MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

A Project Report on

Collaborative Coding Plaform for all Coding Enthusiasts

Submitted by,

Aniket Kumar Yadav **(Exam Seat No. B224065)**

Pantawane Shreyash Ashok **(Exam Seat No. B224070)**

Guided by,

Mrs. Bhagyashree Alhat

A Report submitted to MIT Academy of Engineering, Alandi(D), Pune, An Autonomous Institute Affiliated to Savitribai Phule Pune University in partial fulfillment of the requirements of

BACHELOR OF TECHNOLOGY in

Computer Engineering

School of Computer Engineering and Technology

MIT Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

Alandi (D), Pune – 412105

(2022–2023)

MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

CERTIFICATE

It is hereby certified that the work which is being presented in the BTECH Project Report entitled **"Collaborative Coding Plaform for all Coding Enthusiasts"**, in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Engineeringand submitted to the **School of Computer Engineering and Technology** of **MIT Academy of Engineering, Alandi(D), Pune, Affiliated to Savitribai Phule Pune University (SPPU), Pune**, is an authentic record of work carried out during Academic Year **2022–2023**, under the supervision of **Mrs. Bhagyashree Alhat**, **School of Computer Engineering and Technology**

Aniket Kumar Yadav **(Exam Seat No. B224065)**

Pantawane Shreyash Ashok **(Exam Seat No. B224070)**

Mrs. Bhagyashree Alhat	**Pranav Shriram**	**Prof. Ranjana Badre**
Project Advisor	**Project Coordinator**	**Dean**

Director/Dy. Director(AR) **External Examiner**

DECLARATION

We the undersigned solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of **Mrs. Bhagyashree Alhat**.

We assert the statements made and conclusions drawn are an outcome of our project work. We further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this Institute/University or any other Institute/University of India or abroad.
3. We have followed the guidelines provided by the Institute in writing the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Aniket Kumar Yadav **(Exam Seat No. B224065)**

Pantawane Shreyash Ashok **(Exam Seat No. B224070)**

Abstract

In this new era of Computer Technology and Automation there is an outburst in new websites and apps which help people in their betterment. As we can see there has been huge development in Technology in the last decade. As of now we all are recovering from COVID-19 pandemic, in this period we got to know the importance of Computer technology. To develop Career in this field people need to know about coding and for doing so in groups so they can get help right in there so they can grow widely, the collaborative coding and editing sites must be there. This project is to help this type of people and lead them to be a master coder and as all the project is associated with cloud services so it can be available anywhere and anytime for users with safe environment exposure. If you are working in a team so it will help you to work on all the modules at a same time and if anyone member is not getting the idea of project or done some mistakes , then other members can help him or get the errors or bug out in a meantime.

Acknowledgment

We want to express our sincere gratitude to our guide Mrs.Bhagyashree Alhat for her constant encouragement and valuable guidance during the completion of this paper and work behind it. We also want to express our gratitude to School Dean Mrs.Ranjana Badre for her continuous encouragement. We would also like to thank everyone who helped us in testing our project. We would be failing in our duty if we do not thank all the other staff and faculty members for their experienced advice and evergreen co-operation.

Aniket Kumar Yadav

Pantawane Shreyash Ashok

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Now-a-days, Coding is very important in part of academics of Computer Science aspirants or work life of Software Engineers or associate fields, to practice the code there are many platforms are present in market like Leetcode, Codeforces,etc. but none of the platforms gives facility to code with others like having a team together to code or getting assistance in learning code in real-time virtual instance. or even in during interviews of software companies many candidates got caught for cheating and implementing plagiarism code, to avoid that we must have some platforms which won't allow plagiarismic content or highlight the content to other person(interviewer).

In this new era of computer science and automation there is an outburst in new webs and apps which help people in their betterment. As we can see there has been huge development in Technology in the last decade. As of now we all are recovering from corona virus pandemic, in this period we got to know the importance of Computer technology. To develop Career in this field people need to know about coding and for doing so in groups so they can get help right in there so they can grow widely, the collaborative coding and editing sites must be there. This project is to help this type of people and lead them to be a master coder and as all the project is associated with cloud services so it can be available anywhere and anytime for users with safe environment exposure.

1.2 Project Idea

The Idea for the project is developing an platform for coders to learn coding with other peoples at a time or his coding must be surveillance by other person so he will be guide him by correcting him or helping him in that part. Coding is used the communication between the computer models and many computer science aspirants wants to learn and satisfies their curiosity by learning the coding in the first step, so to make their first step stronger and most rooted we wants to help them in learning coding with their other friends and even with their guide so it will be helpful for them. This refers to the learning with other person that person will keep on checking on your progress, helps you in building the concepts and keeps an eye on errors and bugs.

There is a concept called pair programming, by which two or more developers can work together on the same project, they help each other and learn from each other. This makes team more coherent and develops the better product. We are also looking for the same algorithm but with not enabling or restricting the people from copy-paste the work of other/by making it plagiarism free platform for it's users.

1.3 Motivation

1. Growing interest of people of all age group in coding

As, the jobs and IT industry is booming now so the world is becoming more curious about it and eager to get into it. so as in a result they are prefering to learn the new and most relevant techniques adhere to it.

2. Growing the case of plagiarism in interview process

During covid-19 virus pandemic all the hiring process of all IT companies/firms shifted to online so many candidates choose to copy the code over the internet and get hired, so this may result to harm the firm and wasting the resources on unwanted things.

3. Helping Working individuals of companies in working together in a project.

If a team is working on a project, and if everyone is working all individuals then there is chances for error or bugs getting into it, as the environment or dependencies will get into trouble, so if they work all together on same environment and on same dependencies this will never occur.

4. Developing the understanding of project in each member of team.

If group members are working together then they might develop the understanding with each other towards project. and this will create an friendly work environment in company

1.4 Project Challenges

1. Two peoples might work on same module in project.

If in any company there are many projects going on,, and in a project there are group of 6 people working,,then there might be chances that some group members might start working on same modules and get the different variables and classes to code.

2. Everyone have different style of working and writing the code.

In a group every members must have unique ideas and after communicating them with your group members you will try to implement them in real while writing code you might see that the way to writing the code or expressing it might be or will be different from other

3. If any unauthorized members got an access to for project then it might create the issue of security. Let's say, as an example, If the person who don't know anything about the project got the access then he might delete the work others have done in nuissance.

1.5 Proposed Solution

Basically, here we are building an website application which is just like a coding challenge platform but here you can code with your partner and build and solve many problems. Even, in case of Software engineers it will facilitate them to code together on particular project and get the work done fast as a team. With the help of website development languages like html, css, javascript,socket.io for connecting the users at the same platform and Mongodb for database purpose. Here we have deployed this project on cloud so the scaling of project and traffic control won't be an issue anymore for us to handle.

1.6 Major Contribution

The many famous and present IDE's like Visual Code, DevC++, CodeBlocks, etc. are designed for coding in individual arena which might give the users solo performing space or self paced coding experience but it can't give the group learning or help at an instance if you are wrong or special montioring happening at a time with others. In present IDE's you have ask from your mentor it but here in collaborative platforms they can notice you and your incomings and help you. The sites where you can do collaborative coding are not giving you that user experience so people might get bored there. or at some places people can have music or lighting system which will distract them from their basic and main motto or goal.

1.7 Project Report Organization (Chapter wise summary)

This report comprises of following chapters:

Chapter 1: The project overview is presented in this chapter. It goes into the project's background, difficulties, and solutions.

Chapter 2: A comprehensive review of the literature is presented in this chapter. In this chapter, we addressed the research on handwriting recognition that numerous writers had presented to conferences and journals.

Chapter 3: The objectives of the project and a brief description of what we're doing are also provided, along with a list of the specifications we'll be using.

Chapter 4: An overall description of the requirements is being discussed.

Chapter 5: The system architecture is described in detail. The project's models in their entirety, as well as the implemented algorithms, are discussed.

Chapter 6: This section covers the implementation and output of the system.

Chapter 7: Results analysis is covered in this section.

Chapter 8: The project's conclusion has been briefly discussed, along with the scope.

Chapter 2

Literature Review

2.1 Related work And State of the Art (Latest work)

[1] Collaborative IDE for E- classroom with progress tracking of students

Author Bhargav V. Parekh understanded the importance of an IDE for collaborative coding as he mentioned that between the year 2018 and 2019, online coding bootcamps increased 177

For collaboration on editor he used Firepad(firepad.io) whose features includes cursor position synchronization, text highlighting, user attribution, presence detection, and version checkpointing. To deploy and run terminal on web he used Weblinux and GoTTY. Other software/ Platform requirements were i) JAVA JDK 8+ (to build back-end REST APIs), ii) Apache Maven (for creating exe version of back-end code), iii) AWS EC2 (used t2.micro as a deployment instance), iv) MongoDB(for storing JSON-like documents with schema), v) AngularJS NodeJS(for viewing deploying of frontend server) and finally vi) Go language (for compilation of code)

Here author demonstrated the project by creating a live python collaboration platform where Instructor created a live classroom and shared join ID with other user, later user need to enter his/her name, description and room ID for accessing the IDE. After joining various functionalities such as progress tracking, anti-cheating, run/ compiling code, saving/ uploading code were present for better collaboration

between Instructor and student.

[2] Vivace: a collaborative live coding language and platform

Authors from Polytechnic Institute – IPRJ/UERJ developed a collaborative live music coding (which they called "freak coding") language named : Vivace, which borrows elements from ixi language. Vivace is used to generate sound, video, poetry images which can be shared and edited to other coders via web browser. Whole project was in development with dedicated congress, International Conference on Live Coding (ICLC) which motivated authors to use ShareJS (JavaScript for using advantages of Web Technologies), etherpad, Web Audio API for complete development of Vivace.

While coding every parameter is changed by the user over time with the values written in numerical sequences, surrounded by brackets. For proper composition of music mnemonic commands are added which can be reversed, transposed and even can be replaced with sequence based on list comprehensions. Authors say Vivace is made keeping in mind the creativity of the so-called freak coder, they even tried to demonstrate their project to mass audience in Brazil where live music generation was taking place on their platform and coders around the world were collaborating live, real time sharing of code generation of music was shown to mass audience on a large projector. On the same event authors promoted audience to collaborate for making the platform more human friendly and interactive to mass audience.

[3] Live coding the code: an environment for 'meta' live code performance

Authors Andrew Thompson Elizabeth Wilson from Queen Mary University used open source Lissajous and Flow for the creation of live coding of music where user can enter text sequences for the creation of musical parameters, which can be recorded and manipulated as well. Authors used Gibber as a browser-based live coding platform. Contrary to to above mentioned Vivace platform Lissajous + Flow uses meta live coding, where user can encapsulate a function and transfer it to sequencer constructor. As for the architecture "Lissajous" and "Flow" were connected via FlowLissajous plugin, where user can use several functions such as update(), model() view() for music creation.

Using Gibber whole environment is designed in such a way that listeners not only can listen the music but can add/remove musical macros while affording the original macro as well. For the creation of music 3 main actions were added in the language namely: EVAL, REC-START and REC-STOP, using which user can alter the notes, add effects and change the timings for that particular code. For the implementation purpose authors used SuperCollider, which they found comparatively powerful as the OSC messages can communicate with SC server. And finally for the future purpose authors suggested to use Siren environment which will bring the coding environment close to user.

