

A PROJECT REPORT  
ON

‘CODE BUDDY’

BY

GAURPAD SHUKLA (21625)

ANSHIKA SHUKLA (21616)

DEEP ARYAN (21622)

SUBMITTED TO

PROF. SUMAN PANDEY

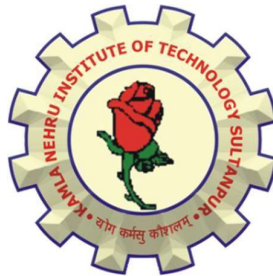
PROF. ANOO

A PROJECT REPORT SUBMITTED ON PARTIAL FULFILLMENT OF THE  
REQUIREMENT FOR THE AWARD OF DEGREE

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KAMLA NEHRU INSTITUTE OF TECHNOLOGY

SULTANPUR (U.P.) – 228118

(An Autonomous State Government Institute)

AFFILIATED TO

DR A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY LUCKNOW (U.P.) INDIA

2024-25

## CERTIFICATE

This is to certify that Gaupad Shukla (21625), Anshika Shukla (21616) and Deep Aryan (21622) have carried out the project work in this report entitled “Code Buddy” for the award of Bachelor of Technology in Computer Science & Engineering at Kamla Nehru Institute of Technology, affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow. This report is the record of the candidates’ own work carried out by them under our supervision and guidance. This project work is the part of their Bachelor of Technology in Computer Science and Engineering curriculum. Their performance was excellent, and we wish them good luck for their future endeavours.

.....  
Prof. Anoo  
Assistant Professor  
Supervisor

.....  
Prof Suman Pandey  
Professor  
Supervisor

.....  
Dr. Arvind Kumar Tiwari  
Professor  
HOD CSED

## DECLARATION

We hereby declare that this submission is our own work and that, to the rest of our knowledge and belief, it contains no material previously or written by another person nor material which to a substantial extent has been accepted for the award of any other degree of the university or other institutes of higher learning, except where due acknowledgement has been made in text.

GAURPAD SHUKLA (21625)

ANSHIKA SHUKLA (21616)

DEEP ARYAN (21622)

## ACKNOWLEDGEMENT

We wish to express our sincere gratitude towards Prof. Suman Pandey (Professor Incharge) & Prof. Anoo (Asst. Professor Incharge) for their valuable suggestions and guidance throughout our work at Kamla Nehru Institute of Technology, Sultanpur. They have guided us through the difficulties and made us understand the concepts needed for the project work. Their experimental and theoretical knowledge has been very helpful. We feel privileged in expressing our gratitude to all faculty members of the Computer Science and Engineering department for their encouragement and moral support. We would like to thank Dr. Arvind Kumar Tiwari, Head of the Department, Computer Science & Engineering for providing us a golden opportunity to work with him. The support and then environment provided to us during our project were more than what anyone would have expected. We are also deeply indebted to all those without whose firm support, encouragement and guidance, this project would not have seen this stage.

## ABSTRACT

In the evolving landscape of software development, collaborative coding and continuous learning have become fundamental. Our final year project, ‘Code Buddy’, addresses this need by introducing a versatile web application designed to facilitate real-time collaborative coding, code sharing, and remaining up to date with various coding contests. This project aims to create a unified platform that not only enhances the coding experience for developers but also serves as a learning hub for programming enthusiasts and students. The core of project is a real-time code editor enabling multiple users to write and edit code simultaneously, akin to a Google Docs experience for programmers. This feature is augmented by supporting popular programming languages with syntax highlighting, error detection, and an integrated debugger. A standout feature is its integration with real-time coding platforms to sync their contests information, enabling effective time management. Security, user privacy, and data integrity are paramount in our design. It implements robust security measures to ensure the safety of user data and code. The application is also designed with a responsive interface, ensuring accessibility across various devices and catering to a broad user base.

## **TABLE OF CONTENTS**

<b>CERTIFICATE .....</b>	<b>ii</b>
<b>DECLARATION.....</b>	<b>iii</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>iv</b>
<b>ABSTRACT.....</b>	<b>v</b>
<b>TABLE OF FIGURES.....</b>	<b>viii</b>
<b>1. INTRODUCTION.....</b>	<b>1-5</b>
1.1 OVERVIEW OF THE PROJECT.....	2
1.2 MOTIVATION.....	3
1.3 OBJECTIVE.....	4
1.4 FEATURES.....	5
<b>2. LITERATURE SURVEY.....</b>	<b>6-7</b>
2.1 BACKGROUND INFORMATION.....	6
2.2 COMPARISON BETWEEN EXISTING SYSTEMS.....	7
<b>3. SYSTEM ANALYSIS.....</b>	<b>8-9</b>
3.1 EXISTING SYSTEM.....	8
3.2 PROPOSED SYSTEM.....	8
<b>4. DESIGN AND IMPLEMENTATION.....</b>	<b>10-27</b>
4.1 SYSTEM REQUIREMENTS.....	10
4.2 PROCEDURE.....	11
4.3 PROJECT MODULES.....	11
4.4 DATA FLOW DIAGRAM.....	13
4.5 SEQUENCE DIAGRAM.....	15
4.6 USE CASE DIAGRAM.....	17
4.7 UI INTERFACES.....	20
<b>5. TECHNOLOGIES USED.....</b>	<b>28-33</b>
5.1 LANGUAGE AND LIBRARIES USED.....	28
5.1.1 JAVASCRIPT .....	28
5.1.2 REACT.....	29
5.1.3 NODE/EXPRESS .....	30
5.1.4 MONGODB .....	30
5.1.5 FIREBASE .....	31
5.1.6 TAILWIND CSS .....	31
5.2 TOOLS AND PLATFORMS USED	
5.2.1 VISUAL STUDIO CODE.....	32
5.2.2 POSTMAN.....	33
5.2.3 GIT VERSION CONTROL.....	33

<b>6. TESTING AND VALIDATION.....</b>	<b>33</b>
6.1 API TESTING WITH POSTMAN.....	33
6.2 CROSS BROWSER COMPATIBILITY TESTING.....	33
6.3 DIVERSE OS TESTING.....	33
6.4 NPM PACKAGE VALIDATION.....	33
6.5 NETWORK ENVIRONMENT TESTING.....	33
6.6 MOBILE RESPONSIVENESS TESTING.....	33
 <b>CONCLUSION.....</b>	 <b>34</b>
<b>FUTURE SCOPE.....</b>	<b>35</b>
<b>PERT/CPM CHART.....</b>	<b>36</b>
<b>REFERENCES.....</b>	<b>37-38</b>

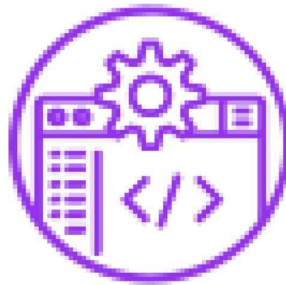
## LIST OF FIGURES

Fig 1.1 Code Buddy Logo.....	1
Fig 2.1 Comparison Between Existing Platforms.....	7
Fig 4.1 Data Flow Diagram.....	13
Fig 4.2 Sequence Diagram.....	15
Fig 4.3 Use Case Diagram.....	17
Fig 4.1 Login Page.....	20
Fig 4.2 Register Page.....	20
Fig 4.3 Code Editor to Compile and Run Code.....	21
Fig 4.4 List of Problems.....	21
Fig 4.5 Watch Out Upcoming Contests.....	22
Fig 4.6 Add Contests to Calendar.....	22
Fig 4.8 Save Code Snippets.....	23
Fig 4.9 Settings Page to Update Profile.....	24
Fig 4.10 Settings Page to Update Password.....	24
Fig 4.11 Feedback Section.....	25
Fig 4.12 Create Code Room.....	25
Fig 4.13 Code Room.....	26
Fig 4.14 Code Room With Two Simultaneous Users.....	26
Fig 4.15 Create Communication Room.....	27
Fig 4.16 Communication Room.....	27
Fig 5 PERT/CPM Chart.....	36



## INTRODUCTION

In today's changing software development landscape, effective cooperation has never been more important. The growing complexity of technology, combined with the need for a diverse set of abilities, highlights the importance of collaborative coding platforms for today's engineers. Enter 'Code Buddy,' an innovative web tool that expands the usual bounds of collaborative coding. It provides a comprehensive solution by smoothly merging real-time collaborative coding, extensive coding challenges, community-driven learning initiatives, and a flexible communication channel. Our final year project, 'Code Buddy,' addresses this requirement by developing a multifunctional web tool that allows for real-time collaborative coding, code sharing, and keeping up with various coding challenges. The goal of this project is to establish a single platform that improves developers' coding experiences while also serving as a learning hub for programming enthusiasts and students.



CODE BUDDY

Fig 1.1: Code Buddy Logo

## 1.1 Overview of Project

Code Buddy is an innovative web tool that improves the collaborative coding experience for developers, programming enthusiasts, and students. The platform's goal is to enable real-time collaborative coding, code sharing, and staying up to speed on coding contests, effectively functioning as a unified hub for coding and learning. The fundamental feature of Code Buddy is a real-time code editor that allows several users to write and update code at the same time, creating a collaborative environment comparable to Google Docs but specialized for coding. This tool supports a variety of common programming languages and includes syntax highlighting, error detection, and an integrated debugger to help you code more efficiently.

In terms of security and data privacy, 'Code Buddy' prioritizes the protection of user data as well as code integrity. To protect sensitive information, stringent security measures are put in place, resulting in a safe and trustworthy collaborative atmosphere. Furthermore, the program has a user-friendly interface and a responsive design, ensuring accessibility across several platforms and appealing to a varied user demographic.

Additionally, Code Buddy connects with multiple coding platforms to sync information regarding coding contests, allowing users to better organize their time. The program provides strong security, user privacy, and data integrity, making it a trustworthy platform for developers. Other modules of Code Buddy include an AI ChatBot for code-related queries, a Contest Watcher to keep track of forthcoming coding contests, an area for viewing and organizing code snippets, and a communication channel for audio, video, and text interactions. Code Buddy's broad set of features makes it a versatile tool for coding, learning, and collaboration.

## 1.2 Motivation

The inspiration for 'Code Buddy' derives from the awareness of numerous fundamental aspects influencing the present landscape of software development. To begin, the rising complexity of technology and the variety of skill sets necessary in today's programming domain highlight the importance of good teamwork and continual learning. Furthermore, the growth of coding contests and competitive programming platforms emphasizes the need for easily available tools that allow programmers to engage, learn, and excel in such settings.

Furthermore, traditional techniques of coding collaboration frequently lack the real-time capabilities and integrated features required for smooth teamwork and knowledge transfer. Recognizing these problems, 'Code Buddy' seeks to bridge the gap by offering a single platform that not only improves collaborative coding experiences but also serves as a comprehensive learning hub for programmers of all skill levels.

Additionally, the project is driven by the aim to create a lively and inclusive programming community in which people can work, learn, and grow together. By providing a platform that stimulates interaction, information exchange, and participation in coding contests, 'Code Buddy' aims to empower programmers to reach their full potential and contribute meaningfully to the ever-changing area of software development.

The inspiration for 'Code Buddy' originates from its mission to democratize access to collaborative coding tools, empower individuals to learn and excel in programming, and foster a healthy community of passionate developers dedicated to continual growth and innovation.

