IT254 : Web Technologies and Applications (minor)

Mini Project Proposal Report

VroCode : A social media for programmers

Submitted by

Abhishek Chavan 191MT001
Akshay Bistagond 191EC203
Diptesh Banerjee 191ME298

Under the Guidance of

Dr. Sowmya Kamath S and Ms. Thanmayee S

Dept. of Information Technology,

Date of Submission: 24/11/2021

Department of Information Technology
National Institute of Technology Karnataka, Surathkal.

2021-2022

Declaration

We **Abhishek Chavan, Akshay Bistagond, Diptesh Banerjee**, hereby declare that the project entitled **"VroCode: A social media for programmers"** was carried out by us during the **5th** term of the academic year 2021 – 2022. We declare that this is our original work and has been completed successfully according to the direction of our guide Prof **Dr. Sowmya Kamath S (NITK)** and as per the specifications of NITK Surathkal.

Place: NITK, Surathkal

Date: 24/11/2021

Abhishek Chavan

Akshay Bistagond

Diptesh Banerjee

Abstract

VroCode is a social media web application where our target users are programmers and developers. In this project we developed an application that can bring developers from different places together. In this website developers and programmers can find people with whom they can collaborate, interact, share developed programs and they also get the opportunity to showcase their skills and get better exposure in the respective community. This application can also act as a human resource for recruiters.

Since many developers tend to use their time in search for courses, projects, practice judges which results in less productivity and motivation. These issues are addressed in VroCode which are discussed ahead in detail. This application has authentication, for Login and register with Forget password as feature. VroCode also has a chat application to message any other user for their work and also it has an IDE to work practice (Currently: C++ and python). This application has almost all features of social media (Eg: Facebook, Instagram, twitter) and also acts like a place to access projects and clarify doubts (Eg: Stackoverflow). VroCode is a MERN stack project. To create the beautiful UI, we used the front end library React and Backend is handled by nodeJs, server by expressJs and we are using a non-relational database, MongoDB. Chat uses socket and IDE uses bull queue and redis for small operations.

Demo Video
Github Link

Contents

List of Figures

1 Introduction

1.1 Scope of the Work

From the market survey, we found out that there is no website that can cater to the appropriate needs of a developer. We observed that in Linkedin it is very difficult to share code, even though reach and collaboration are easy it becomes inconvenient for the developers to share links and information individually. On contrary, in GitHub, it's amazingly easy to share and present code but it is extremely difficult to reach out to developers, collaboration is secure and hence very difficult. Other social media apps like Facebook and Instagram are not at all appropriate for developers and programmers. There is no website that fulfills all the requirements and needs of a developer in his day-to-day life. The aim of our website is not to replace the mainstream websites like LinkedIn and GitHub but to make it easy for the developer to navigate between the websites with ease and to bring everything in one place.

VroCode is a social media app that is targeted towards the welfare of programmers and developers who can share, showcase or collaborate with others without much hustle and bustle. The basic features of the website are a profile, log-in and registration system, a ranking system based on contribution and expertise, navigation to numerous competitive coding websites so that users can improve their skill sets. A custom online IDE where they can test their code. A follower/following-based feed page where users can check out the posts of other people. People can comment on the posts, codes, and media can be shared in the posts. The post has a GitHub integration so that people can checkout GitHub code whenever liked by the poster. There is a personalized one-on-one chat system where people can chat with each other. They can access the private chat from the user profile.

1.1.1 Chat

The personal chat can be accessed in the profile of the user. when clicked on the chat icon, it creates a new conversation which has a conversationId in which senderId and receiverId and some arguments are stored. The convrsation information is saved in the mongoDB model. When the user clicks in the conversation, he is promped to send a message. For every message there is a messageId, senderId and some arguments which are stored in MongoDB. The messages are then rendered accordingly. The messages with senderId not equal to currentuserId are shown in the left while the visa versa is shown in the right. The chat application is shown in the figure:5.6.

1.1.2 Authentication

This is a basic authentication where a user can login using their username and password and a new user can register themselves by giving some basic details like username, email, name, password and role. The register takes the information from the register from and sends it to the server.On the server a new user is created with the details sent and is stored in the database. when the user attempts to login the username and password are sent and after confirming the credentials the user details are sent to the client and a context is created to maintain the login status. Passwords are saved in a hashed form on the database and verification is also done after hashing the given password. If a user forgets password then a random alphanumeric password is set for that user and that user is given that password to login and the user can change the password after logging in.

