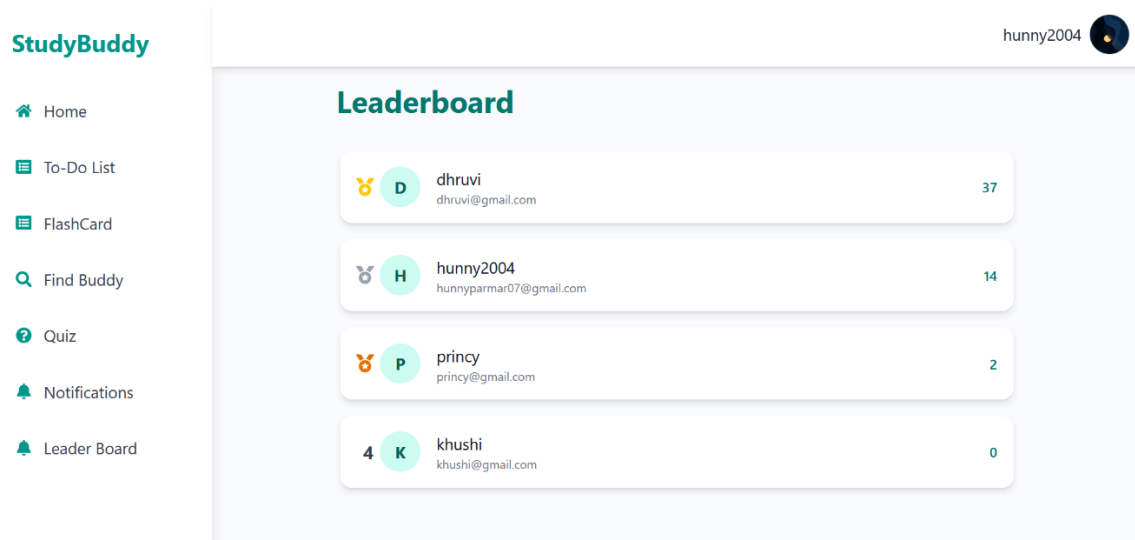
Score: **4 / 20** | Time taken: 0 min 31 sec

Detailed questions not available for this quiz.