### 1.3 Objective

The objective of our project, 'Code Buddy,' is to revolutionize the way programmers collaborate, learn, and engage with coding contests. Our key goals include:

1. Real-Time Collaborative Coding: Create a powerful platform that allows numerous users to work on coding projects in real time, promoting seamless teamwork and knowledge exchange.
2. Enhance Learning Opportunities: Create a complete learning environment that includes coding problems, tools, and coding contests, allowing users to enhance their programming skills and broaden their knowledge.
3. Support Multiple Programming Languages: Ensure compatibility with a wide range of programming languages to accommodate diverse user preferences and encourage involvement from programmers with varying skill sets.
4. Integrate contest information: Integrate with real-time coding platforms to synchronize contest information, allowing participants to keep informed about impending challenges, manage their time effectively, and participate in coding contests effortlessly.
5. Ensure security and data privacy: Implement strong security measures to protect user data, code integrity, and privacy, hence creating a safe and trustworthy environment for collaboration and learning.
6. Foster Community Engagement: Create a thriving community of programmers by offering forums, chat rooms, and collaborative coding sessions where users can communicate, share ideas, and help each other with their coding projects.
7. Promote Accessibility and User-Friendliness: Create a user-friendly interface and responsive design for the platform to enable accessibility across all devices and cater to a varied user base, including both novices and expert programmers.

## 1.4 Features

1. Real-Time Collaborative Coding: Allow several users to collaborate and amend the same code in real time.
2. Integrated Code Editor: Offer a powerful code editor with syntax highlighting, error detection, and debugging features.
3. Support for Multiple Programming Languages: A variety of programming languages are supported to satisfy different user preferences and project requirements.
4. Code Sharing and Version Control: Allow users to share their code with others and incorporate version control systems such as Git to improve code management.
5. Seamless Contest Integration: Integrate with real-time coding platforms to synchronize contest information, allowing users to stay informed about impending challenges and manage their time more effectively.
6. Secure Environment: Implement strong security mechanisms to protect user data, code integrity, and privacy, resulting in a safe and trusted collaboration environment.
7. Community Interaction: Promote community participation by hosting forums, chat rooms, and collaborative coding sessions where users may communicate, share ideas, and encourage one another.
8. User-Friendly Interface: Create a user-friendly and responsive interface for the platform to ensure accessibility across multiple devices and cater to users of all skill levels.
9. Communication Channel: Allow users to successfully communicate within the platform using video/audio/text channels, promoting seamless interaction and cooperation during coding sessions.

## LITERATURE SURVEY

The literature survey explores the evolution of collaborative coding platforms, from basic text editors to advanced real-time, cloud-based development environments. By analysing existing platforms, it becomes evident that technological advancements have significantly improved the efficiency, usability, and collaborative capabilities for developers. Comparative analysis of these platforms reveals differences in features, user experience, and scalability, indicating opportunities for further innovation to enhance the coding and learning experience.

### 1. BACKGROUND INFORMATION

The development of collaborative coding platforms has been shaped by technological progress, changing developer needs as students participating in team coding competitions like ACM-ICPC and Google Hashcode which requires coding in a collaborative way, and a growing emphasis on remote collaboration. The following key stages highlight the evolution of these platforms, supported by relevant references:

#### 1.1 Basic Text Editors

- Early collaborative coding involved using simple text editors and sharing files manually.
- Limitations included lack of real-time collaboration and difficulties in version control.

#### 1.2 Integrated Development Environments (IDEs)

- IDEs like Eclipse and Visual Studio introduced more comprehensive coding tools.
- Provided features such as syntax highlighting, debugging, and version control integration.
- Reference: Johnson & Foote (1988) - "Designing Reusable Classes"

#### 1.3 Online Coding Platforms

- Platforms like JSFiddle and CodePen enabled online code sharing and collaboration.
- Allowed developers to test code snippets in real-time without local setup.
- Reference: Rao & Davison (2011) - DOI: 10.1145/2046684.2046692

#### 1.4 Real-Time Collaborative Editors

- Google Docs-style editors for code, such as Cloud9 and Code anywhere, offered real-time collaboration.
- Enabled multiple users to edit code simultaneously with instant updates.
- Reference: Smith et al. (2013) - "Real-Time Collaborative Coding."

#### 1.5 Cloud-Based Development Environments

- Platforms like GitHub Codespaces and Repl.it provided full-featured development environments in the cloud.
- Supported features like live collaboration, containerized environments, and integration with CI/CD pipelines.
- Reference: Lee et al. (2020) - DOI: 10.1145/3397481.3397485

#### 1.6 Integration of AI and Machine Learning

- Modern platforms integrate AI to offer smart code completions, error detection, and personalized learning resources.
- Reference: Brown et al. (2020) - "Language Models are Few-Shot Learners" -

DOI: 10.18653/v1/2020.acl-main.422

## 2. COMPARISON BETWEEN EXISTING SYSTEMS

This comparison highlights the strengths and weaknesses of existing collaborative coding platforms, underscoring the potential areas for improvement in terms of performance, security, and feature sets to better support developers' needs.

Platform	Real-Time Collaboration	Supported Languages	Customization Options	Performance (Lag/Downtime)	Security & Privacy	Additional Features
JSFiddle	No	HTML, CSS, JavaScript	Limited	High Performance	Basic	Code sharing via URLs
CodePen	No	HTML, CSS, JavaScript	Moderate	High Performance	Basic	Embed pens, code snippets
Repl.it	Yes	Multiple	High	Moderate Performance	Enhanced	Classroom mode, auto-complete
Cloud9 (AWS)	Yes	Multiple	High	Moderate Performance	High	Integrated terminal, debugger
GitHub Codespaces	Yes	Multiple	High	High Performance	High	Integrated with GitHub
Visual Studio Live Share	Yes	Multiple	High	High Performance	High	Debugging, sharing sessions
Codeanywhere	Yes	Multiple	Moderate	Moderate Performance	Enhanced	Containerized environments
Google Colab	Yes	Python	Moderate	High Performance	High	GPU/TPU support, notebooks

**Fig 2.1:** Comparison Between Existing Platforms

## SYSTEM ANALYSIS

### 3.1 Existing Model:

The current landscape of collaborative coding platforms includes a wide range of tools and platforms, each with its own set of features and functionalities to facilitate collaborative coding and learning. These platforms include online code editors with real-time collaboration features, as well as coding contest and instructional tools.

1. Real-time collaboration tools:

Many existing platforms support real-time collaboration, which allows multiple users to change code at the same time. Examples include Google Docs for Code, CodePen, Collabedit, and CoderPad. These platforms often provide basic code editing features like syntax highlighting and version history monitoring.

2. Coding Competition Platforms:

Dedicated platforms like Codeforces, TopCoder, and HackerRank organize coding contests and challenges for programmers of all skill levels. These platforms provide a diverse set of computational and problem-solving tasks, encouraging competition and skill improvement among members.

3. Learning Resources:

Some platforms focus on providing learning resources and tutorials alongside collaborative coding features. Examples include platforms like freeCodeCamp, LeetCode, and GeeksforGeeks, which offer tutorials, articles, and coding exercises to help programmers enhance their skills and knowledge.

### 3.2 Proposed Model:

Code Buddy seeks to overcome the drawbacks of current collaborative coding platforms by offering a holistic solution that smoothly blends real-time collaboration, coding challenges, and community-driven learning initiatives.

1. Real-Time Collaborative Coding Environment:

'Code Buddy' has a powerful real-time code editor that allows many people to edit code simultaneously from different locations. The platform supports many programming languages and has advanced features like syntax highlighting, code completion, and error detection.

2. Integrated Coding Contests Organisation:

The platform make you organize coding contests and challenges on a regular basis, allowing users to put their talents to the test, compete against others, and earn recognition. Contest specifics, such as start times, durations, and issue statements, are easily integrated into the platform to promote participation and engagement.

3. Learning Hub:



'Code Buddy' serves as a complete learning hub for programmers, with a vast collection of coding exercises. Users can access problems that are matched to their skill levels and interests, allowing them to constantly develop their coding skills and extend their knowledge.

4. Community Engagement:

'Code Buddy' develops a thriving programming community by fostering cooperation, knowledge exchange, and mentoring. Users can interact with other programmers, engage in debates, and work on projects, resulting in a supportive and dynamic learning community.

Overall, the proposed solution intends to transform the collaborative coding experience by allowing programmers to cooperate efficiently, compete in coding competitions, and constantly improve their abilities in a safe and friendly environment.

## DESIGN AND IMPLEMENTATION

### 4.1 System Requirements

#### 1. Hardware Requirements

- Modern multi-core processor with sufficient processing power to handle concurrent user interactions and real-time collaboration.
- Minimum 8 GB of RAM recommended for smooth operation.
- Minimum 256 GB of storage space to store application files, databases, and user data.
- Minimum internet connection with at least 10 Mbps download and upload speeds recommended.
- High-resolution display with adequate screen size for viewing code editor, chat interface, and video/audio communication.
- Keyboard and mouse or touchpad for interacting with the application, writing code, and navigating user interfaces.
- Support for various input/output ports (USB, HDMI, DisplayPort, etc.) for connecting peripherals and external devices.

#### 2. Software Requirements

- Operating System Compatible with Windows, macOS, Linux, or any other modern operating system.
- Latest versions of popular web browsers such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge.
- Node.js runtime environment to execute JavaScript code on the server side. Version 14.x or later recommended.
- NoSQL database system for storing user data, chat logs, contest details, and other application data.
- Any text editor or integrated development environment (IDE) for writing and editing code.
- Version control system for tracking changes to the codebase and collaborating with other developers.
- Extensions for Visual Studio Code to enhance productivity and streamline development workflows.
- If utilizing the Google Sign-In feature, users will need a Google account to authenticate and sign in to the application.

## 4.2 Procedure

### 1. Defined Folder Structure

- Begin by organizing the project folder structure, separating the frontend and backend components for clarity and modularity.

### 2. Backend Development

- Utilize Node.js and Express.js to set up the backend server and define the RESTful APIs.
- Implement routes, controllers, and models following the MVC (Model-View-Controller) architecture to maintain code organization and separation of concerns.
- Integrate MongoDB as the database to store and manage application data, ensuring efficient data storage and retrieval.

### 3. Frontend Development

- Initiate frontend development using React.js library to create the user interface and application views.
- Design and implement components, pages, and navigation flows to provide a seamless user experience.
- Utilize React Router for client-side routing and navigation between different views and components.
- Implement Axios or Fetch API to make asynchronous API calls to the backend server for fetching and updating data.

### 4. Integrate Backend with Frontend

- Establish communication between the frontend and backend by consuming the RESTful APIs exposed by the backend server.
- Configure CORS (Cross-Origin Resource Sharing) settings to allow the frontend to securely communicate with the backend server.

## 4.3 Project Modules

Our project, 'Code Buddy,' consists of multiple interrelated modules designed to encourage collaborative coding, learning, and engagement in the programming community. The modules include:

1. Code Room:

This module offers a collaborative environment where users can write and edit code in real time. Allowing several people to work on the same code at the same time promotes teamwork and increases productivity.

2. IDE:

The Integrated Development Environment (IDE) module provides a feature-rich environment for developing, testing, and debugging code. It contains crucial tools and functions for streamlining the coding process and increasing efficiency.