[4] Synerg(e)ia: A Networked Collaborative Live Coding Environment

Main idea in Synergia is that all the users can create and use their own / existed SuperCollider workspace buffer and use it passing the text buffers. Using Emacs audio and sharing can be compiled on-the-fly using various musical languages such as Foxdot, Emacs Impromptu etc. For architecting the system project communicates clients using OSCGroups via SuperCollider for peer-to-peer messaging. User must use Linux Debian using VPN client to connect to the server, after establishing the connection user can take advantage of various functionality as multiple windows, buffer visibility, screen motion etc, and can use control commands to customise emacs variables like color and font.

As the project requires live coding synchronisation between server and various clients

it occurs to be a high stake necessity for which authors used clock mechanism at backend. As for the audience after logging to synergia interface and connecting via SSH, they get the access to performers optical and musical nodes in real time using VPN configuration. According to authors other option for developers is to use SuperColliders Pen class. As mentioned in the introduction, Synergia is developed through the prism of certain core principles of remote live coding platforms, as stated by Lee Essl (2014). At the same time, we also focus on compatibility with our free software. Finally, the possibility for users to customize the platform. Therefore, a operating with Synergia is sufficient to cover the distance limitations of a communicating process over a network. This applies to both the performer and the audience. Moreover, in such communication systems, messages have an abstract and subjective form that is derived through the prism of artistic experimentation. In this case, the sender and receiver can be both performers and audience. More specifically, performers can convey different messages through improvisational musical expressions. In addition, we also receive intuitive feedback from our viewers. The audience receives both musical and visual messages from the artist and he can evoke an emotional response through gestures and non-verbal communication. In Synergia this is duplicated in text representation due to the limitations of remote visual feedback. Our vision is for fully interactive live coding performances, where an interactive communication relationship develops between the performer and the audience. Certain results in performance terms are derived from this relationship, in a sense replicating the physical act of live coding. Artists with Synergia suggest that the bond of mutual interaction can be strengthened. First, by the general concept of collaborative live coding, and second, by giving them the opportunity to "shape" the environment into a desired and convenient form in which live coding sessions take place.Another important aspect of Synergia is its ability to support live audiences. Anyone can join his live coding session on his Synergia by connecting to the server via OSCGroups and starting his Slang process locally. This skill is also important as physical performance requires feedback from the audience. Synergia allows viewers to send real-time her messages to the performer from a local slang process using a VPN configuration. However, the audience feedback feature needs to improve her terms of accessibility and convenience. Taking this a step further,

one idea is to introduce function for interaction between performer and audience. Also, most research in the field of remote live co-coding has focused solely on music. The same applies to synergies. A future goal is to integrate 's visual and sensory connections into interactive performance practice. With 's focus on incorporating telematic performance and visual expression into Synergia, we strongly believe it will provide a fair amount of expression for performers. A potential visual limitation could be the need for a graphical user interface. Synergia is accessible via Terminal, so visual option is limited. For example, one option is to sync the image and sound with SuperCollider's Pen8 class. Additionally, openFrameworks 9 has a number of ASCII art implementations that can run in the Synergia environment. At the same time, we strive to make it as easy as possible for the user to access his Synergia.

[5] The Live programming Lecturing Technique :

Interesting aspects of discussion during lectures and various student strategies yielded from research-related work and data collection. Interviews indicated that during the lecture, students seemed to experiment with different techniques for engagement, collaboration, and learning. This is consistent with the results of Bonwell et al. [9] and Man of the World [10]. All interviewed candidates reported copying (or attempting to copy) the instructor's code while writing it. A added that this was a good technique for an introductory course, but took away the focus as the topic became more advanced. Another technique used was to take note of the items of interest heard or seen during the lecture and, with sufficient time and concentration, to look them up after the lesson and return to these. A majority also told that the main secret to success was simply paying attention to what was going on. This allowed to provide a bird's eye view, thereby providing an overview of code examples, and drawing attention to what the instructor said to during walkthroughs and coding. His students surveyed enthusiastically explained the benefits of live programming. A common feature reported in our results is that it is a beneficial learning method. They say it's easy to ask questions and can lead to new examples, demo changes, and interesting distractions along the way. A thorough explanation with examples from is very well received and is very helpful. In addition, observation also gives students an opportunity to see the big picture and understand the strategies used

to solve problems. As promoted by Gaspar and Langevin [15], the focus is on the process rather than the final product, allowing us to learn more about the programming process and the strategies used. It is also stated that by doing this, you will learn various tips and tricks related to development tools, shortcuts, and other related information. Another interesting observation from the interview is that the audience (students) perceives this type of lecture as interactive and active. This is true not only for her period when encoding simultaneously, but also when purely observing. Students feel activated. As reported by Freeman et al. [12] learning outcomes improved in support of active learning strategies. Their results show that this is supported regardless of class size or course type. Our interviews are inconclusive on this point, suggesting that requires further investigation. As shown, some students preferred the small class size and her 4,444 others found her one-on-one learning ideal. students investigated the speed of names as an important task. If the instructor moves too quickly, it becomes very difficult to understand the topic, understand the process, and understand the rationale for the decisions made during the session. Judging "too fast" is a subjective opinion, but they often point out that they usually fail at some point. This may be due to some amazing keyboard shortcut command that does magic trick, or is lost while modifying the file or the description is not understandable. Lost, her students surveyed note that regaining attention is nearly impossible, with catching up. Unsatisfactory explanations are also an issue and have been reported by multiple reporters. Following requires energy to understand and learn. Instructors must be able to comment on or explain the code written in a concise, accurate, and reasonable manner. If that fails, the entered code is left without a context connection and appears lost with no practical way to remember it. As Gaspar and Langevin [15] investigated, the programming learning process goes beyond syntax and detailed program code. It's easy for a student to get her fixated on code details, and it can be difficult to get the big picture of what the is doing. Still, to understand what we're doing on a small scale, we need to look at the big picture. I also asked if you could provide some thoughts on how this lecture technique could be further improved. Some of the respondents immediately said that better explanations came to mind first. The reason is that is aware of it in his lectures, so he doesn't necessarily say the obvious. These seemingly insignificant details are

especially vital to her understanding of the introductory course. The second item on our list of improvements relates to the results of reported issues: speed. Slow speed will make the student feel bored and too elementary, but too fast speed will make him lose the attention of Auditorium. They point out that there is no solution on how to balance the's speed, but if it still doesn't suit their tastes, it is perceived as an active barrier.

[6] Collaborative Interfaces for Ensemble Live Coding Performance

This research is a practice-led investigation into collaborative user interfaces within the practice of live coding; the act of writing computer code for generating improvised music live in front of an audience. It examines the impact of user interface design parameters on group creativity and explores the roles of data, text, and programming languages as media for musical communication. Utilising a multi-faceted research method that combines iterative "participatory design" (Spinuzzi, 2005) with performance-led "research in the wild" (Benford et al., 2013), this research couples ethnographic and autoethnographic observations to gain insight into the practice of ensemble live coding and inform software design. Three novel collaborative interfaces have been developed as part of this research that explore various facets of musical collaboration in live coding. Each inter- face was developed through an iterative and reflexive methodology focused on user-centred design and was employed in a cyclical process of artistic practice and refinement based on user evaluation and in-depth study. The first interface, titled Troop, is a shared text editor that allows multiple executors to work together on the same body of code. The second, CodeBank, examines how collaborative, personal work influences creativity and improvisation. Finally, PolyGlot combines multiple live coding languages into one collaborative interface, allowing live coding musicians to perform together regardless of their language skills. In addition to these three graphical interfaces, the existing Live coding language FoxDot has been extended to facilitate the exchange of musical information within the ensemble. Each interface was used in live performances by the Yorkshire Programming Ensemble and evaluated through group interviews. This interview explores the themes and experiences of immediacy, trust and risk related to human-computer interaction and ensemble communication.

2.2 Limitation of State of the Art techniques

2.2.1 Various Technical Limitations are :

1. Database being a community server, encryption and data protection features aren't present in MongoDB.

2. Conflict Resolution - When 2 or more clients try to edit same line of code, conflict occurs at server, though it uses principle of first come first served, it glitches can be seen sometimes.

3. Project can be accessed through browser on mobile device but mobile application doesn't exists till now.

4. As project is still in its infantry mode, deployment is done to check its compatibility only, more than 4 users writing/editing code on the same task created by host may cause website to crash.

2.2.2 Other Limitations :

As the login/ sign-up is done using Google/ Facebook, user may have to wait for some time for the conformation from the their servers as they block any spam mails registration. Same can be said true while creating a new project and adding GitHub repository link.

UI can be made more interactive using Flutter and Bootstrap.

2.3 Discussion and future direction

As we know that host can monitor the clients when they are coding and interacting, he/she can also block any user if the user acts suspicious or caught coping. Still using new tricks student can copy using other devices as well, for this we can implement Deep Learning model for identifying if the student is paying attention at the class by tracking his/her body and iris movement, also we can implement object tracker to track as if any other device comes in video frame it alerts the host.

Addition of virtual board, creation of virtual classrooms, attaching calendar, result showing etc, can make our project more interactive and accessible.

2.4 Concluding Remarks

The methodology of the project was made transparent in accordance with the aforementioned literature review. socket.io creation, database connection, linking with GitHub repository are all included. It can be used for a wide range of educational purposes. The intended system has the ability to identify various users and connect them for interacting and collaborating while coding.

Chapter 3

Problem Definition and Scope

3.1 Problem statement

Collaborative Coding Platform for all Coding Lovers/Enthusiasts.

3.2 Goals and Objectives

1. To build an application for learners to get help while learning.

2. To build an application for company employees for building an project together.

3. To build an application for companies to hire candidates without letting them cheat or copy over the internet and be valuable or value added asset for the organization.

4. To monitor and giving the instructions while coding by other team member.

5. If any error can occur while coding then that might get resolved whoever will see the first while working on project at a time.

3.3 Scope and Major Constraints

3.3.1 Project Management

Working together on same platform in a team provides better visibility, keep the deadlines and project on track, everyone gets an clear idea about project.

3.3.2 Learning

While teaching online, teacher can teach everyone at a time but when it comes to monitor them it can't happen so at a beginning level many doubts remain unsolved if student is shy so while collaborative coding when student will perform the teacher can detect it and revise the topic and give a clear about in brief to student.

3.3.3 Interviews get plagiarism free

As we have stated earlier, many candidates do copy from the other sources during interview so the collaborative platform can help if it will be attach with camera and also screen monitoring with disabling the keys of coping and pasting. or even interview can monitor if it is an one to one interview so it can't be happening.

3.3.4 Quality of work

When a team of members come together to code so discussing with each other they can code but with coding they might face the error and bugs in it, but in group if anyone see them they can debug it in meantime so in deadline we can create a quality of work together.

3.4 Hardware and Software Requirements

3.4.1 Hardware Requirements

1. A Laptop/Personal Computer

3.4.2 Software Requirements

1. Visual Studio Code (VS-Code)

2. Programming Languages:- HTML, CSS, Javascript, Node.js and Express.js

3. Database:- MongoDB

4. JSON server

5. A variety of web browsers for compatibility testing

6. Salesforce driven "Heroku" Cloud for deployment of the website.

3.5 Expected Outcomes

The expected outcome of "Collaborative Coding Platform" is as per name group of people can collaborate with each other to code and produce the quality of work. Even learning is also a big outcome of the project, as teachers can help student if found not not-understood the topic while implementing the concept. Additionally, the Interviews can get plagiarism free as we can monitor them with camera and listen the voice with microphone.