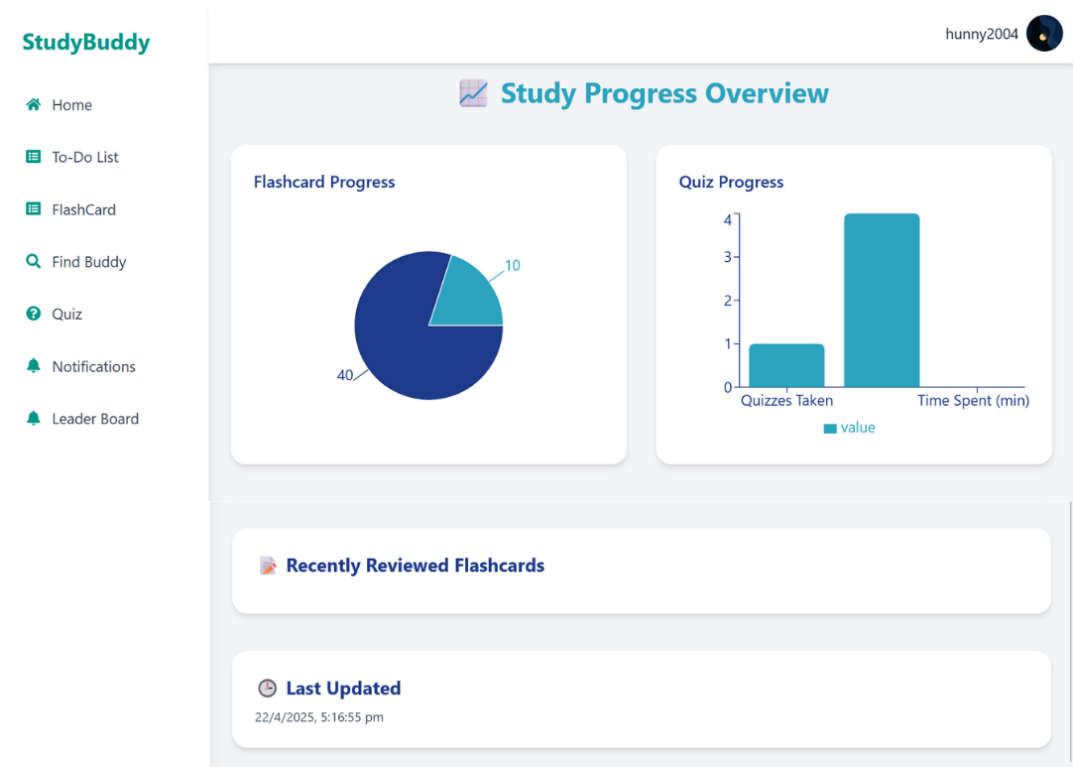
LiveChat :-



Leader Board :-



Progress Tracking :-





System Architecture

The StudyBuddy platform is developed using a modern full-stack architecture that integrates frontend and backend technologies with real-time communication, cloud-based storage, AI-based quiz generation, and secure authentication mechanisms.

The architecture follows a client-server model with well-defined responsibilities at each layer:

- **Frontend :-**

Technology Stack:

- React.js with Vite for fast development and optimized builds
- Zustand for lightweight and scalable state management
- Axios for handling HTTP requests to the backend APIs
- Socket.io-client for real-time chat and audio communication

Responsibilities:

- Renders the user interface and manages user interactions
- Manages client-side routing and state (e.g., user authentication, chat state)
- Sends requests to backend APIs for data processing (e.g., quiz creation, flashcard storage)
- Establishes WebSocket connections for live chat and voice call functionality
- Handles drag-and-drop UI for task management

- **Backend**

Technology Stack:

- Node.js with Express.js for RESTful API development
- Socket.io (server-side) to enable real-time bidirectional communication
- Gemini API integration for AI-based quiz generation from user-uploaded content
- JWT (jsonwebtoken) for secure, stateless authentication
- Multer for handling file uploads such as PDFs and profile pictures

Responsibilities:

- Handles business logic and routes (e.g., user registration, login, task updates)
- Processes uploaded files and extracts data for Gemini API
- Manages authentication and authorization
- Stores, retrieves, and manipulates data via MongoDB

- Coordinates chat events and real-time communication using Socket.io

- **Database**

Technology Stack:

- MongoDB as the NoSQL database
- Mongoose for schema modeling and interaction with MongoDB

Responsibilities:

- Stores user data, tasks, flashcards, quiz questions, chat messages, and performance statistics
- Provides indexed and efficient querying for fast data retrieval (e.g., smart matching)
- Supports object relationships using referencing (ref in Mongoose)

- **File Uploads & Storage**

Technology Stack:

- Multer for multipart/form-data parsing in Express

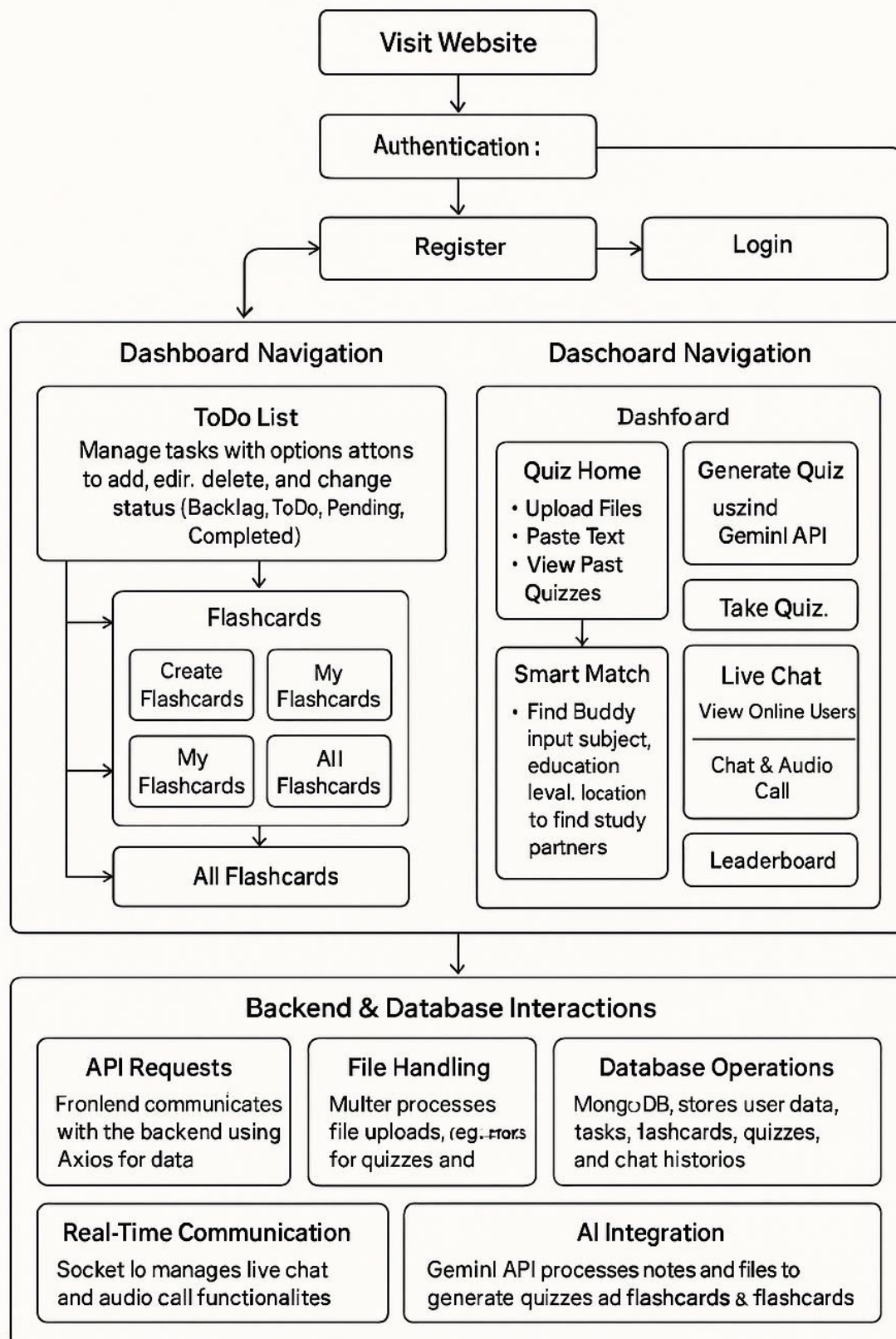
Responsibilities:

- Handles secure file uploads for:
- PDF documents used in quiz generation
- Profile pictures for user profiles
- Temporarily stores files for further processing (e.g., text extraction before sending to Gemini API)

- **System Integration Overview**

- The React frontend communicates with the Express backend using REST APIs (via Axios).
- The backend authenticates users using JWT and stores sessions securely.
- Quiz generation is triggered by uploading files, which are processed and sent to the Gemini API, then stored in MongoDB.
- Real-time features (chat and audio) are handled by Socket.io, using WebRTC and signaling channels.
- The MongoDB database stores all persistent data and supports fast querying for modules like Smart Matching and Leaderboard.

❖ System flowchart



Code

Methodology

❖ Code rules/guidelines/style

- We use React.js for building the frontend and Node.js for the backend logic.
- MongoDB is our primary database, used along with Mongoose for data modeling.
- Tailwind CSS is used for creating a clean, responsive UI design.
- All API calls are handled using Axios, with authentication tokens managed using localStorage and interceptors.
- Asynchronous operations across frontend and backend are wrapped in try-catch blocks for effective error handling.
- Zustand manages global application states such as user data, chat, and call information.
- Multer is used on the backend to handle file uploads like PDFs and profile images.
- Socket.io powers real-time chat and call functionalities between users.
- The Gemini API is used to generate quizzes and flashcards from user-uploaded files or text input.