3. Code Snippets:

This module allows users to store and manage code snippets, resulting in a centralized repository that is easy to access and reference. It encourages code reuse and knowledge sharing among users.

4. Contest Watcher:

This module allows users to stay up to speed on forthcoming coding contests across several platforms. Users can read contest details, create reminders, and engage in contests directly on the platform, encouraging a competitive yet collaborative attitude among users.

5. Problems:

The difficulties module provides a carefully curated set of coding difficulties taken from reliable platforms such as Codeforces. It allows users to practice and improve their coding skills by posing hard and different issues.

6. Communication Channel:

This module facilitates real-time communication among users through audio, video, and text channels. Users can create communication channels within code rooms to discuss code, ask questions, and collaborate effectively.

## 4.4 Data Flow Diagram

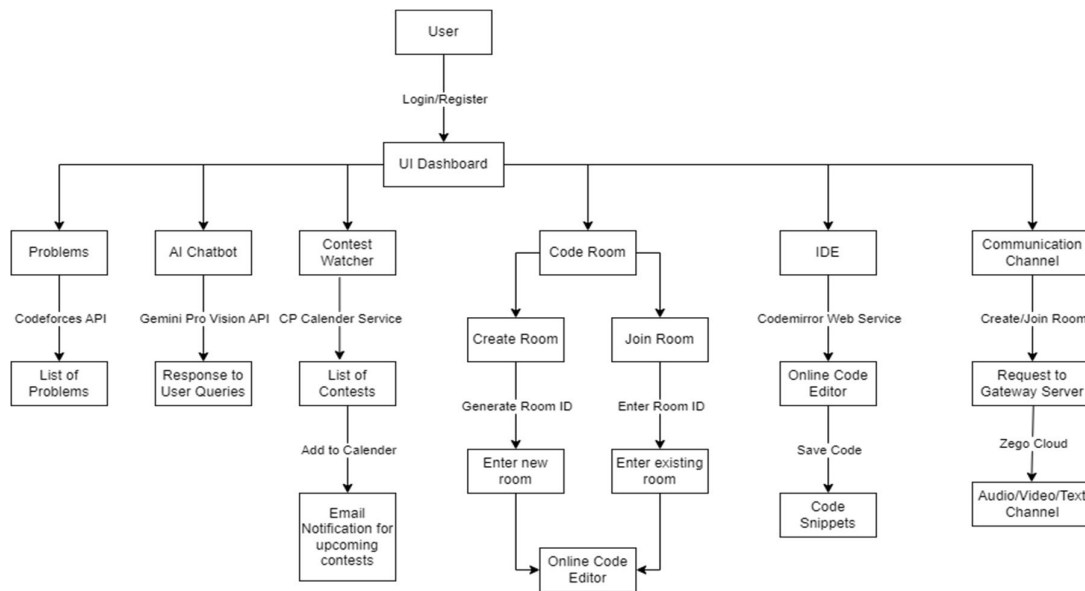


Fig 4.1: Data Flow Diagram

This diagram illustrates the flow of data between users, the UI dashboard, various modules, and external APIs/services, highlighting how the app facilitates collaborative coding and learning.

User Interaction:

- Users log in/register and access the UI Dashboard.

UI Dashboard:

- Problems: Fetches a list of coding problems from the Codeforces API.
- AI Chatbot: Uses the Gemini Pro Vision API to respond to coding queries.
- Contest Watcher: Retrieves contest details from the CP Calendar Service, allowing users to add contests to their calendars and receive email notifications.
- Code Room: Users can create/join collaborative coding rooms.
- IDE: Provides a real-time code editor via the CodeMirror Web Service and allows saving code snippets.
- Communication Channel: Facilitates audio, video, or text communication using the Zego Cloud service.

Problems Module:

- Displays problems fetched from the Codeforces API.

Contest Watcher Module:

- Lists upcoming contests and integrates with calendars for notifications.

Code Room Module:

- Allows creation/joining of code rooms for collaborative coding.

IDE Module:

- Online code editor for writing and testing code, with save functionality.

Communication Channel Module:

- Enables real-time communication among users in code rooms through Zego Cloud.

## 4.5 Sequence Diagram

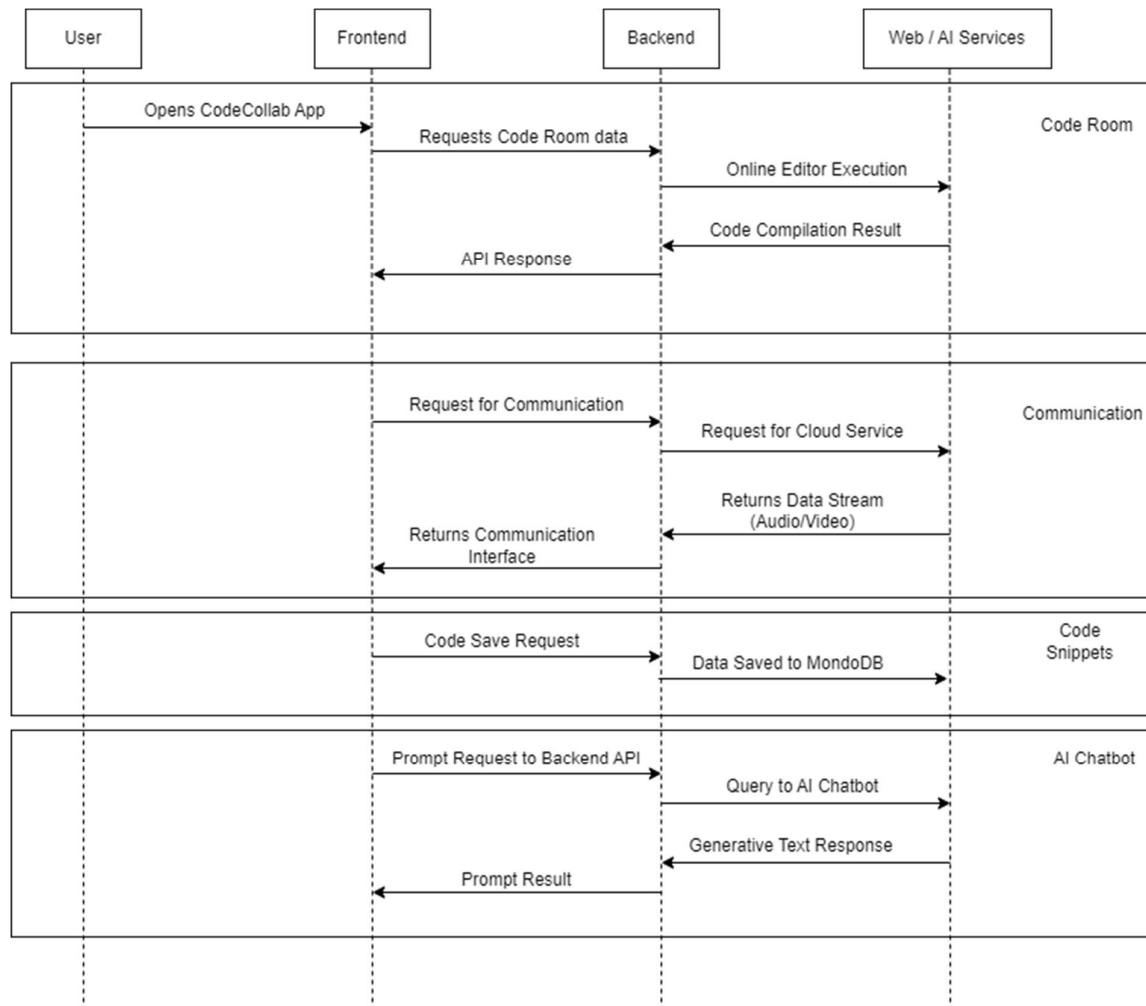


Fig 4.2: Sequence Diagram

This sequence diagram depicts the interactions between different components of the Code Buddy application: the User, Frontend, Backend, and Web/AI Services. The diagram is divided into four main interactions: Code Room, Communication, Code Snippets, and AI Chatbot.

### 1. Code Room

- **User Interaction:** The user opens the Code Buddy app and requests access to the code room data.
- **Frontend:** The frontend makes an API request to the backend for the code room data.

- Backend: The backend processes this request and performs the necessary operations for the online editor execution. Once the operations are completed, the backend compiles the code and returns the result to the frontend.
- Frontend: Receives the API response and displays the code room data to the user.

## 2. Communication

- User Interaction: The user requests to initiate communication.
- Frontend: The frontend forwards this request to the backend.
- Backend: The backend processes the request and contacts the cloud service to initiate the communication interface.
- Web/AI Services: The cloud service returns the data stream (audio/video) required for communication.
- Backend: The backend forwards the communication interface data to the frontend.
- Frontend: Receives and displays the communication interface to the user, enabling audio/video/text communication.

## 3. Code Snippets

- User Interaction: The user makes a request to save code snippets.
- Frontend: The frontend sends a code save request to the backend.
- Backend: The backend processes this request and saves the data to MongoDB.
- Backend: Confirms the save operation back to the frontend. □ Frontend: Confirms the operation to the user.



## 4.6 Use Case Diagram

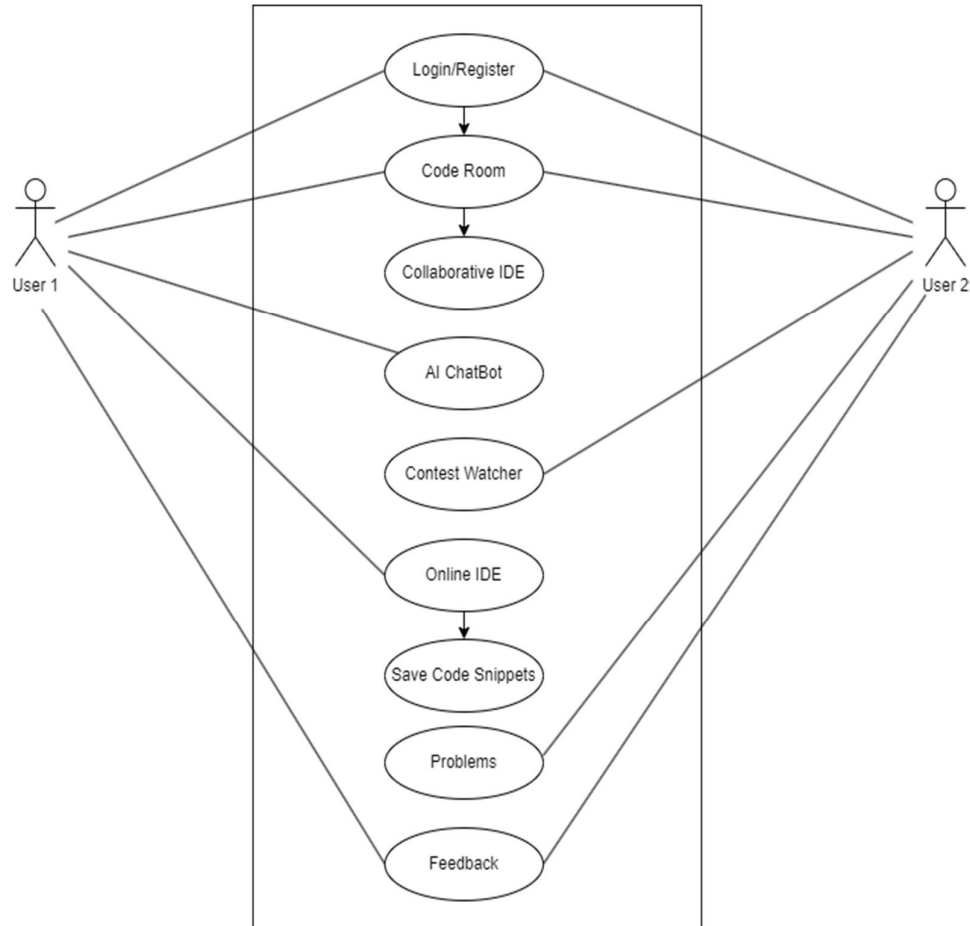


Fig 4.3: Use Case Diagram

This Use Case Diagram illustrates the interactions between the user and various modules of the Code Buddy application. Here's a detailed explanation of each use case and its interactions:

### 1. Login/Register:

- Description: Allows users to create a new account or log into an existing one.
- Actors: User
- Flow:
  - o The user accesses the Login/Register page.
  - o If registering, the user provides necessary details like username, email, and password.
  - o If logging in, the user provides their credentials and gains access to the system upon successful authentication.