Chapter 4

System Requirement Specification

4.1 Overall Description

4.1.1 Product Perspective

The platform is a new collaborative coding platform where we all can code together. At the first user have to login here then as per requirement of user they can create the file if any company employee is user then there must be project or if any student is an user then there must be a IDE file for learning purpose. He can invite others to join him and as others join him they can start working together but if they want to communicate then they can communicate with each other on chat room provided there and according to disscussion they can code together and fix bugs of each other. In case of Interview scenario, the interviewer can create the file or project and give problem statement to candidate and monitor him by using camera and microphone.

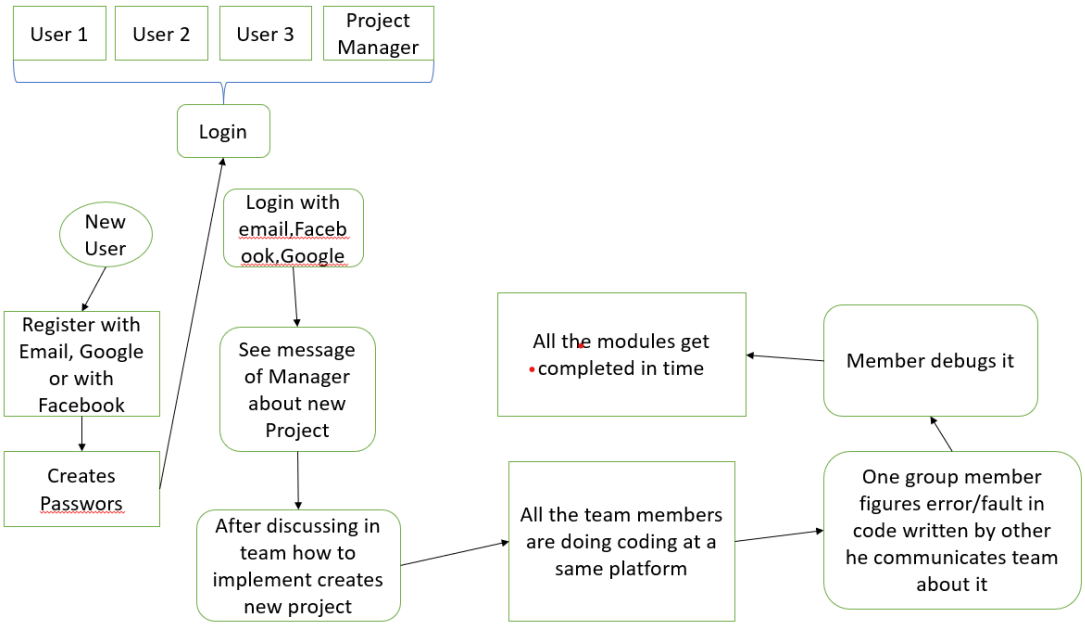


Figure 4.1: Figure 4.1 : Block Diagram

4.1.2 Product Function

1. To register the user, process the data provided, and store it in the Database.
2. To authenticate user password with already stored password.
3. To send the invitation to other verified user on the selected room.
4. To generate a project and verify it if it's an second time opening of file.
5. To let the user change information after login.
6. To allow the user to forget and regenerate the password.
7. To allow users to set primary email or phone among many options.

4.1.3 User Characteristics

1. Create a profile for the user.
2. Update password as needed.
3. Update previous projects/files as required.

4. Update user profile.

4.2 Specific Requirements

4.2.1 User Requirements

1. User should be able to solve create project/file.

2. User should be able to communicate with other users.

3. User should be able to monitor on code written by other and the work progress.

4.2.2 External Interface Requirements

1. Register: A form-type interface to get information from the user and save it in the database.

2. Login: A portal to allow users to enter and verify the login details with google and Facebook.

3. Edit profile: A page to change and view your profile on site.

4. Forgot password: A page to reset the user password.

4.2.3 Functional Requirements

[h]

Name	Sign-up

4.2.4 Performance Requirement

1. The load time for user interface screens shall take no longer than two seconds.

2. The login information shall be verified within two seconds.

3. The project should be created within three second.

4. The chatroom invitation should be sent within 4 seconds to other user.

5. Camera and microphone must work with laptop.

6. Site shall be able to work with 4 Mbps connection

4.3 Project Planning

This project was carried out in several stages. The following Gantt chart can be used to demonstrate the project stages and time frame. This above figure depicts the

	2021		2022			
	Aug - Oct	Nov - Dec	Jan - Mar	Apr - Jun	Jul - Sep	Oct - Dec
Literature Survey						
Finalization of Problem Statement						
Requirement Analysis, Feasibility and Scope						
Motivation, Objective						
UML Design						
Coding						
Testing and Correction						
Documentation						

Figure 4.2: Gantt Chart

Gantt chart of our project with it's timeline frame, as we have started out project in third year of academics in 2021 so it shows all the activities which we have performed from then like finalizing our Problem statement and carring out the literature survey, then searching for requirement analysis, scope of project then actually implementing the project and finally testinh=g and deploying it on cloud.

4.4 Modeling

4.4.1 UML Diagram

1. Use Case Diagram

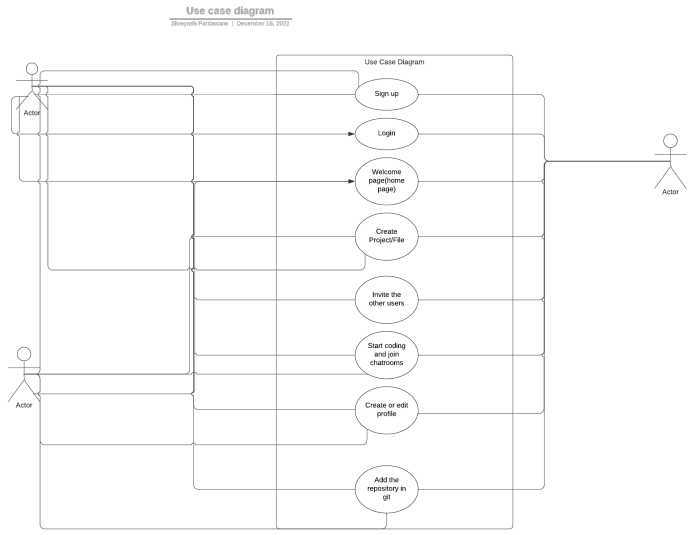


Figure 4.3: Use Case Diagram

In Use Case diagram, we have given the information about the case of Company where at the first employees have to login with their user credentials after login they have to create an project there they have to invite the other group members who are working with them.After accepting the invite they can continue to coding in collaborative environment. there they can have chat and also do codings and project manager can monitor the progress.

2. Architecture Diagram

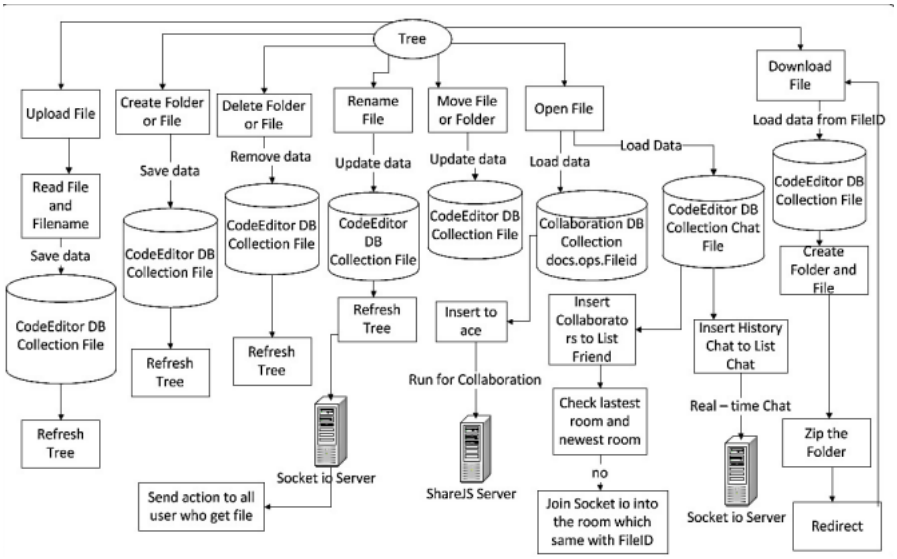


Figure 4.4: Login Page Architecture

This above figure shows the Login page architecture as here we are taking the user data from facebook. The first user wants to login with facebook account so server requests the details from facebook and facebook sends it to application server, then other user got invitation via facebook and he joined in the application. The user data is saved in database and all the activities done by user are also saved on database which we create on MongoDB.

Chapter 5

Methodology

5.1 System Architecture

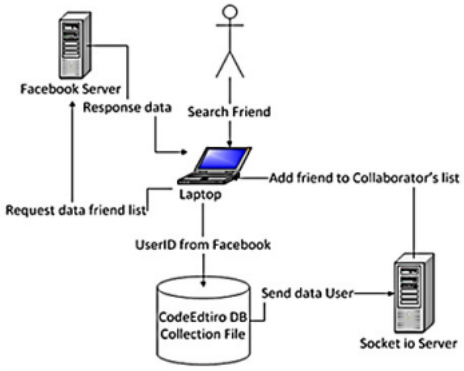


Figure 5.1: Add Collaborator Architecture

At the time of sign up the user has to set a username and password by filling up the required information and saving it in the database. At the time of logging in the user has to enter the predefined username and password. If the login details are valid then the user will be directed to the next authentication page else if the login credentials are invalid user has to retry to enter the details. The user can invite the other user by email only. When he will share the link of his room or send the request to join the room other member can't the room

5.2 Approach/Algorithms

5.2.1 User Registration

Our approach requires the user to register his details and store it in the database for future use. During registration, the user requires to be verified via email/phone.

5.2.2 Validating User

During the Login process user requires to be validate by username/email and password.

5.2.3 Inviting the other member

With the help of email or facebook we invite the other user in room where we can code and have conversation with other person.

5.2.4 Connecting the User

For connecting the different user we use the client server model here where socket.io becomes the communicator.

Chapter 6

Implementation

6.1 System Implementation

For the creation of project initially nodeJS environment is created. Later the implementation is done and executed on web application, where front-end is created using JAVA's popularly used development tool MERN stack development environment while the back-end is backed by community server of MongoDB using its MongoDB Compass application.