Testing

Test ID	Module/Feature	Test Type	Description	Expected Outcome	Actual Outcome	Result	Tested By
TST01	User Registration/Login	Manual	Register new user and login with credentials	User is registered and redirected to dashboard	Works as expected	Pass	Team
TST02	Profile Setup	Manual	Upload profile pic and select preferences	Profile is saved and visible on dashboard	Profile updates correctly	Pass	Team
TST03	Flashcards (Manual)	Unit	Create, edit, and delete flashcards	Flashcards are saved, updated, and removed	All operations worked	Pass	Team
TST04	Flashcards (AI-Generated)	Integration	Upload PDF and generate flashcards with Gemini AI	Flashcards extracted correctly from PDF content	Text extracted well; minor spacing in a few cards	Pass	Team
TST05	Folder View	Manual	View all folders, search, and navigate to folder pages	Folders load and search filters results	Smooth folder transitions	Pass	Team
TST06	Quiz Generator (Text)	Unit	Paste text and generate quiz questions	Questions and MCQs are generated from text	Accurate and varied MCQs generated	Pass	Team
TST07	Quiz Generator (PDF)	Integration	Upload PDF and generate quiz	Quiz generated based on document content	Handled large files with good results	Pass	Team
TST08	Quiz Submission	Manual	Take quiz and submit to see score and time	Score and time are displayed with feedback	Results match expected answers	Pass	Team

TST09	Quiz History	Manual	View previous quizzes with full breakdown	Lists all attempts with correct/wrong answers	All data saved and shown clearly	Pass	Team
TST10	Study Chat	Integration	Live message between two users via Socket.io	Real-time chat with correct user info	Delivers messages instantly and reliably	Pass	Team
TST11	To-Do List	Manual	Create and move tasks across status boards	Tasks created and change status as dragged	Responsive drag-and-drop works well	Pass	Team
TST12	Country Filter (Search)	Unit	Filter study buddies by country using react-select	Correct list appears when country is chosen	Accurate results and dropdown UI is responsive	Pass	Team
TST13	Performance	Manual	Load time & responsiveness across components	App loads fast and transitions are smooth	Good performance even with large flashcard sets	Pass	Team
TST14	Accessibility	Manual	Keyboard nav, color contrast, ARIA roles	Meets basic accessibility standards	Minor improvements needed in button labels	Pass	Team
TST15	Security	Manual	Basic auth checks, input validation, session protection	Inputs validated, restricted access works properly	Secure endpoints with JWT and form validation	Pass	Team

Future Enhancement

❖ **Future Enhancement**

Dark Mode

Implement a user-selectable dark theme for better night-time usage and reduced eye strain.

Video Calling Support

Extend current audio call functionality to support video conferencing for more engaging study sessions.

Voice Notes in Chat

Allow users to send voice notes in the live chat module for faster communication.

Flashcard Sharing

Enable users to share their flashcards with friends or groups.

Study Groups Feature

Let users create or join group rooms for collaborative study with integrated group chat and shared to-dos.

Gamification

Introduce rewards, badges, and achievements for completing quizzes, flashcards, and daily tasks.

AI-based Smart Suggestions

AI-driven study buddy recommendations and flashcard generation based on learning history.

Multi-language Support

Allow users to switch between languages for global usability.

Admin Dashboard

Provide limited moderation tools for managing inappropriate content or fake profiles (if needed).

Bibliography & References

References are as follows:

React :-

<https://reactjs.org>

Nodejs :-

<https://nodejs.org>

MongoDB :-

<https://www.mongodb.com>

Tailwindcss :-

<https://tailwindcss.com>

<https://axios-http.com>

<https://zustand-demo.pmnd.rs>

<https://socket.io>

<https://quizlet.com>

<https://studystream.live>