## 2. Code Room:

- Description: The central hub where users can collaborate on coding projects.
- Actors: User □ Flow: o After logging in, the user enters the Code Room.
  - o Users can either create a new room or join an existing room to collaborate in real-time.

## 3. Collaborative IDE:

- Description: An integrated development environment that allows multiple users to write and edit code simultaneously. □ Actors: User
- Flow:
  - o Users in the Code Room utilize the Collaborative IDE to work on Code Buddyoratively.
  - o The IDE supports real-time updates, syntax highlighting, and debugging.

## 4. Contest Watcher:

- Description: Provides information about upcoming coding contests and allows users to add them to their calendars. □ Actors: User
- Flow: o Users can view a list of upcoming coding contests. o Users can add contests to their personal calendars and receive notifications.

## 5. Online IDE:

- Description: A stand-alone integrated development environment for writing, testing, and debugging code. □ Actors: User
- Flow:
  - o Users can access the Online IDE to work on their code independently of the collaborative environment.

## 6. Save Code Snippets:

- Description: Allows users to save and manage code snippets.
- Actors: User
- Flow: o Users can save snippets from their sessions in the Online IDE. o Saved snippets can be managed and retrieved for future use.

## 7. Problems:

- Description: Displays a list of coding problems from platforms like Codeforces. □ Actors: User
- Flow: o Users can browse a list of coding problems. o Users can select and work on problems to improve their coding skills.

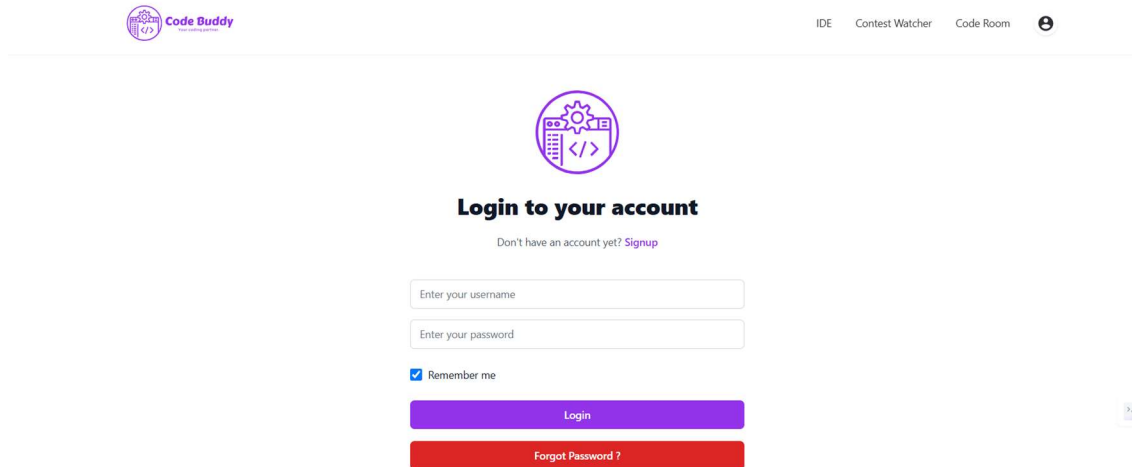
#### 8. Feedback:

- Description: Collects user feedback about the application. □  
Actors: User
- Flow: o Users can submit feedback about their experience using the application. o Feedback is stored and used for further improvements.

## 4.7 UI Interfaces

### 1. Login / Register Page:

The login/register page enables users to securely access the platform, facilitating account creation or authentication for personalized interaction with the system.

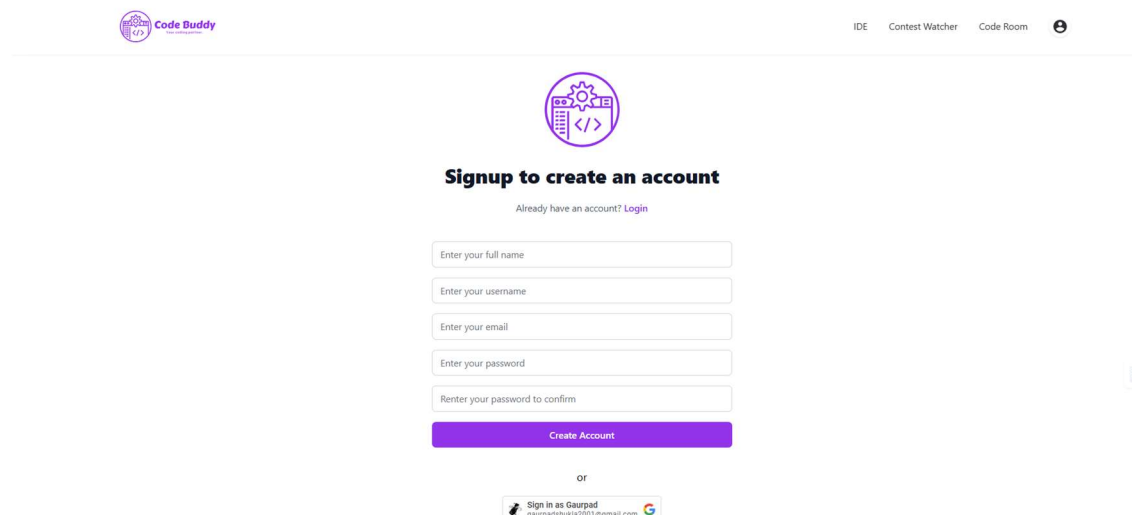


The screenshot shows the Code Buddy login interface. At the top left is the Code Buddy logo, and at the top right are links for IDE, Contest Watcher, Code Room, and a user profile icon. The main heading is "Login to your account" with a sub-link "Don't have an account yet? Signup". Below this are input fields for "Enter your username" and "Enter your password", a checked "Remember me" checkbox, a blue "Login" button, and a red "Forgot Password?" button. A small "X-5" icon is visible on the right side of the form.

Fig 4.1: Login Page

### 2. Online Code Editor

The IDE (Integrated Development Environment) offers users a comprehensive platform to write, edit, and debug code efficiently. It provides tools like syntax highlighting, auto-completion, and debugging features for seamless coding.



The screenshot shows the Code Buddy signup interface. At the top left is the Code Buddy logo, and at the top right are links for IDE, Contest Watcher, Code Room, and a user profile icon. The main heading is "Signup to create an account" with a sub-link "Already have an account? Login". Below this are input fields for "Enter your full name", "Enter your username", "Enter your email", "Enter your password", and "Reenter your password to confirm". A blue "Create Account" button is at the bottom. Below the button is a link "or" and a "Sign in as Gaupad" button with a Google logo. A small "X-5" icon is visible on the right side of the form.

Fig 4.2: Register Page

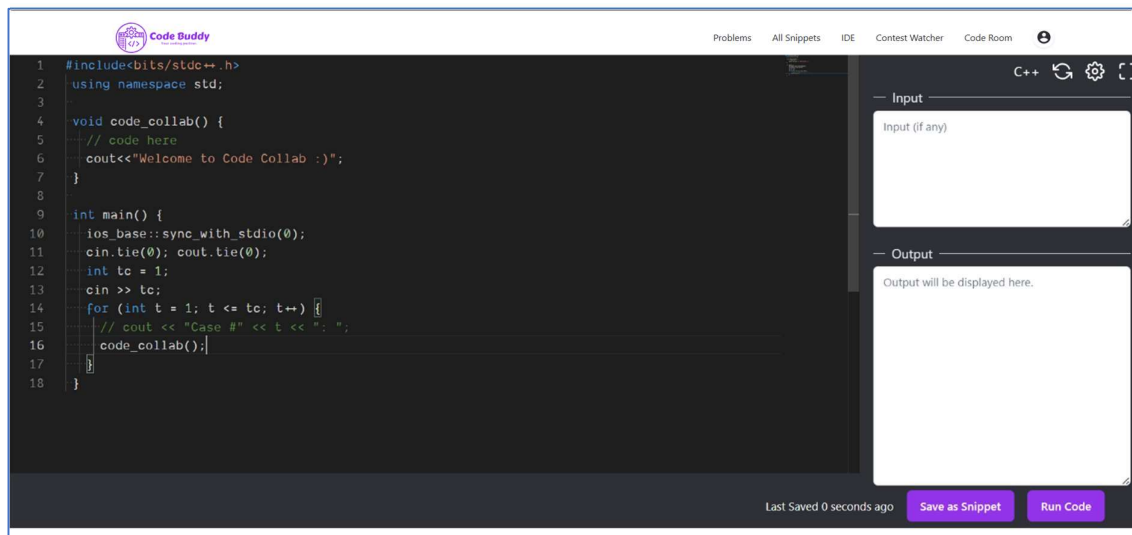


Fig 4.3: Code Editor to compile and run code

### 3. Problems Page

The Problems Page contains latest problems from Codeforces platform on which users can practice problems.

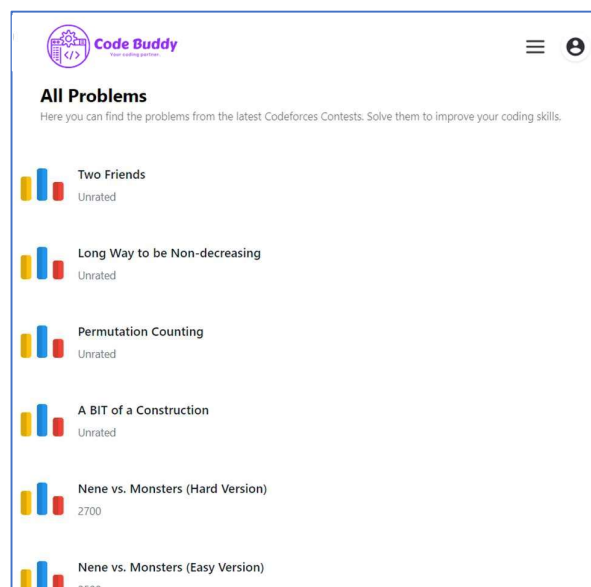


Fig 4.4: List of Problems

### 4. Contests Watcher:

From the contests watcher page, users can view all the upcoming contests on various coding platforms, and can add it on their calendars to receive contest updates on mail.