For real time data sharing between client and server Socket.io is used, whose functionalities such as 'join-room', 'chatMessages' 'disconnect' can easily be used by user at client side

6.2 User Interface

Figure 6.1: Landing Page

After entering the website user will be able to see this landing page

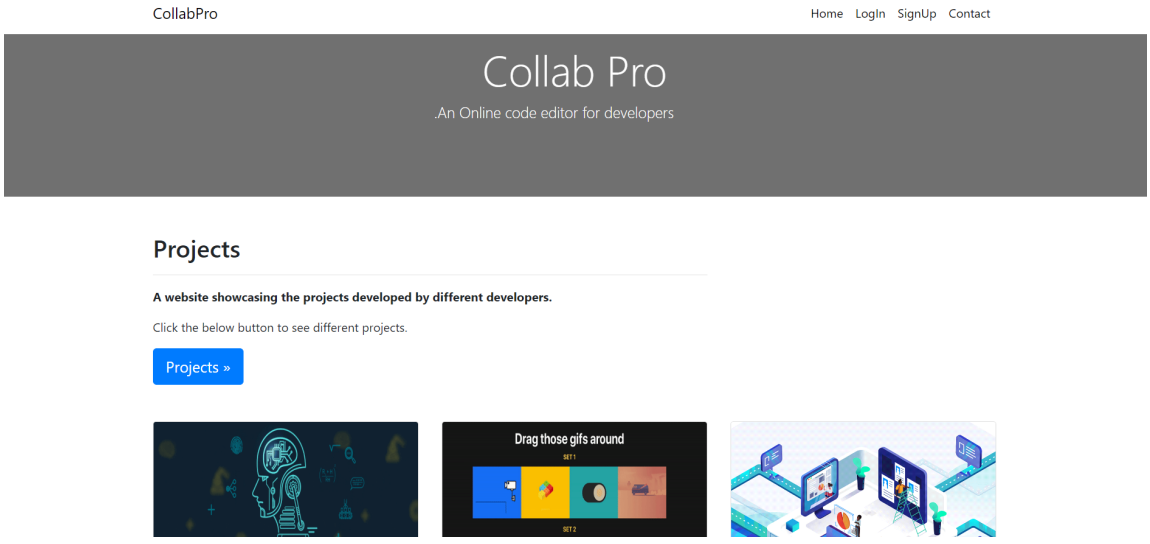


Figure 6.2: Home Page

On Home Page user will be able to see this UI and he/she can navigate to projects, Login/Sign up, Contact etc.

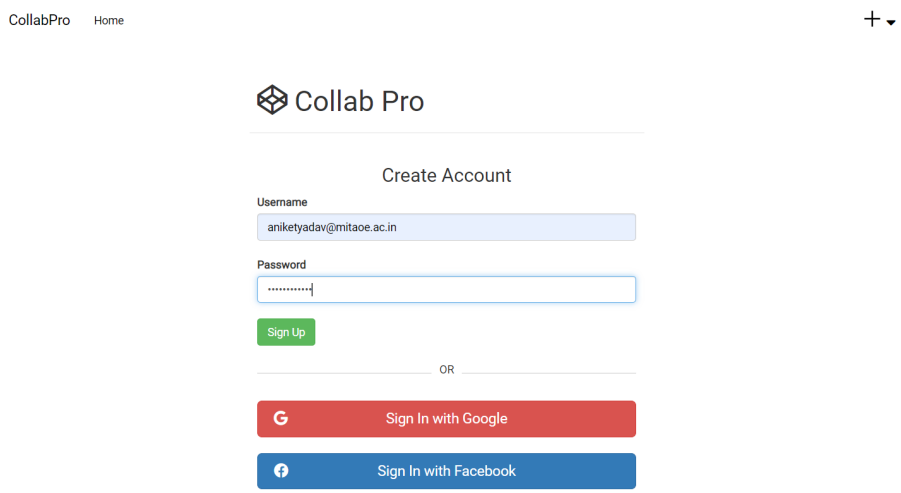


Figure 6.3: Sign-Up page

This UI for signUp will be visible to user

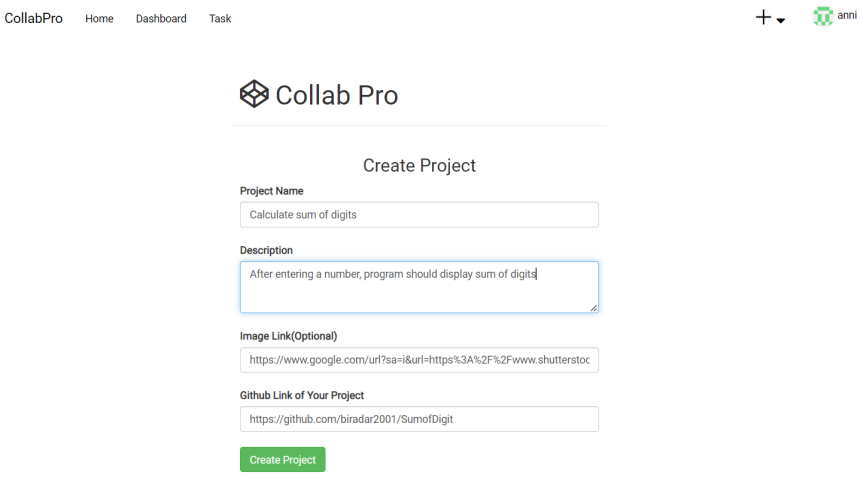


Figure 6.4: Creating new project

For creating a new project, user can add image's as well as Github repository link

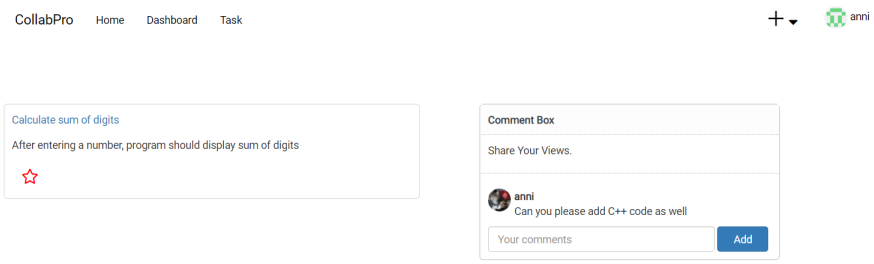


Figure 6.5: Project description

Here registered user can comment, star and access the existed projects on server

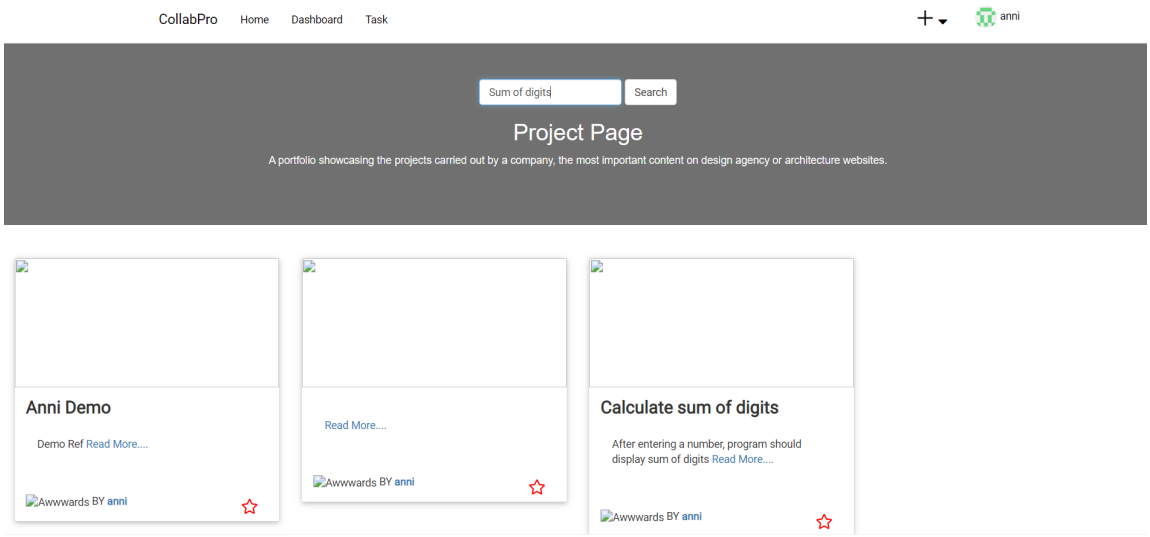


Figure 6.6: Project Page

UI for accessing all projects at once

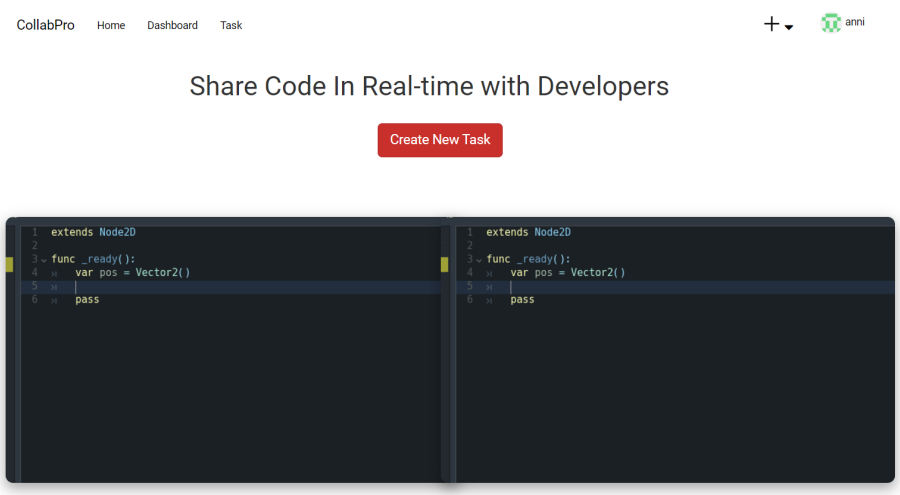


Figure 6.7: Creating a new task

New task creation page. Instructor can start new task and share the link with students to connect

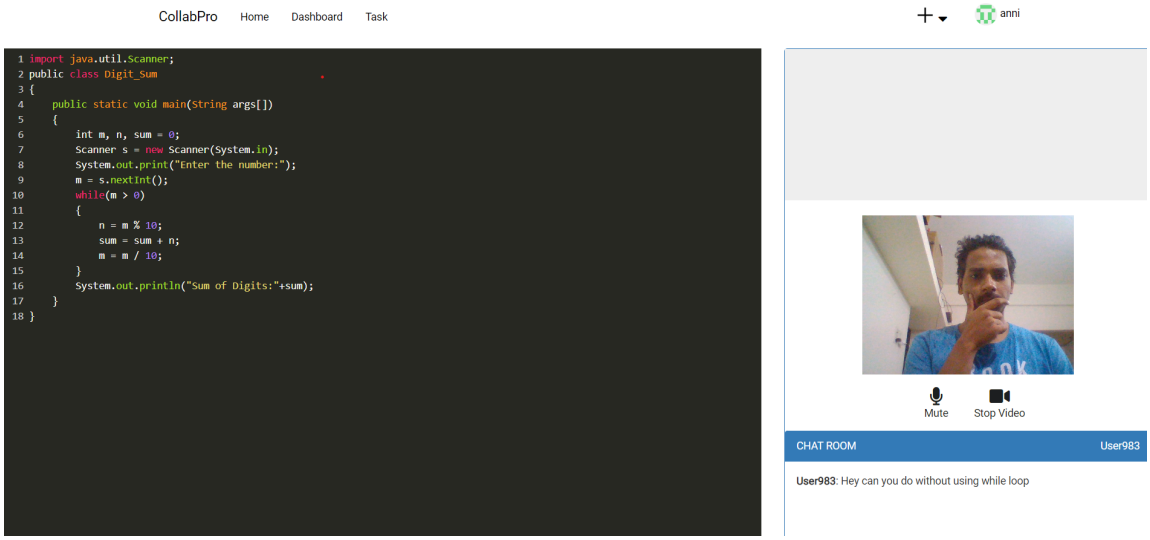


Figure 6.8: Real-time coding and chatting

This is main front-end where user will be spending maximum time, teacher-student can interact through mic and camera. Also chatting functionality is also provided to users