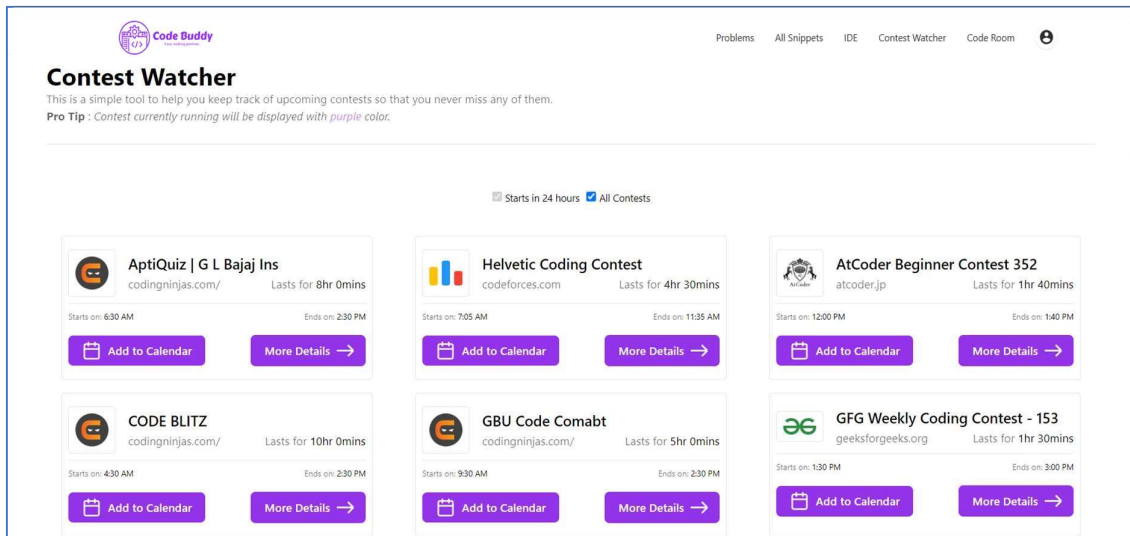


Fig 4.5: Watch out upcoming contests

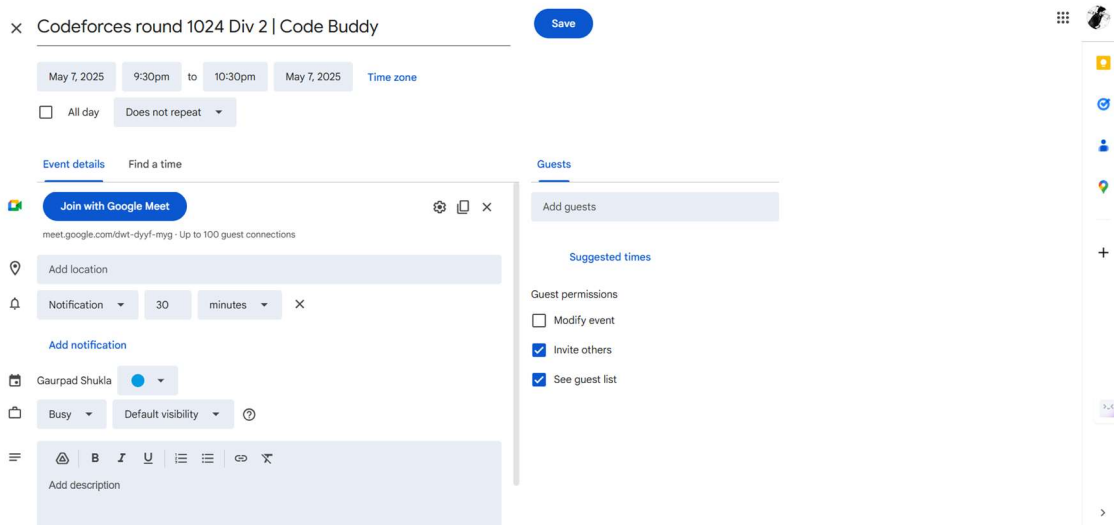


Fig 4.6: Add Contests to Calendar

## 5. All Snippets

The "All Snippets" page allows users to access and manage their saved code snippets, facilitating organization and quick reference to previously written code segments for enhanced productivity. The code can also be shared.

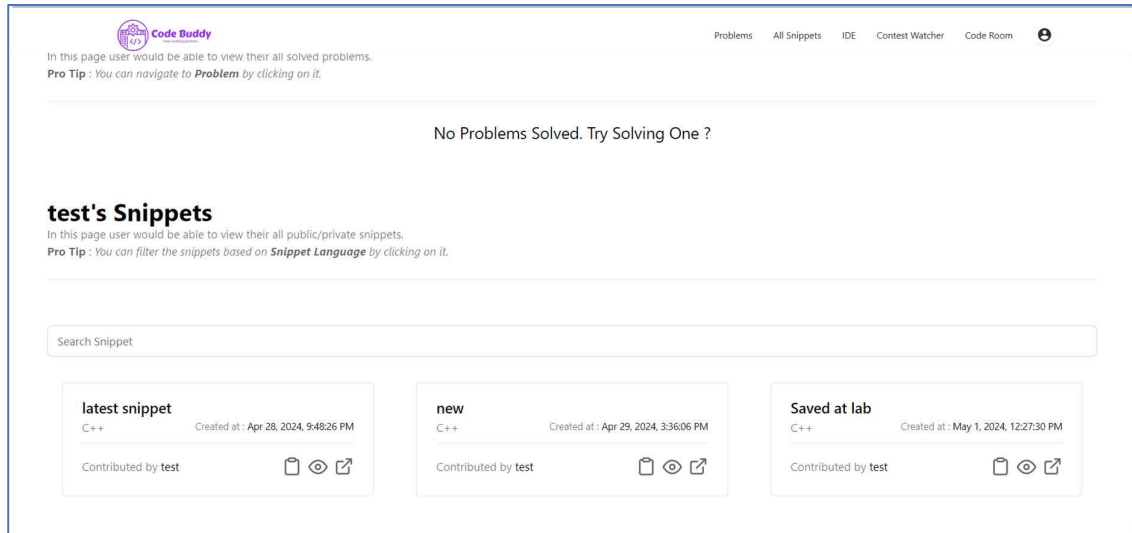


Fig 4.7: Saved Code Snippets

## 6. Settings Page

The settings page empowers users to personalize their account by adjusting account details like username and name, as well as enhancing security through password management for a tailored and secure experience.

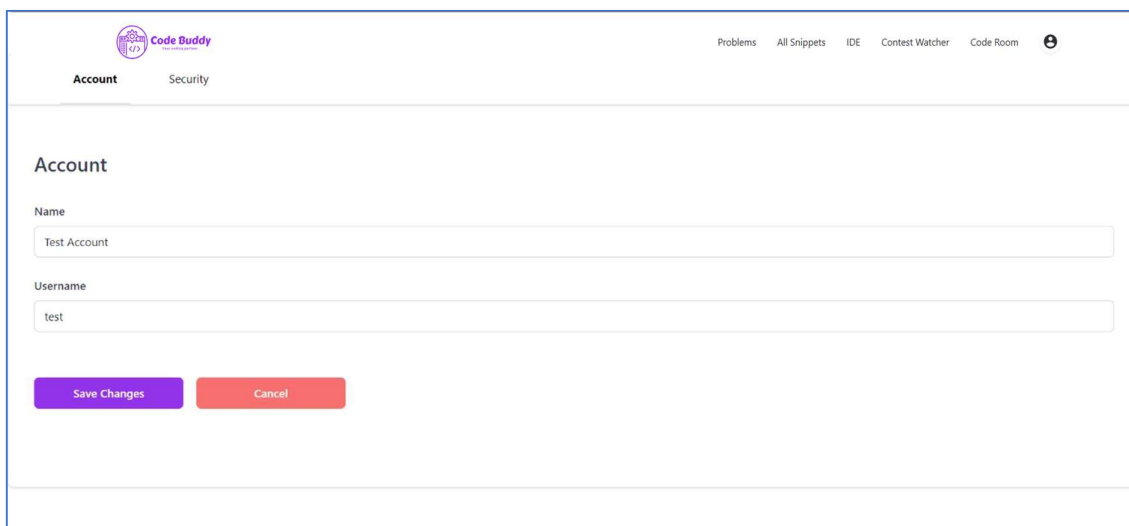
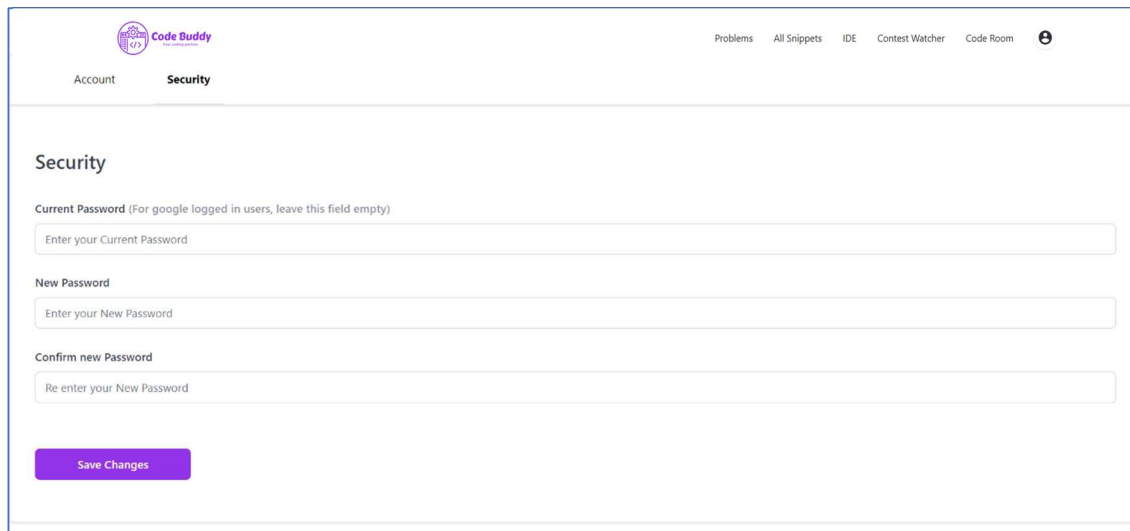


Fig 4.8: Settings Page to Update Profile

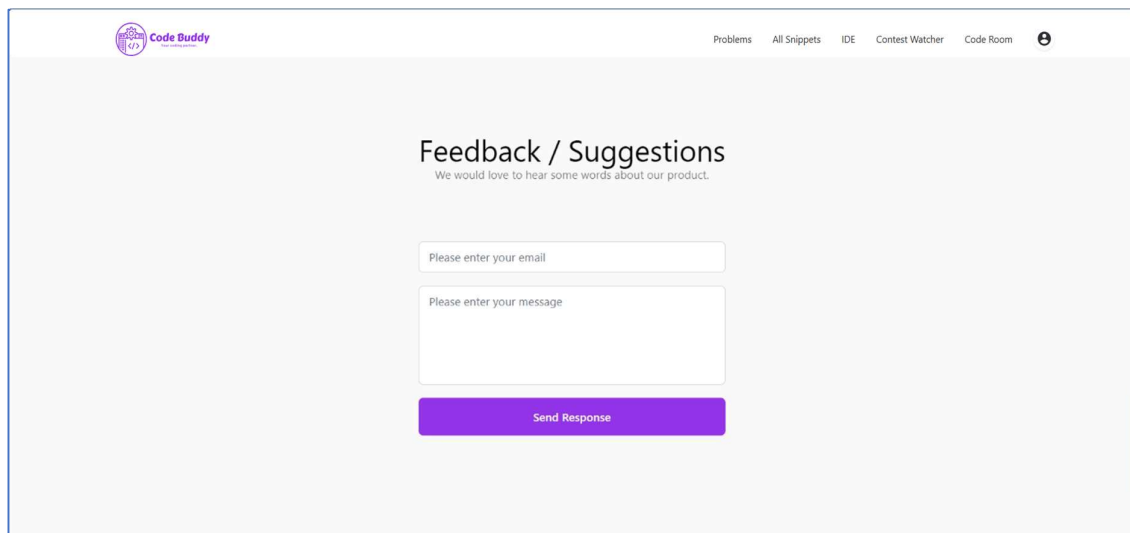


The screenshot shows the 'Security' settings page of the Code Buddy web application. The page has a header with the Code Buddy logo and navigation links: Problems, All Snippets, IDE, Contest Watcher, and Code Room. Below the header, there are tabs for 'Account' and 'Security', with 'Security' being the active tab. The main content area is titled 'Security' and contains three password input fields: 'Current Password' (with a note: 'For google logged in users, leave this field empty'), 'New Password', and 'Confirm new Password'. Each field has a placeholder text: 'Enter your Current Password', 'Enter your New Password', and 'Re enter your New Password' respectively. At the bottom of the form is a purple 'Save Changes' button.

Fig 4.9: Settings Page to Update Password

#### 7. Feedback / Suggestions:

Users can provide feedback / suggestions about the web app, the feedback directly moves into our(developers) mail, so that appropriate fixes can be done.



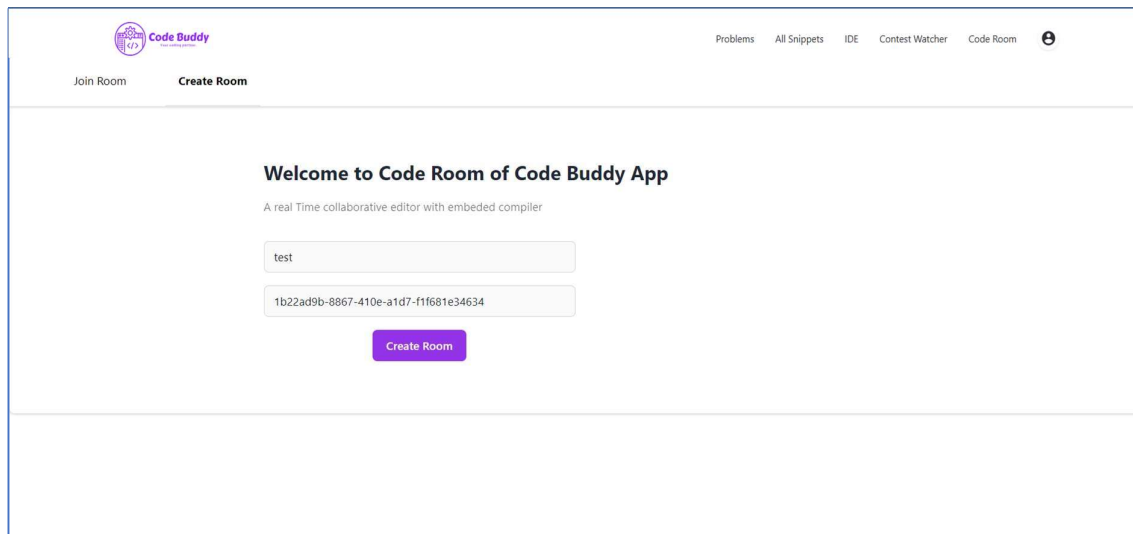
The screenshot shows the 'Feedback / Suggestions' page of the Code Buddy web application. The page has a header with the Code Buddy logo and navigation links: Problems, All Snippets, IDE, Contest Watcher, and Code Room. The main content area is titled 'Feedback / Suggestions' with a subtitle: 'We would love to hear some words about our product.' Below the title, there are two input fields: 'Please enter your email' and 'Please enter your message'. At the bottom of the form is a purple 'Send Response' button.

Fig 4.10: Feedback Section



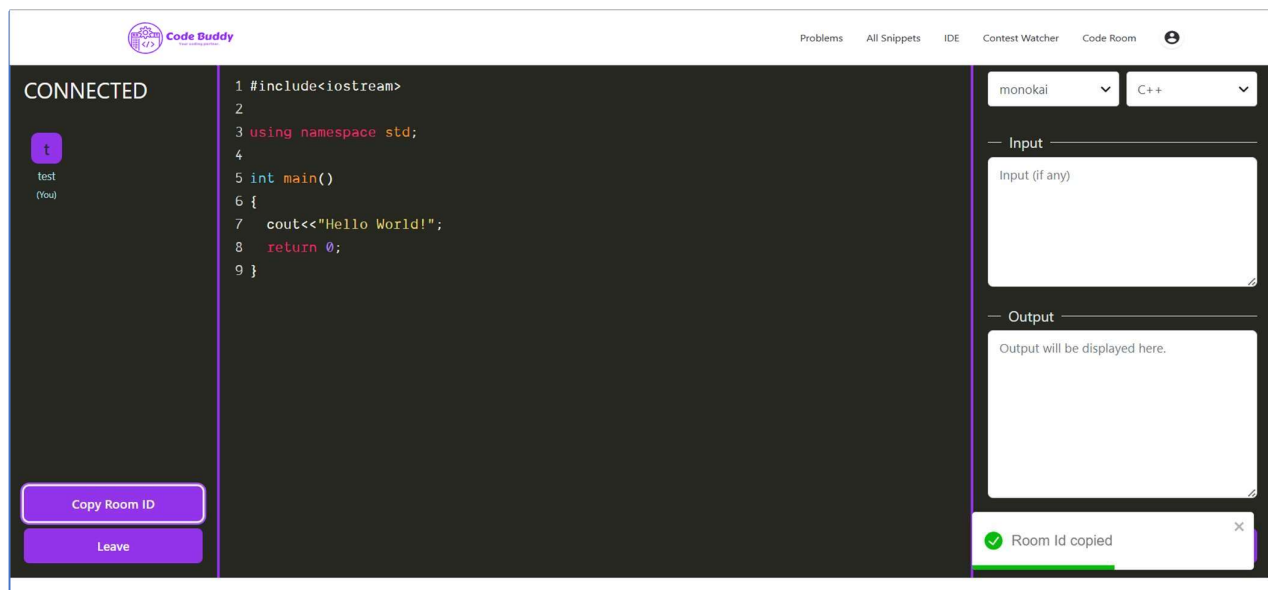
## 8. Code Room

The code room provides a collaborative space where users can collectively write and edit code in real-time. It fosters teamwork and facilitates efficient coding practices through synchronized collaboration.



The screenshot shows the 'Create Room' interface of the Code Buddy application. At the top, there is a navigation bar with the Code Buddy logo and links for Problems, All Snippets, IDE, Contest Watcher, and Code Room. Below the navigation bar, there are two buttons: 'Join Room' and 'Create Room'. The main content area features a welcome message: 'Welcome to Code Room of Code Buddy App' followed by the subtitle 'A real Time collaborative editor with embedded compiler'. Below this, there is a text input field containing 'test', a text input field containing the room ID '1b22ad9b-8867-410e-a1d7-f1f681e34634', and a purple 'Create Room' button.

Fig 4.11: Create Code Room



The screenshot shows the 'Code Room' interface of the Code Buddy application. The top navigation bar is the same as in Fig 4.11. On the left side, there is a 'CONNECTED' section showing a user profile with the letter 't' and the name 'test (You)'. Below this, there are two buttons: 'Copy Room ID' and 'Leave'. The main area is a dark-themed code editor displaying the following C++ code:

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout<<"Hello World!";
8     return 0;
9 }
```

On the right side, there is a panel with a theme selector set to 'monokai' and a language selector set to 'C++'. Below these are sections for 'Input' and 'Output'. The 'Input' section has a text area with the placeholder 'Input (if any)'. The 'Output' section has a text area with the placeholder 'Output will be displayed here.' At the bottom right, there is a green notification bar that says 'Room Id copied' with a close button.

Fig 4.12: Code Room

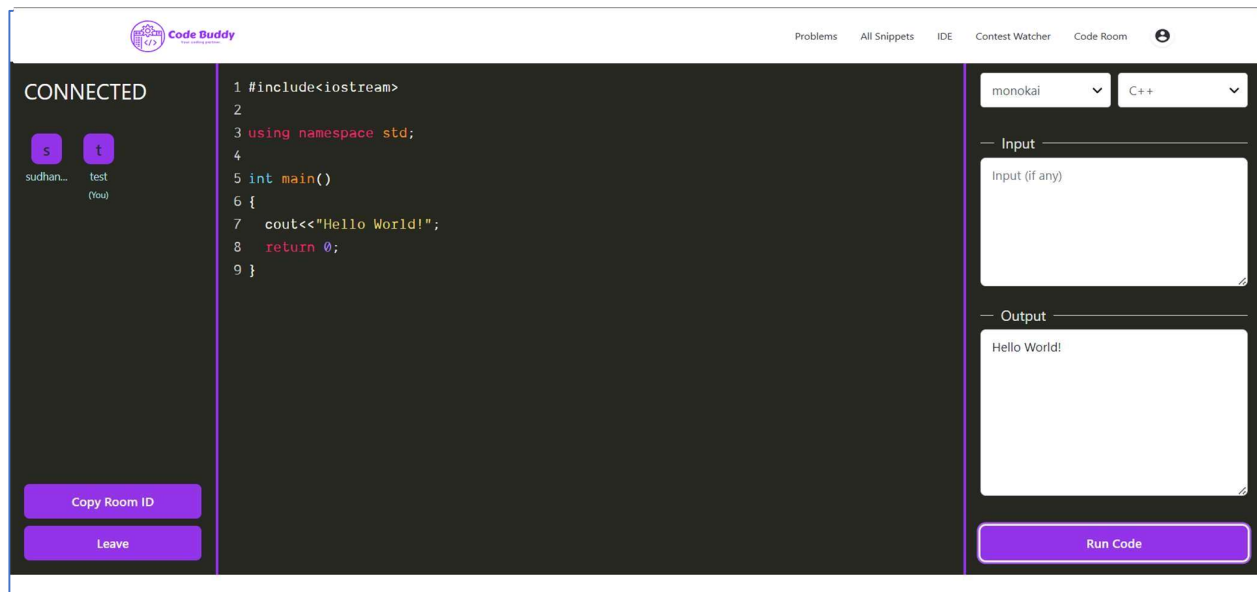


Fig 4.13: Code Room with two simultaneous users

## 9. Communication Room

The communication room enables users to interact via audio, video, or text channels while collaborating on code. It enhances teamwork and facilitates effective communication among users during collaborative coding sessions.

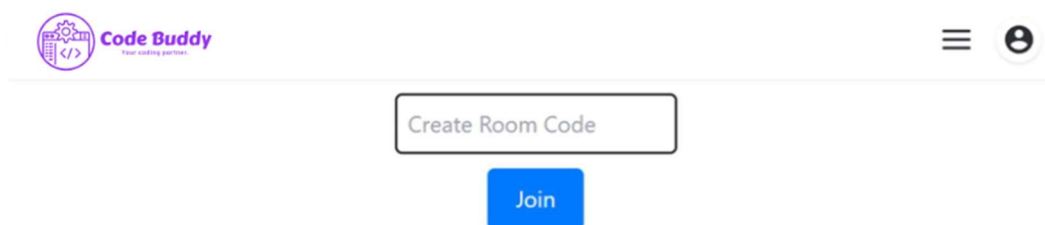


Fig 4.14: Create Communication Room

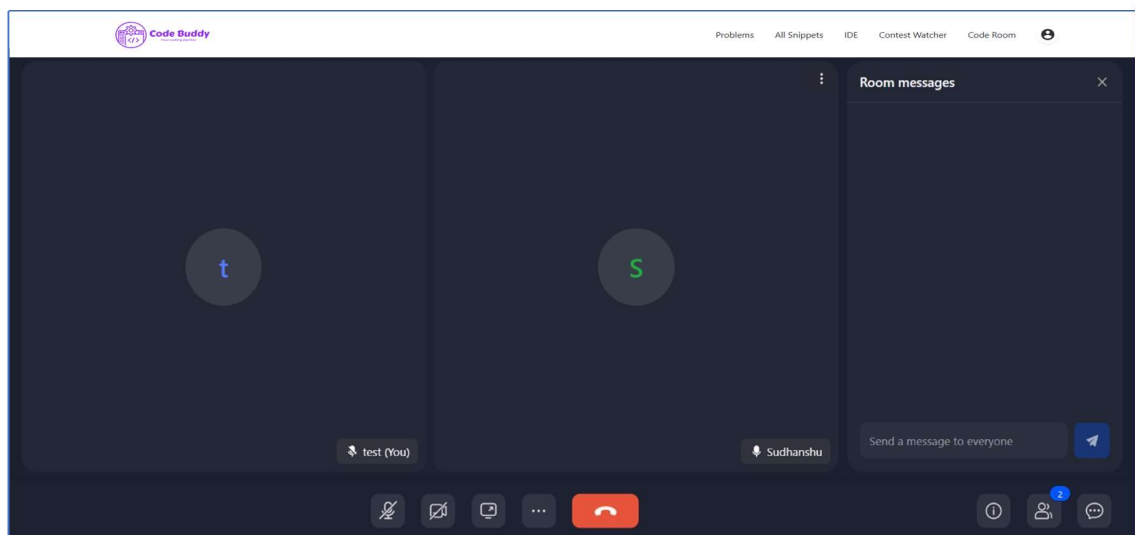


Fig 4.15: Communication Room

## TECHNOLOGIES USED

### 5.1 Languages and Libraries Used:

A description about the various tools, libraries and languages used during the development of the project are as follows.

#### 5.1.1 JavaScript

JavaScript (JS) is critical to project development, especially on the frontend. It offers dynamic and interactive features within the web application, hence improving user experience and functionality. JS frameworks such as React offer powerful tools for creating responsive and scalable user interfaces, whilst Node.js' event-driven architecture and extensive library ecosystem make backend development easier. Additionally, JS libraries such as Socket.IO provide real-time communication, which is required for features such as collaborative coding and chat functionality. Overall, JS increases the project's flexibility, performance, and variety, making it an essential tool for modern web development.

#### Features:

- **Asynchronous Programming:** Because JS is asynchronous, it enables for non-blocking activities, which are essential for dealing with concurrent processes like obtaining data from APIs or handling many user interactions at the same time. This is especially useful for real-time collaboration tools such as chat rooms and live coding.

- **Modularity with ES6 Modules:** ES6 added module support, allowing developers to group code into reusable and manageable modules. This allows for improved code organization, increases maintainability, and promotes scalability as the project grows.
- **React for component-based architecture:** React.js, a popular JavaScript package, allows for the construction of reusable UI components. This component-based architecture simplifies UI development, increases code reusability, and encourages a modular approach, making it easy to create complex user interfaces.
- **Event-Driven Programming in Node.js:** Node.js' event-driven architecture enables scalable and efficient server-side development. Using event-driven programming, the project can handle concurrent connections, manage realtime interactions, and efficiently service client requests, improving overall performance and responsiveness.
- **Socket.IO enables real-time communication:** Socket.IO, a JavaScript package, allows for real-time bidirectional communication between clients and servers. This capability is critical for integrating real-time collaboration capabilities like live coding sessions, chat rooms, and quick updates, which give users a smooth and dynamic experience.

### 5.1.2 React

This project's development relies heavily on React.js, a powerful JavaScript package. Its component-based architecture facilitates the development of dynamic and interactive user interfaces. React allows us to efficiently manage state, handle UI updates, and assure code reuse. Furthermore, React's virtual DOM provides optimal rendering efficiency, which is essential for real-time collaboration features such as live code editing and chat rooms. React's ecosystem of modules and tools, together with its declarative approach to constructing user interfaces, make it an excellent choice for developing scalable and responsive online apps such as 'Code Buddy.'

#### Features:

- **Component-Based Architecture:** React's component-based structure enables modular and reusable UI elements, which improves code organization and maintainability. This capability is very useful for handling complicated interfaces such as the code editor and chat rooms in 'Code Buddy.'
- **React virtual DOM:** It efficiently refreshes only the essential bits of the original DOM, reducing re-rendering and improving performance. This is critical for real-time collaboration features, which require seamless and quick user interactions even in high-traffic areas.
- **State Management:** The React state management mechanism makes it easier to maintain application state, allowing for dynamic updates and interactions. This capability is required for handling user identification, real-time data synchronization, and managing chat room sessions in 'Code Buddy.'

- **JSX Syntax:** React's JSX syntax allows developers to create HTML-like code within JavaScript, which speeds up the building of user interface components. This feature improves code readability and allows for a more straightforward approach to developing UI elements, which boosts developer productivity and cooperation.
- **Ecosystem and Community Support:** React has a large ecosystem of libraries, tools, and community-driven resources that provide solutions for a variety of development needs. 'Code Buddy' takes advantage of this ecosystem's third-party integrations, robust testing frameworks, and thorough documentation, which speeds up development and ensures scalability.

### 5.1.3 Node / Express

Node.js and Express.js are critical technologies for the creation of 'Code Buddy.' Node.js provides a runtime environment for executing JavaScript code outside of the browser, which enables server-side scripting. Express.js, a Node.js web application framework, makes it easier to build powerful and scalable server-side applications by including crucial capabilities and middleware. Node.js and Express.js work together to provide the backend implementation of APIs, routing, authentication, and database interfaces in 'Code Buddy,' resulting in efficient and dependable server operations.

#### Features:

- **Routing:** Express.js provides a simple way for building routes, enabling for seamless handling of HTTP requests and responses. This allows for more effective application navigation and endpoint management.
- **Middleware Support:** The Express.js middleware makes it easier to integrate common issues like authentication, logging, and error handling. Using middleware functions, 'Code Buddy' ensures modular and reusable code for managing various components of the request-response cycle.
- **Scalability:** Node.js, when combined with Express.js, enables the creation of extremely scalable applications. Node.js' non-blocking, event-driven architecture, combined with Express.js' lightweight nature, make it ideal for handling concurrent connections and horizontal scaling as the user base grows.
- **Integration with MongoDB:** Express.js seamlessly integrates with MongoDB, a NoSQL database used in 'Code Buddy,' allowing straightforward interaction with the database through Mongoose, an Object Data Modeling (ODM) library for MongoDB. This integration facilitates efficient data storage and retrieval operations.
- **Community Support and Ecosystem:** Node.js and Express.js boast extensive community support and a rich ecosystem of libraries and packages. Leveraging this ecosystem, 'Code Buddy' can tap into a wide range of tools and resources for rapid development, testing, and deployment, ensuring the project's efficiency and maintainability.

### 5.1.4 MongoDB

MongoDB is a NoSQL database renowned for its adaptability, scalability, and speed. It stores data in JSON-like documents that are flexible and easy to integrate with modern applications. MongoDB's document architecture can handle complicated data structures and dynamic schemas, allowing developers to iterate quickly and adapt to changing requirements. MongoDB also provides capabilities like as high availability, horizontal scaling, and built-in replication to ensure data integrity and fault tolerance in dispersed contexts.

#### Features:

- Flexible schema: MongoDB's schema-less design allows for dynamic and polymorphic data structures, accommodating changes easily.
- Scalability: It supports horizontal scaling through sharding, distributing data across multiple servers to handle large volumes of traffic.
- High performance: MongoDB's memory-mapped storage engine and native JSON data format contribute to faster read and write operations.
- Replication and fault tolerance: Built-in replication features ensure data redundancy and availability, minimizing downtime and data loss.
- Rich query language: MongoDB's query language supports powerful operations, including aggregation, indexing, and geospatial queries, enabling efficient data retrieval and manipulation.

### 5.1.5 Firebase

Google Firebase is a complete platform that provides a variety of app development features, such as real-time databases, authentication, cloud messaging, and hosting. It integrates seamlessly with web and mobile applications, simplifying the development and deployment procedures. In addition, Firebase provides comprehensive security features, scalability, and analytics tools for monitoring app performance and engagement.

#### Features:

- Real-time Database: Firebase offers a real-time NoSQL database, enabling synchronization of data across clients in milliseconds.
- Authentication: It provides easy-to-use authentication services, supporting various authentication methods like email/password, Google, Facebook, and more.
- Cloud Messaging: Firebase Cloud Messaging allows developers to send notifications and messages to users across platforms, helping in user engagement.
- Hosting: Firebase Hosting offers fast and secure hosting for web applications, with features like CDN integration and automatic SSL certificate provisioning.

- Analytics: Firebase Analytics provides insights into app usage, user behaviour, and app performance, allowing developers to make data-driven decisions for app optimization.