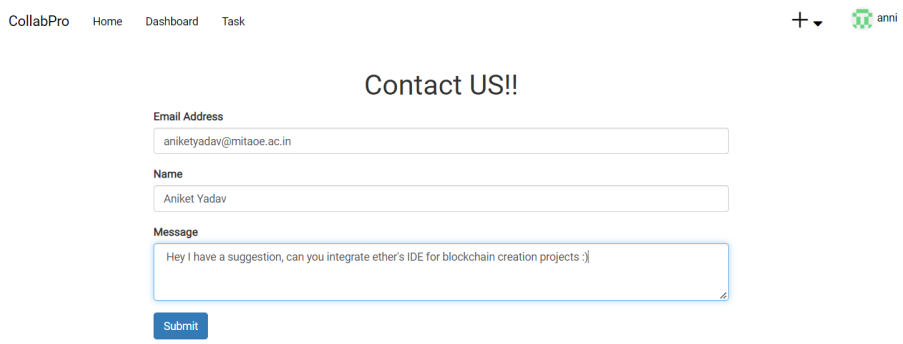


Figure 6.9: Contact Page

For inquiries and suggestion users can mail their quires through contact us page

6.3 Data Description

The project has following databases :

1. Comments
2. Profiles
3. Projects
4. Tasks
5. Users

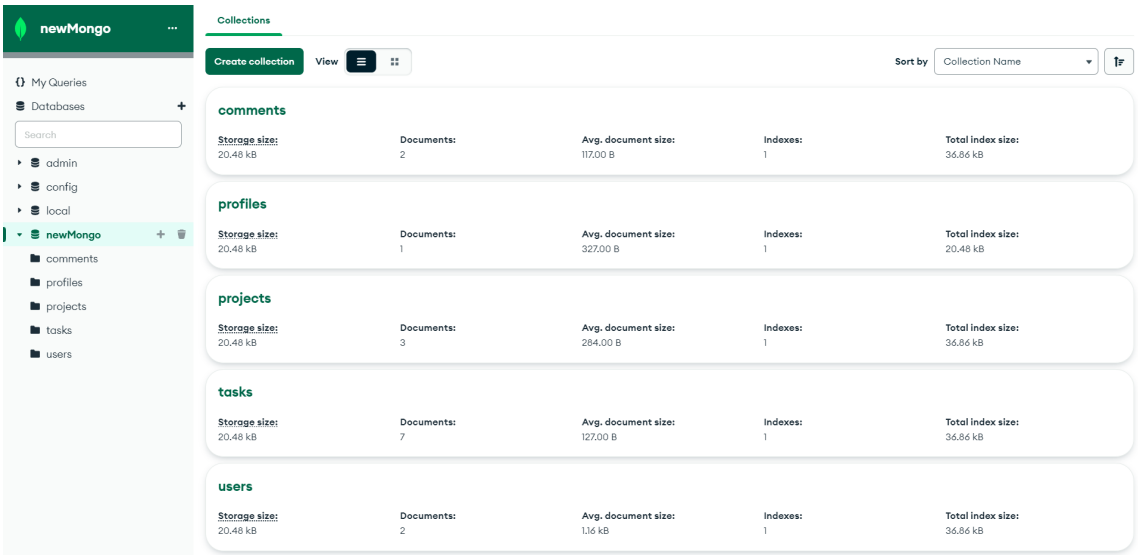


Figure 6.10: All databases created for the project

All the database created are shown above

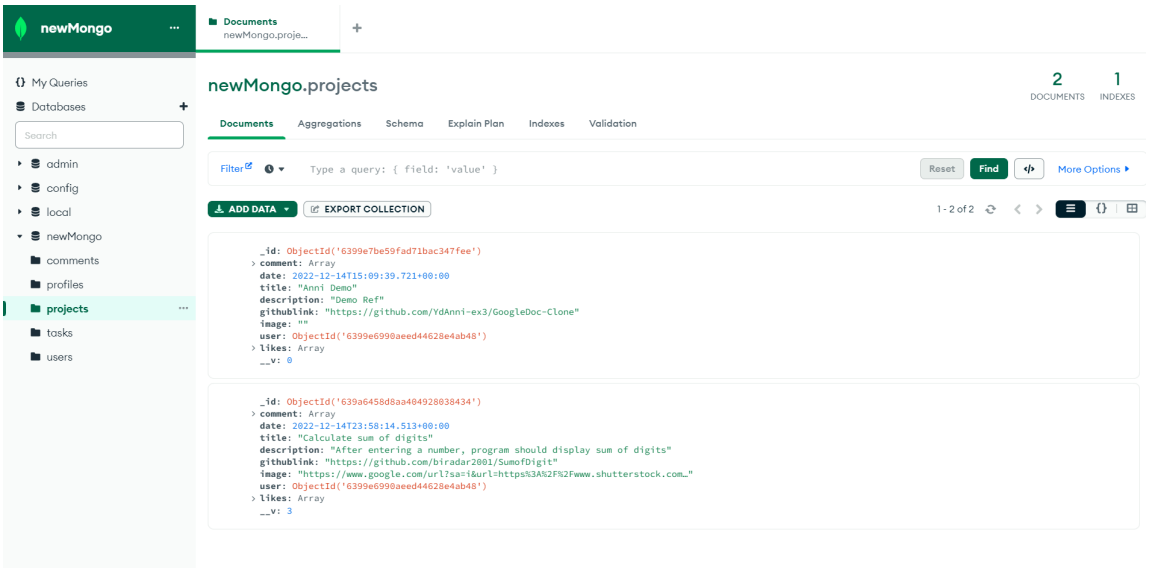


Figure 6.11: Projects database

Figure above shows various projects created by the users previously

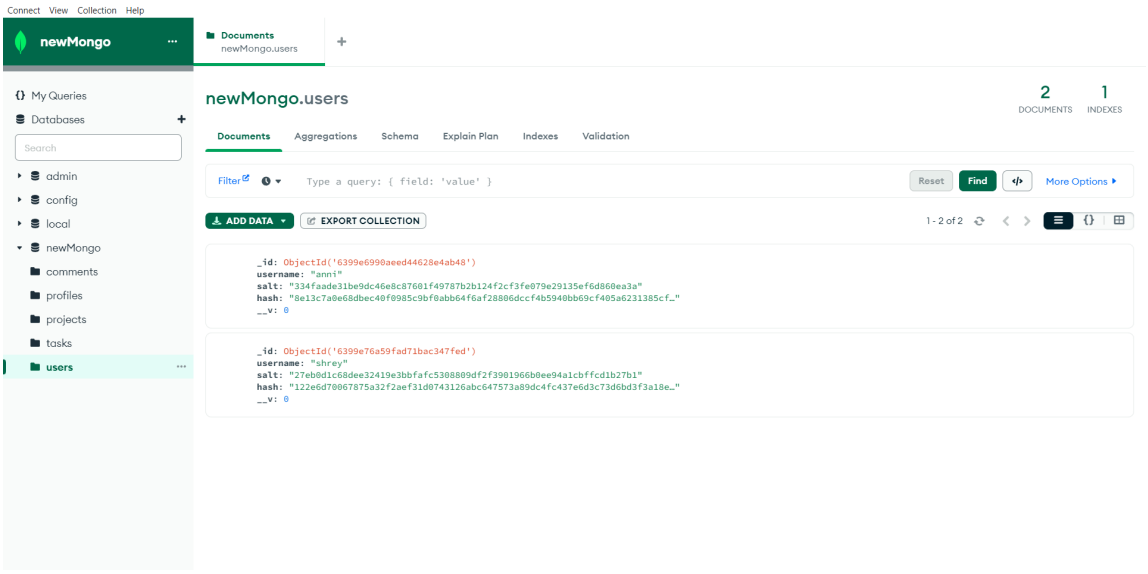


Figure 6.12: Users database

Above figure displays total users officially registered to access the project

6.4 Functional Implementation

Website involves following functions : 1. Registration

1. Login

2. Profile creation

4. New project creation / accessing existing projects

4. Creating new tasks

5. Commenting

6. Contact feature

6.5 Output

As shown in the figure various users/ hosts can talk, video call, code/ compile, chat without any delay on the website. Hosts/ interviewer can assign students task to complete on same platform as team. As a plus point, host can easily access the live video feed of students and get the information of how students are actually collaborating on the task provided.

Host have access to databases, he/she can easily see what changes have been made and when by whom, it provides transparency to the host for checking if the student is doing any malpractice and accordingly take action.

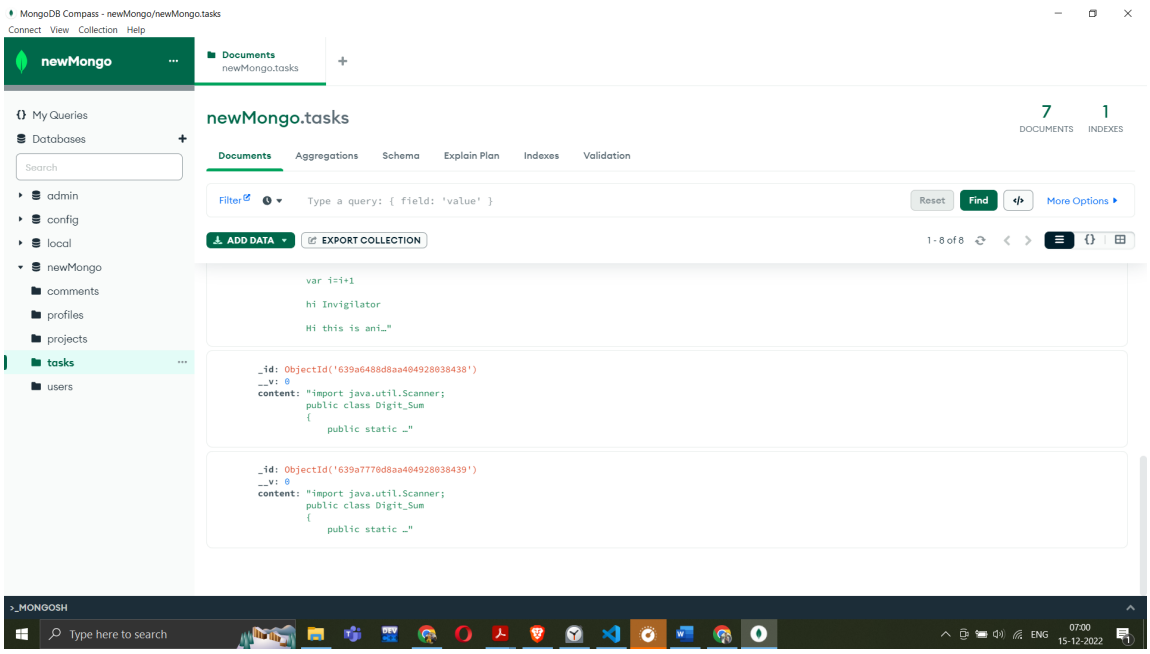


Figure 6.13: Tasks created by clients

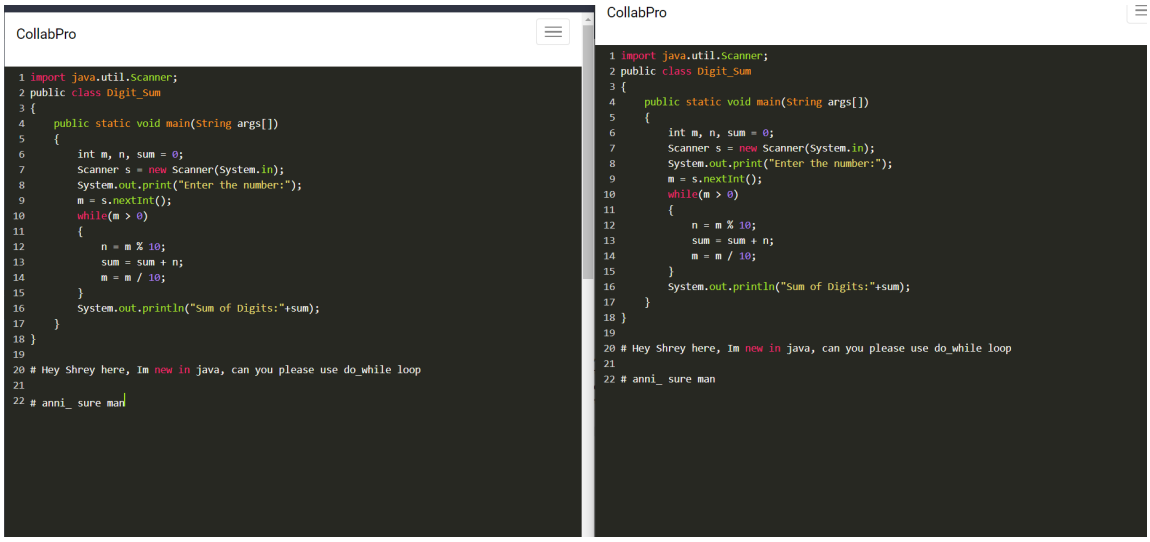


Figure 6.14: Real-Time code collaboration between host and students

6.6 Standard Industry practice adopted

There were some of the standard industry practiced adopted in creation of this collaborative coding platform

1. Use of MERN stack development environment:

i) MongoDB : It is used as data storage for backend, here we are using Mongo's community server for establishing connection with users data.

ii) Express JS : Being a open-source platform it is used as a back end web application framework for building RESTful APIs with Node.js

iii) React and Node JS : Being most popular open-source JavaScript frameworks, React is used for building user interface, while node is used to build server-side code

2. Socket.io : Used as a bii-directional communicator between client and server

3. Taking the Market survey for the apps that provide the intended services and taking the feedback from the users about apps and the scope of improvements.

4. Getting the problem statement from the above and deciding the goals and objective of the project.

5. Doing the literature reviews for the techniques available to solve the present problems and how their effectiveness and efficiency can be improved.

Chapter 7

Result Analysis/Performance Evaluation

7.1 Result Analysis

The performance of proposed objective to get help in learning is we can say achieve as users can connect each other

1. We can say that connection between hosts and clients was successfully established using socket.js.

2. Data processed in IDE (i.e, comments, profiles, projects, tasks, users) are stored in backend using MongoDB community server.

Chapter 8

Conclusion

8.1 Conclusion

1. Each collaborator will hold a copy of the shared document along with the server.

2. In case different users are updating different part of the same document, server applies the changes to its copy and forwards the changes to other collaborators.

3. Socket.io works properly for proper implementation of the project.

4. Security has been increased via using multi-digit roomID code.

8.2 Future Scope

As we know that host can monitor the clients when they are coding and interacting, he/she can also block any user if the user acts suspicious or caught coping. Still using new tricks student can copy using other devices as well, for this we can implement Deep Learning model for identifying if the student is paying attention at the class by tracking his/her body and iris movement, also we can implement object tracker to track as if any other device comes in video frame it alerts the host.

Addition of virtual board, creation of virtual classrooms, attaching calendar, result showing etc, can make our project more interactive and accessible.

This paper introduced our design and implementation of the real-time application for collaboration. CollabPro is a web application that helps programmers to create and see the result of the executed source code by terminal, collaborate in real-time with other programmers by chat or invite to join the same project and manage the project such as import, export, shared projects. CollabPro supports C, C ++ programming languages. CollabPro has the main features: provide workspace to make, execute and build the source code, real-time collaboration, chat, and build the terminal