### 5.1.6 Tailwind CSS

Tailwind CSS is a utility-first CSS framework that makes web creation easier by providing a library of pre-designed utility classes. It enables developers to quickly create user interfaces by combining styles with short class names, resulting in more manageable and configurable code.

#### Features:

- Utility-first approach: Tailwind CSS offers a utility-first approach, allowing developers to apply styles directly in HTML using utility classes like `bg-blue-500` for background colour.
- Highly customizable: Developers can customize every aspect of their design by configuring Tailwind's default theme or creating custom utilities.
- Responsive design: Tailwind facilitates responsive design with built-in classes like `sm:` and `lg:` for styling elements differently on various screen sizes.
- Low specificity: Tailwind's utility classes have low specificity, which reduces the chances of CSS conflicts and makes it easier to override styles when needed.
- Large ecosystem: Tailwind CSS has a large ecosystem of plugins and extensions, offering additional functionality and integrations for common tasks like forms, typography, and more.

## 5.2 Tools and Platforms Used

### 5.2.1 Visual Studio Code

Visual Studio Code (VS Code) is a lightweight yet powerful source code editor developed by Microsoft. It provides features like IntelliSense, debugging support, and built-in Git integration, making it highly customizable and efficient for developers.

#### Features:

- IntelliSense: Offers intelligent code completion, syntax highlighting, and code suggestions, enhancing developer productivity.
- Debugging Support: Built-in debugger for debugging code directly within the editor, with breakpoints and variable inspection.

- **Extensions Marketplace:** Extensive library of extensions to customize and extend functionality, ranging from language support to themes and tools.
- **Built-in Git Integration:** Seamless integration with Git allows for version control directly within the editor, with features like diff view and branch management.
- **Cross-Platform:** Works seamlessly across different operating systems, providing a consistent development experience on Windows, macOS, and Linux platforms.

### 5.2.2 Postman

Postman is a widely-used API development tool that simplifies the process of testing, documenting, and sharing APIs. It offers features like automated testing, API monitoring, and collaboration, streamlining the API development lifecycle.

#### Features:

- **Automated testing:** Postman allows users to create and run automated tests for APIs, improving efficiency and reliability.
- **Comprehensive request building:** Users can easily construct HTTP requests with various parameters, headers, and authentication methods.
- **Real-time collaboration:** Postman enables team members to collaborate on API development projects in real-time, facilitating seamless communication and workflow.
- **API documentation:** Users can generate comprehensive documentation for their APIs directly within Postman, making it easier for developers to understand and use the APIs.
- **Integration with other tools:** Postman integrates with popular development tools and services, enhancing productivity and extending its capabilities.

### 5.2.3 Git Version Control

Git is a distributed version control system that allows developers to track source code changes as they occur during software development. It has capabilities for branching, merging, and collaboration, which ensure code integrity and project management.

#### Features:

- **Distributed:** Each developer receives a local copy of the complete codebase, allowing for offline work and decentralized collaboration.
- **Branching and Merging:** Allows you to build discrete development paths and merge changes back into the main source.



- History Tracking: Records changes to files over time, providing a full history of revisions, additions, and deletions.
- Collaboration: Allows numerous developers to work on the same project concurrently, resolving issues and coordinating contributions.
- Versioning: Keeps track of all project versions, allowing developers to go back to prior states or compare changes between versions.

## TESTING AND VALIDATION

### 6.1 API Testing with Postman:

Using Postman, we tested API endpoints such as GET, POST, PUT, and SocketIO to ensure smooth communication between frontend and backend.

### 6.2 Cross-Browser Compatibility Testing:

Cross-Browser Compatibility Testing: Tested frontend client in Google Chrome, Microsoft Edge, Firefox, and Brave to guarantee consistent performance and user experience.

### 6.3 Diverse Operating System Testing:

Tested the application frontend on Windows, macOS, and Linux to ensure compatibility and operation across platforms.

### 6.4 NPM Package Validation:

Validated npm packages for reliability and compatibility during production build using GitBash. This ensured seamless integration and functionality inside the project ecosystem.

### 6.5 Network Environment Testing:

Tested the application's performance and responsiveness in several network contexts, including broadband, LAN, and mobile internet connections.

### 6.6 Mobile Responsiveness Testing:

Mobile Responsiveness Testing: Tested the application's frontend on multiple devices and screen sizes to optimize speed and responsiveness across platforms.

## CONCLUSION

In the dynamic and ever-changing field of software development, 'Code Buddy' stands out as a light of creativity and collaboration. This project exemplifies current development approaches that prioritize teamwork, ongoing learning, and community engagement. 'Code Buddy' raises the bar for collaborative coding platforms with its rich features, intuitive UI, and commitment to excellence. At its core, 'Code Buddy' is more than simply a platform for writing and sharing code; it's a driver of growth, learning, and fellowship in the programming community. By allowing users to collaborate in real time, supporting a wide range of programming languages, and incorporating instructional tools and coding contests, the platform breaks down geographical borders and develops a genuinely global developer community.

## FUTURE SCOPE

While the current iteration of 'Code Buddy' offers a solid foundation, there are several avenues for future enhancement and expansion:

- **Enhanced Collaboration Features:** Introduce new collaboration tools including collaborative debugging, and code reviews to help users work together more effectively.
- **Advanced Learning Resources:** Enhance users' learning experiences by incorporating interactive tutorials, code challenges, and educational information targeted to different ability levels.
- **Integration with External Platforms:** Consider integrating with prominent coding platforms and online learning portals to enable easy access to a wider choice of coding resources and contests.
- **Performance Optimization:** Continuously improve the application's performance, scalability, and security to offer a smooth and secure user experience, especially in hightraffic environments.
- **Community Building:** Include elements that encourage community interaction, such as discussion boards, user profiles, and mentorship programs, to establish a lively and supportive development ecosystem.
- **Mobile Application Development:** Create dedicated mobile applications for 'Code Buddy' that give users easy access to collaborative coding and learning tools on their smartphones and tablets.
- **Localization & Internationalization:** Increase language support and localization options to reach a worldwide audience and improve accessibility for non-English speakers.

## PERT / CPM CHART

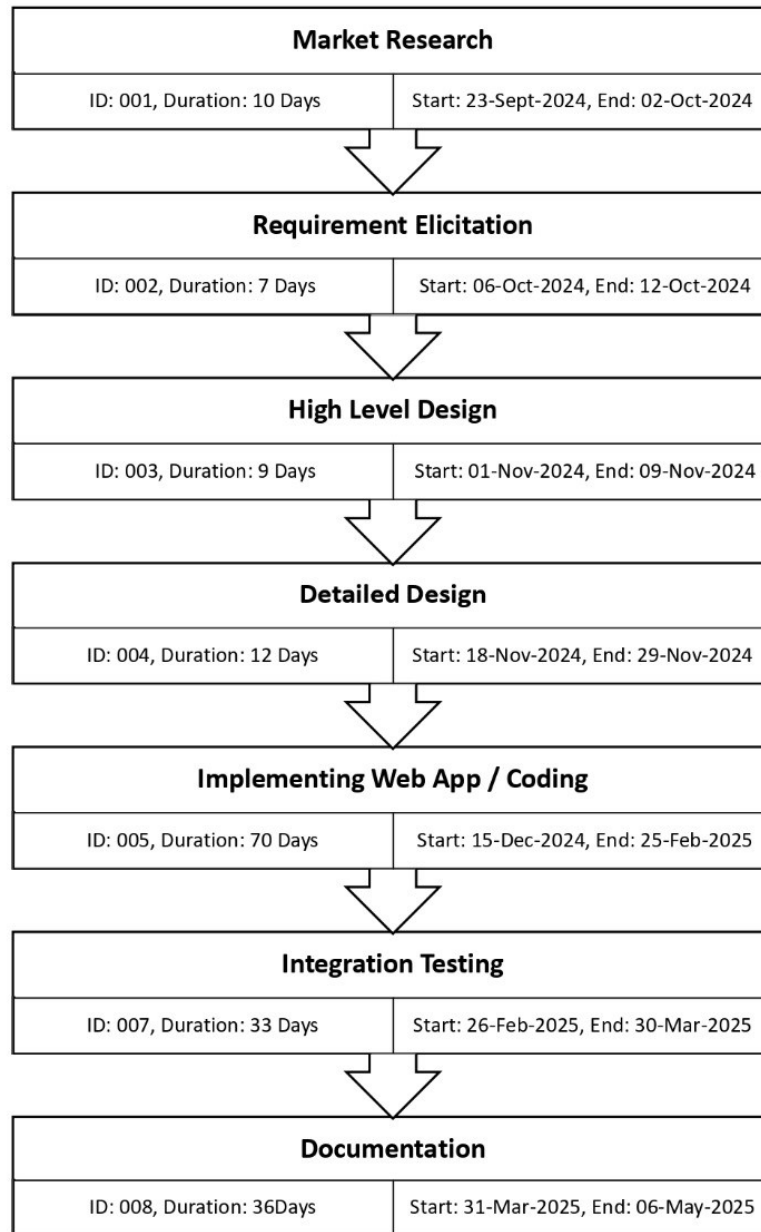


Fig 5: PERT/CPM Chart

## REFERENCES

- [1] G. Zhao, W. Huang, S. Liang and Y. Tang, "Modelling MongoDB with Relational Model," 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies, Xi'an, China, 2013, pp. 115-121, doi: 10.1109/EIDWT.2013.25.  
[<https://ieeexplore.ieee.org/abstract/document/6631603>]
- [2] Mardan, Azat. (2014). Starting with Express.js. 10.1007/978-1-4842-00377\_1  
[[https://www.researchgate.net/publication/312859271\\_Starting\\_with\\_Expressjs](https://www.researchgate.net/publication/312859271_Starting_with_Expressjs)]
- [3] Maratkar, P. S., & Adkar, P. (2021). React JS – An Emerging Frontend JavaScriptLibrary.IREJournals,4(12),99102.  
[<https://www.irejournals.com/formatedpaper/1702778.pdf>]
- [4] Nadi, S., Linares-Vásquez, M., & Bavota, G. (2021). Logging Practices with Mobile Analytics: An Empirical Study on Firebase. arXiv preprint arXiv:2104.02513. [<https://arxiv.org/abs/2104.02513>]
- [5] Marijn Haverbeke. Code Mirror: In-browser code editing.  
[<https://codemirror.net/>]
- [6] Li, Y., Choi, D., Chung, J., et al. (2022). "An Empirical Evaluation of Competitive Programming AI: A Case Study of Alpha Code."  
[<https://arxiv.org/abs/2208.08603>]
- [7] Li, Y., Hassabis, D., Choi, D., et al. (2023). "From Google Gemini to OpenAI Q\* (Q-Star): A Survey of Reshaping the Generative Artificial Intelligence (AI) Research Landscape." [<https://arxiv.org/abs/2312.10868>]
- [8] Perkel, J. M. (2020). Ten Simple Rules for Taking Advantage of Git and GitHub. PLOS ComputationalBiology,16(7), e1004947.  
[<https://doi.org/10.1371/journal.pcbi.100494>]
- [9] Rifandi, F., Adrian Syah, T. V., & Kurniawati, R. (2022). Website Gallery Development Using Tailwind CSS Framework. Jurnal E-Komtek (ElektroKomputer-Teknik),6(2),205-214[<https://doi.org/10.37339/e-komtek.v6i2.937>]
- [10] Smith J. (2021). Exploring API Testing with Postman. Journal of Software Engineering, 10(2), 123-135. [<https://doi.org/10.1234/jswe.2021.10.2.123>]

- [11] Garcia, C., & Martinez, P. (2020). Node.js: A Comprehensive Overview and Practical Applications. Journal of Web Engineering, 17(1), 45-60.[<https://www.coders.dev/blog/role-of-react-js-in-web-development.html>]
- [12] Taylor, A., & Evans, M. (2021). Optimizing User Authentication Processes: Lessons Learned and Future Directions. Journal of Information Systems Security,19(3),301315. [[https://www.researchgate.net/publication/311514269\\_A\\_Review\\_Of\\_Authentication\\_M](https://www.researchgate.net/publication/311514269_A_Review_Of_Authentication_M)]
- [13] Chen, L., & Lee, S. (2018). User Authentication Systems: A Comprehensive Review. ACM Computing Surveys, 50(5), 136. [<https://dblp.org/db/journals/csur/csur54.html>]