Appendices

Appendix A

Code Snippets

A.1 app.js code

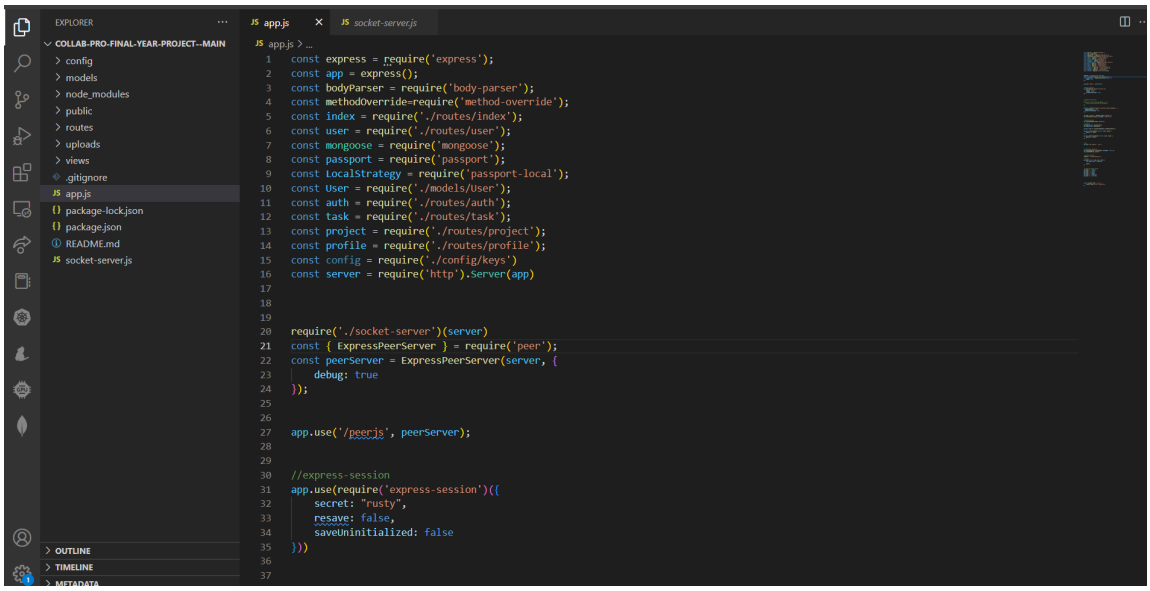


Figure A.1: app.js : 1

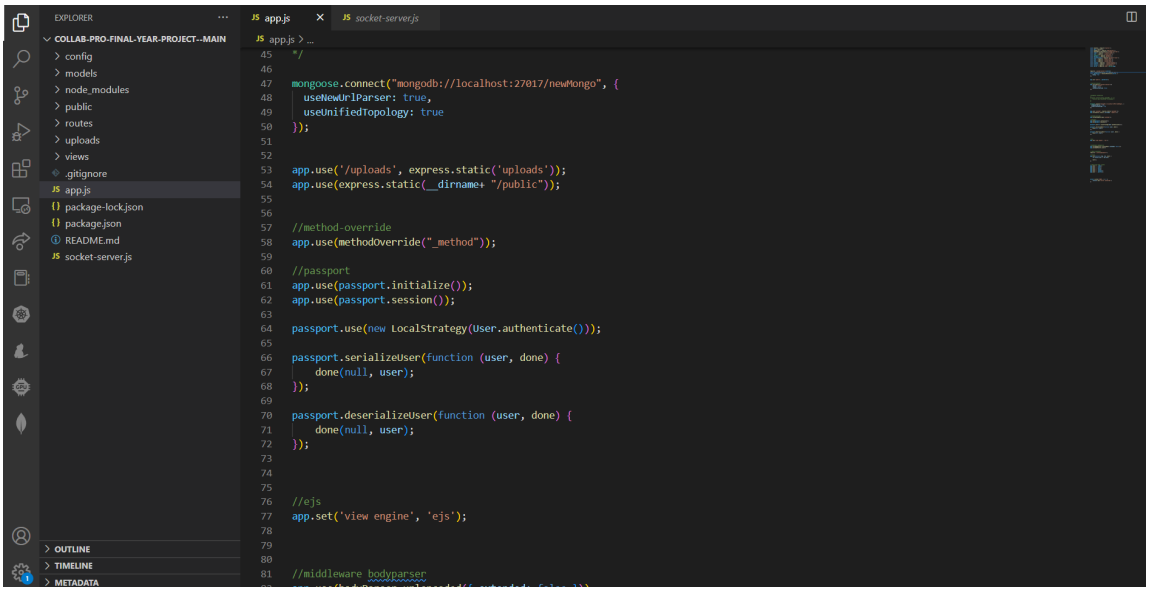


Figure A.2: app.js : 2

A.2 socket-server.js code

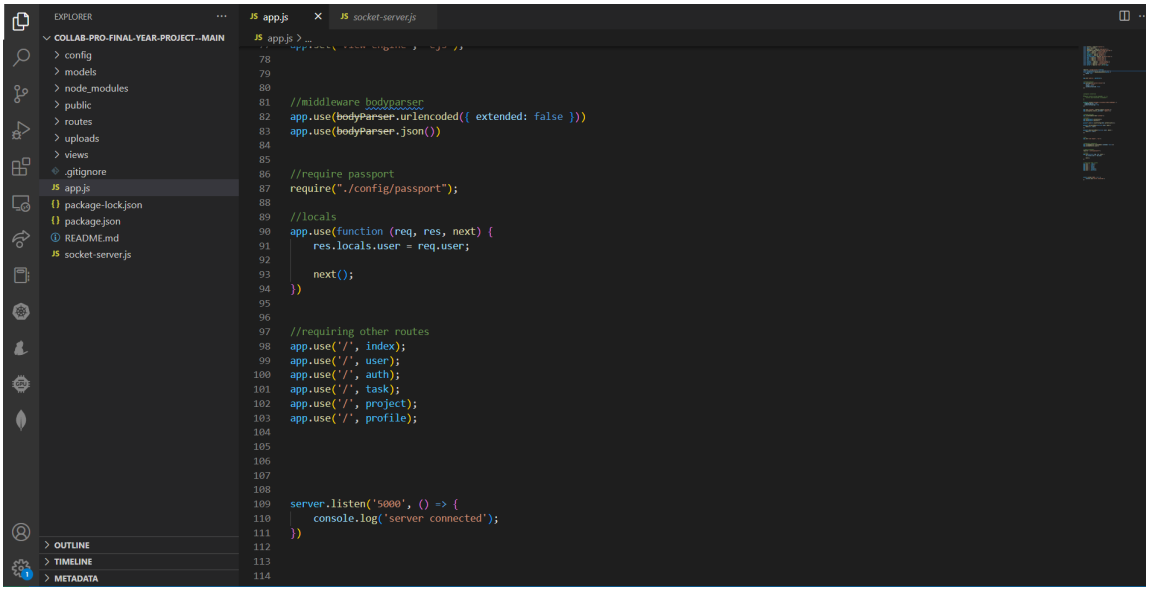


Figure A.3: app.js : 3

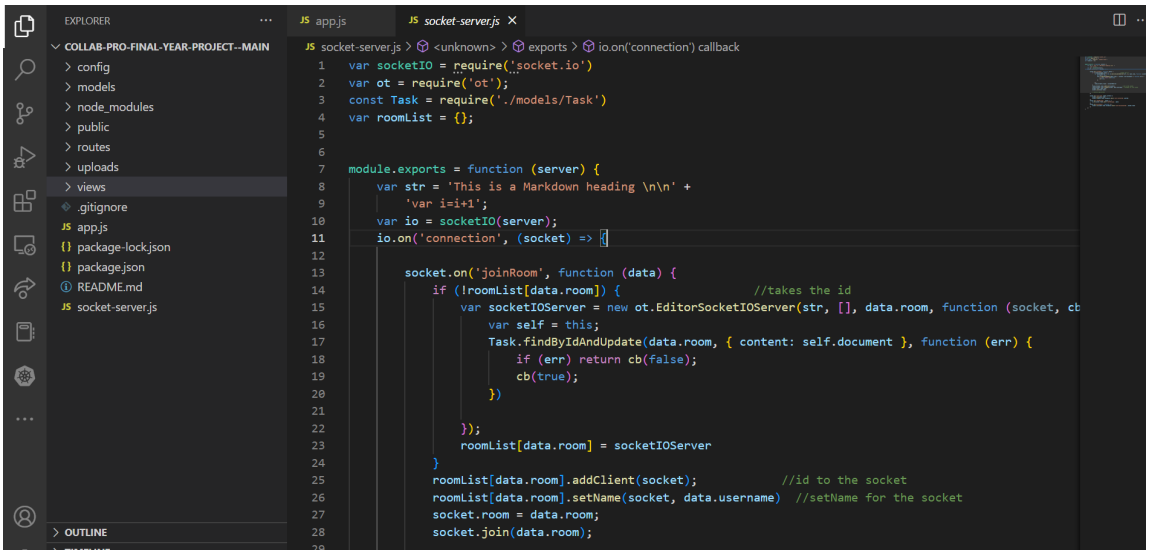


Figure A.4: socket-server.js : 1

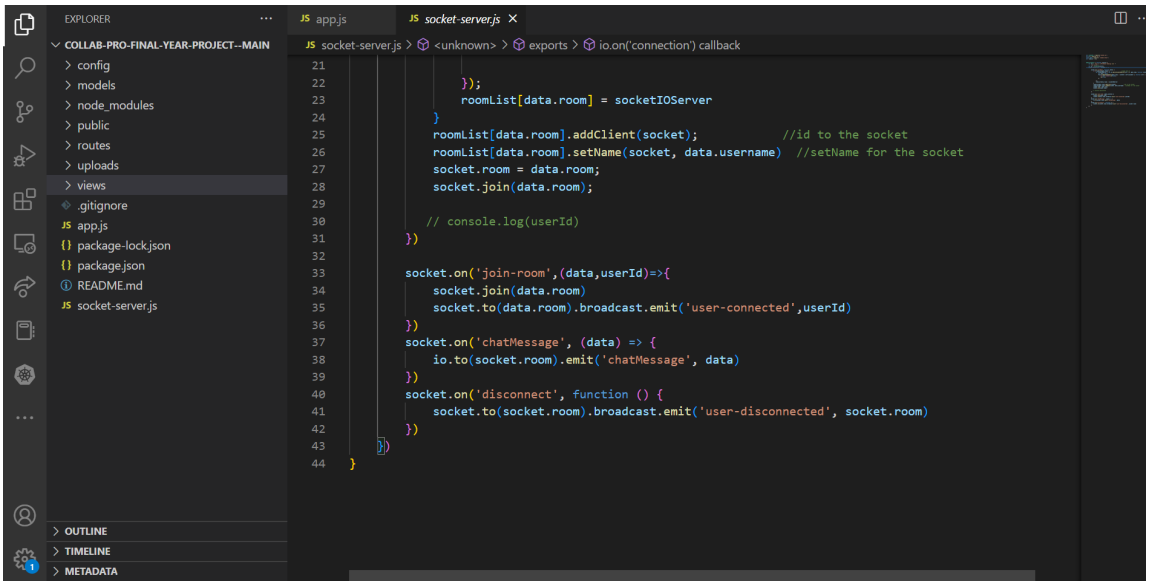


Figure A.5: socket-server.js : 2

Appendix B

Research Paper

Collaborative coding platform for all coding lovers

Aniket Yadav[#1], Shreyash Pantawane[#2],Bhagyashri Alhat[#3]

aniketyadav@mitaoe.ac.in
sapantawane@mitaoe.ac.in
bralhat@comp.maepune.ac.in

School of Computer Engineering and Technology
MIT ACADEMY OF ENGINEERING

I. ABSTRACT

Now-a-days, Coding is very important in part of academics of Computer Science aspirants or work life of Software Engineers or associate fields, to practice the code there are many platforms are present in market like Leetcode, Codeforces etc. but none of the platforms gives facility to code with others like having a team together to code or getting assistance in learning code in real-time virtual instance.

or even in during interviews of software companies many candidates got caught for cheating and implementing plagiarism code, to avoid that we must have some platforms which won't allow plagiarismic content or highlight the content to other person(interviewer).In this new era of Computer Technology and Automation there is an outburst in new webs and apps which help people in their betterment. As we can see there has been huge development in Technology in the last decade. As of now we all are recovering from COVID-19 pandemic, in this period we got to know the importance of Computer technology. To develop Career in this field people need to know about coding and for doing so in groups so they can get help right in there so they can grow widely, the collaborative coding and editing sites must be there. This project is to help this type of people and lead them to be a master coder and as all the project is associated with cloud services so it can be available anywhere and anytime for users with safe environment exposure.

keyword- Collaborative Coding, Covid-19, Cloud services.

II. INTRODUCTION

This paper is based on developing a collaborative coding platform where the interviewer will be able to share the link of online IDE (our projects deployment) where attendees will write code, compile and debug[2]. This will also help interviewers to avoid cheating on technical interviews. It is based on Cloud so each being can use the platform anywhere and anytime. There are other scenarios like in a company a group of people have to work in a particular project and while implementing/working on the project each one of team member have to work together so they can collaborate while coding part together on same platform where can can code together or even project manager can have a watch on what employee is implementing and if there are changes so he can direct those in meantime[6]. There have been some practices/tries before also but they have

Figure B.1: Research Paper — Page : 1

some flaws in the system so this project tries to improvise them in simple manner.

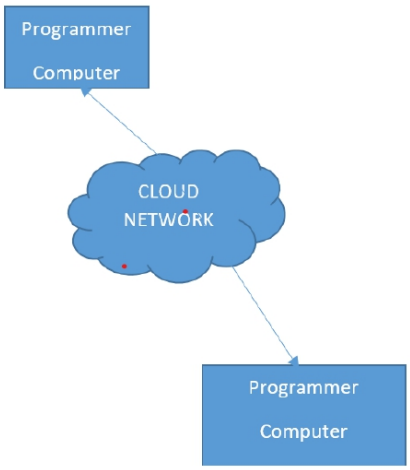


Fig 1 Basic Idea of Project

We have seen how easy it is to copy code and explain it like it was written by the individual itself. The same thing can occur at professional technical interviews as well, where a student can copy the code assigned to him/ her from the internet, run it on a personal IDE and submit it like it was coded by himself. To avoid this mis-conduct we got motivated to develop an online IDE, where the interviewer can monitor candidates in real-time while he/she is coding.

keyword:- Coding Platform, Company Projects.

III. LITERATURE SURVEY

The EarSketch and New Directions for Collaborative Live Coding in the Classroom [1] This paper is an initial step to fill the gap between collaboration. The EarSketch and New Directions for Collaborative Live Coding in the Classroom : rest API is built on top of JavaScript and Python so that EarSketch can be easy to used by students. Users can work either with their own uploaded sounds, or with 4000 available music samples that were engineer and tour DJ, which renders the tracks on the DAW view; and press the play button to listen to the result music.

Using GITHUB in software engineering course: Analysis of students acceptance of collaborative coding platform**:** [2]

Here the experiment course is performed in the last semester of the bachelor's degree study program and it builds on the knowledge that students acquired through the first five semesters. By adding software engineering theory and practices on top of analysis and specification, design, development, testing, deployment and documentation divided into up-to-five-member teams. Each team member was responsible for performing all aforementioned activities on four (plus/minus one) functional use cases (i.e. software tester and operations manager.

Vivace: a collaborative live coding language and platform [3] Vivace enables the emergence of the performance and makes it a kind of a collective game, where the rules, being visible to everyone through the code, eases audiences and specialists alike to join in. Live coding becomes a natural path to the type of use and technological development in which freak coders are involved

Coding together - Coding alone[6] The role of trust in collaborative coding :

Figure B.2: Research Paper — Page : 2

In this digital world, where we all want to grow fast and safer in a small duration. Collaborative Coding is a thing for them but while performing this technique/process all the team members must believe in each other. If any two members are not having trust in each other, the project won't be complete and for such people this process won't work well. To overcome this, first you have to make sure what there won't be trust issues and each module and each unit is discussed very well before implementing.

V. EXISTING SYSTEMS

Weblinux: It is a web app tool providing a standard Linux OS and an IDE in the brows²er, including a terminal, a code editor and a file browser. It provides a client side and offline Linux OS environment based on a JavaScript emulated processor.

REPL base GoTTY: It is read–eval–print loop or language shell based approach in which the user is provided with an interactive computer programming environment (IDE) that takes single user inputs, executes, and returns the result to the user.

VI. Here the first Interviewer will register on the project's website for organizing and scheduling the interview, then only a link will be generated for the same.

Now on the scheduled time the Interviewer will ask the student to add to the same link for the interview.

After joining and agreeing to all agreements, the person giving the interview will be locked on the same platform.

Now it's up to the Interviewer to start the interview. He/She can access the features of tool such as blank document, paint sheet or ask the person to write the code on IDE attached on platform.

In this whole process data will be encrypted and stored on MongoDB for later use. Platform's frontend is developed via HTML, CSS, JavaScript using Angular Framework.

VII. COMPARISON BETWEEN PRESENT SYSTEMS

User testing, this approach is beneficial in two ways: communication, in addition to source-code editing. and tools for practitioners collaborating on data use to collaborate on data. Participants were recruited in two ways to reduce confusion and limit unsolicited emails, the institution's computer science department. students and faculty. These contacts were generally in the computer science (or related) department. separately. Second, the recruitment email was sent to well professional announcements and international institutions. perceptions of the aforementioned applications were administered online through Qualtrics. In total, there were 23 faculty responses . Most faculty respondents (18/23) reported collaborating in realt ime. laborations. with specific examples from the survey responses. transfer, and DevOps. We see a lot of overlap in the types of tools. Due to space constraints, some applications are not included in the table. Docs specifically. Two students did not mention it recently . difficult to set up. laborations were mostly done through pair programming. Students revealed a desire for additional support.

Figure B.3: Research Paper — Page : 3

point things out. Our previous research confirmed something similar.

METHODOLOGY

VIII. ARCHITECTURE OF SYSTEM

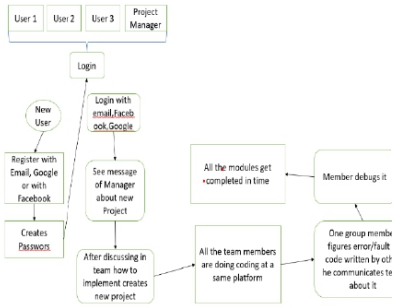


Fig. 1. Block Diagram

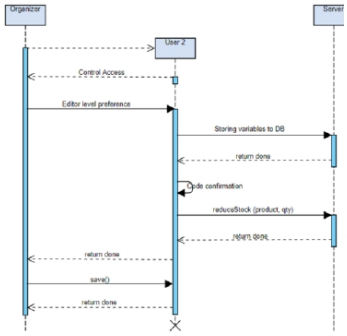


Fig.2 Sequence diagram

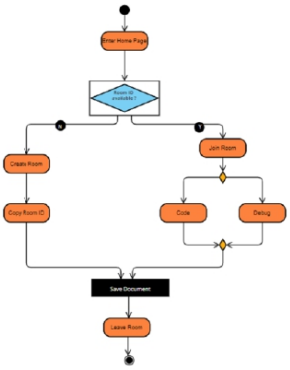


Fig.3 Activity Diagram

- For the Front-End part we had used HTML, CSS and JavaScript using Angular framework.
- In the project we had implemented MEAN stake over MERN.
- For Backend we used MongoDB.
- We implemented socket.io for connecting various nodes over local host

Keywords:- MEAN Stack, socket.io, MongoDB

IX. RESULT

- Connection was successfully established using socket.io
- Data processed in IDE is stored in backend using MongoDB.
- Deployment was successfully done at Heroku platform.

X. SCREENSHOTS OF IMPLEMENTED MODEL

Figure B.4: Research Paper — Page : 4

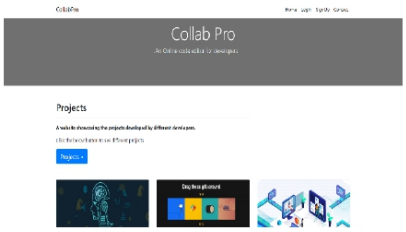


Fig 4 Home Page

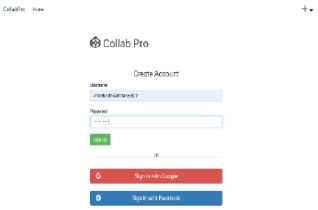


Fig 5 Sign in Page

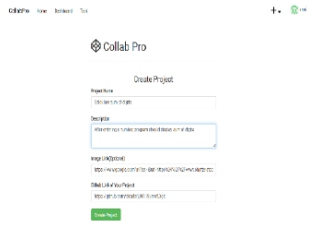


Fig 6 Create the Project

Fig 7 Create the Task in Project

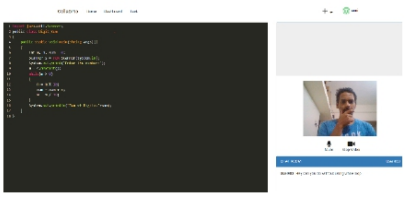


Fig 8 UI While coding

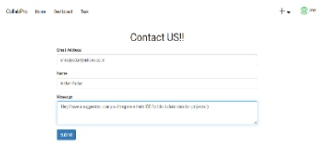


Fig 9 Contact Us Page

X. CONCLUSION

- Each collaborator will hold a copy of the shared document along with the server.
- In case different users are updating different part of the same document, server applies the changes to its copy and forwards the changes to other collaborators.
- Socket.io works properly for proper implementation of the project.
- Security has been increased via using multi-digit roomID code.

XII. REFERENCES

[1] Fiala J, Yee-King M, Grierson M. Collaborative coding interfaces on the Web. InProceedings of the International Conference on Live Interfaces 2016 Jun

Figure B.5: Research Paper — Page : 5

29 (pp. 49-57). REFRAME Books, University of Sussex.

[2] Naganathan, G., Srikanthan, S., Balachandran, A., Gladdy, A. and Shanmuganathan, V., 2022. Collaborative Zoom Coding—A Novel Approach to Qualitative Analysis. International Journal of Qualitative Methods, 21, p.16094069221075862

[3] Saini R, Mussbacher G. Towards Conflict-Free Collaborative Modelling using VS Code Extensions. In2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C) 2021 Oct 10 (pp. 35-44). IEEE.

[4] Xambó, A., Freeman, J., Magerko, B. and Shah, P., 2016. Challenges and new directions for collaborative live coding in the classroom. In International Conference of Live Interfaces (ICLI 2016). Brighton, UK.

[5] Sawant, S., 2021. Online Collaborative Learning Tools and Types: Their Key Role in Managing Classrooms Without Walls. In Human-Computer Interaction and Technology Integration in Modern Society (pp. 12-41). IGI Global.

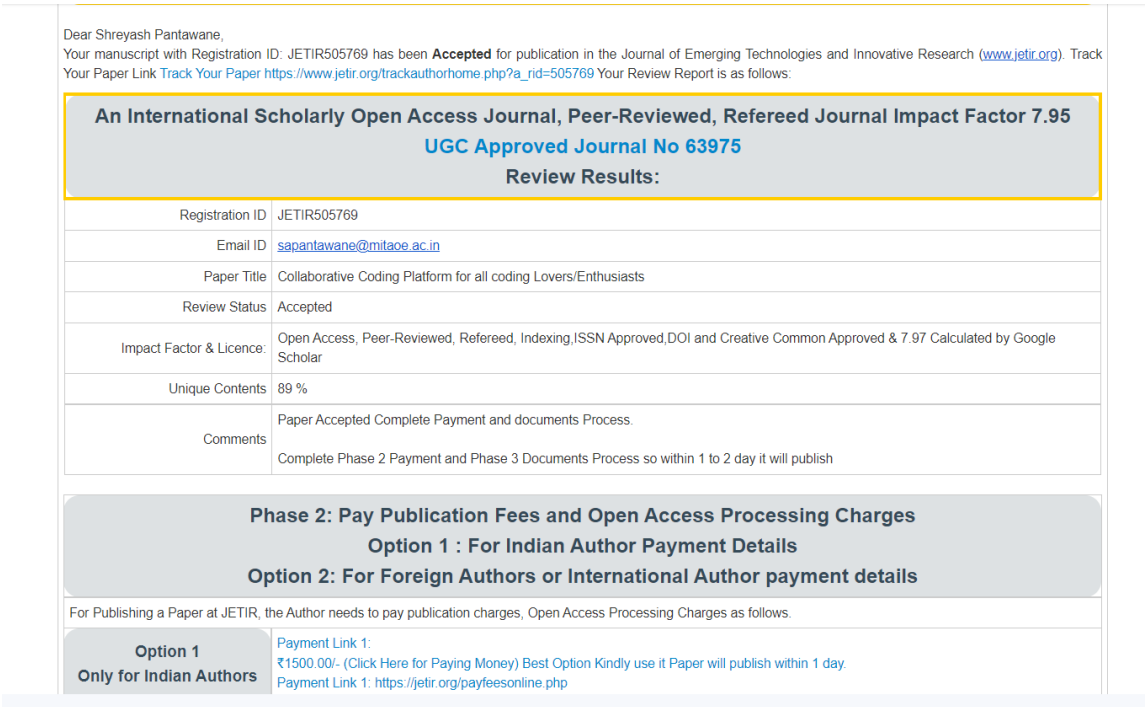
[6] Coding together – coding alone: the role of trust in collaborative programming Fabian Stephany, Fabian Braesemann. University of Oxford. August 24,2019.

Figure B.6: Research Paper — Page : 6

Appendix C

Publications/ Achievement

Certificate / Patent



Dear Shreyash Pantawane,
Your manuscript with Registration ID: JETIR505769 has been **Accepted** for publication in the Journal of Emerging Technologies and Innovative Research (www.jetir.org). Track Your Paper Link Track Your Paper https://www.jetir.org/trackauthorhome.php?a_rid=505769 Your Review Report is as follows:

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor 7.95
UGC Approved Journal No 63975
Review Results:

Registration ID	JETIR505769
Email ID	sapantawane@mitaoe.ac.in
Paper Title	Collaborative Coding Platform for all coding Lovers/Enthusiasts
Review Status	Accepted
Impact Factor & Licence:	Open Access, Peer-Reviewed, Refereed, Indexing,ISSN Approved,DOI and Creative Common Approved & 7.97 Calculated by Google Scholar
Unique Contents	89 %
Comments	Paper Accepted Complete Payment and documents Process. Complete Phase 2 Payment and Phase 3 Documents Process so within 1 to 2 day it will publish

Phase 2: Pay Publication Fees and Open Access Processing Charges
Option 1 : For Indian Author Payment Details
Option 2: For Foreign Authors or International Author payment details

For Publishing a Paper at JETIR, the Author needs to pay publication charges, Open Access Processing Charges as follows.

Option 1 **Only for Indian Authors**	Payment Link 1: ₹1500.00/- (Click Here for Paying Money) Best Option Kindly use it Paper will publish within 1 day. Payment Link 1: https://jetir.org/payfeesonline.php

Figure C.1: The Research Paper got Accepted Email screenshot

Appendix D

Plagiarism Report of Text

ORIGINALITY REPORT

14%	13%	1%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to MIT Academy of Engineering Student Paper	6%
2	libraryguides.vu.edu.au Internet Source	3%
3	www.coursehero.com Internet Source	1%
4	Submitted to Sogang University Student Paper	1%
5	www.slideshare.net Internet Source	1%
6	doku.pub Internet Source	<1%
7	Submitted to University of the Western Cape Student Paper	<1%
8	Submitted to Nottingham Trent University Student Paper	<1%
9	www.tpnsindia.org Internet Source	<1%

10	Submitted to R B Stall High School Student Paper	<1 %
11	www.firebase.com Internet Source	<1 %
12	Submitted to Athens Metropolitan College Student Paper	<1 %
13	qmro.qmul.ac.uk Internet Source	<1 %
14	repository.sustech.edu Internet Source	<1 %
15	staffwww.dcs.shef.ac.uk Internet Source	<1 %
16	www.cse.cuhk.edu.hk Internet Source	<1 %
17	www.irjet.net Internet Source	<1 %
18	zenodo.org Internet Source	<1 %
19	fedetd.mis.nsysu.edu.tw Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

References

Agiomyrgianakis, V., Svoronos-Kanavas, I., Zannos, I., & Diapoulis, G. (2021). Synerg(e)ia: A networked collaborative live coding environment. *iclc.toplap.org/2021/*, 5.

Fiala, J., Yee-King, M., & Grierson, M. (2016). Collaborative coding interfaces on the web. *research.gold.ac.uk*, 8.

J, F., M, Y.-K., & M, G. (2016). Collaborative coding interfaces on the web. *gold.ac.uk*, 9.

Naganathan, G., Srikanthan, S., Balachandran, A., Gladdy, A., & Shanmuganathan, V. (2022). Collaborative zoom coding—a novel approach to qualitative analysis. *journals.sagepub.com*, 8.

Parekh, B. V. (2019). Collaborative ide for e-classroom with progress tracking of students. *scholarworks.calstate.edu*, 42.

Saini, R., & Mussbacher, G. (2021). Towards conflict-free collaborative modelling using vs code extensions. *ieeexplore.ieee.org*, 15.

Sawant, S. (2021). Online collaborative learning tools and types: Their key role in managing classrooms without walls. *igi-global.com*, 30.

Thompson, A., & Wilson, E. (2021). Live coding the code: an environment for 'meta' live code performance. *qmro.qmul.ac.uk*, 10.

Vieira, V., Fabbri, R., & Fabbri, R. (2015). Vivace: a collaborative live coding language and platform. *researchgate.net*, 8.

Xambó, A. (2021). Virtual agents in live coding: A short review. *arxiv.org*, 16.

Xambó, A., Freeman1, J., Magerko, B., & Shah, P. (2016). Challenges and new directions for collaborative live coding in the classroom. *annaxambo.me*, 8.