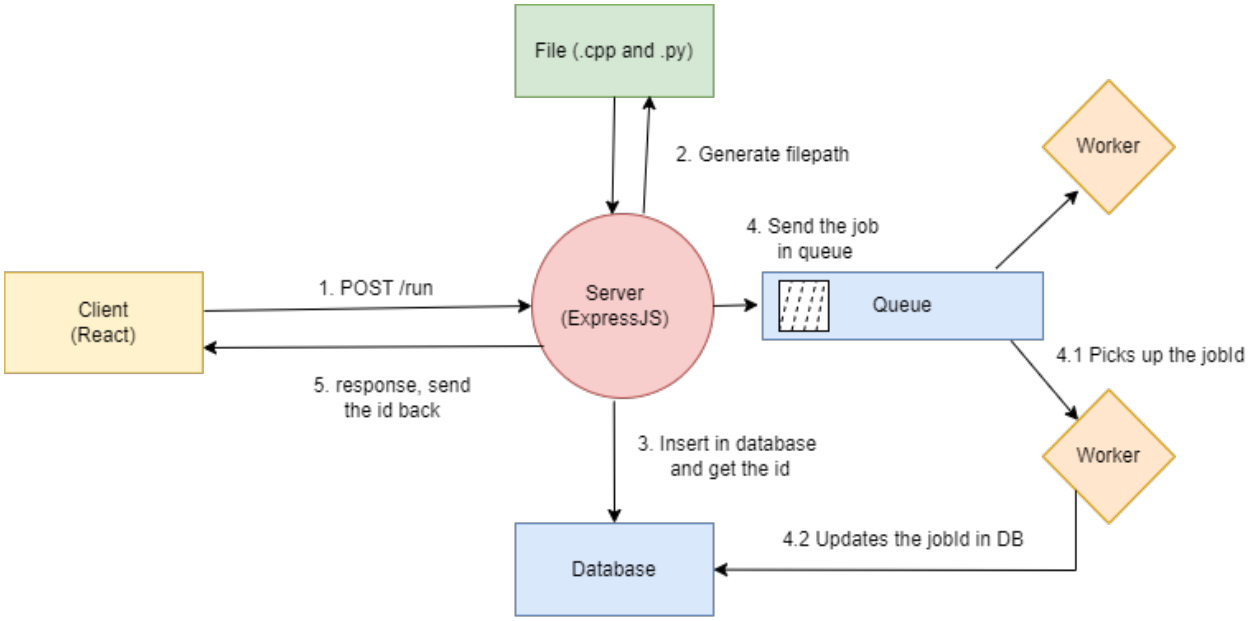
1.1.3 User Management

User session is maintained using the context API of react. All the changes to the user data was made to the context as well as a request to server so that the database will be updated and the client doesn't have to refetch data from the database to display the updated info. A user can follow others then connect with them. A user can also create posts , they can also like, save or share the posts. User will also be notified when someone else likes their post. A user can also edit their info using edit profile option. They can also upload a custom profile picture and a cover picture so that other users can easily recognise them. the pictures are saved on the server using "multer" middleware of nodeJs

1.1.4 IDE

VroCoder IDE is for the user who wish to practice and improve skills in particular language. Presently, we have created IDE for python and C++. This IDE is made using MERN stack with bull queue which uses Redis for saving and retrieving jobs which is requested from the back-end to the server. The job ids are sent to the database and filepath is accessed, these ids are pushed into the queues where the workers picks up the id and processes it with filepath. The id is updated in the database, such n-workers will be working to tackle concurrent users.

User can set a programming language as default, execute and check the output with the status and jobID. CPP is executed on local currently which uses g++ (mingw) which is executed by using shell commands and python is an interpreter which does not require such shell commands.



Future Scope This IDE can be used as online judge, where the user can give custom inputs and have some test cases. VroCoder can be upgraded to a coding playground with other languages. The entire execution part can be shifted on a server or a VM to execute the shell commands whenever the user requests for the output.

1.1.5 Post and feed

When a user creates a post the data is sent to the server and a new post is created and is saved to the database. and whenever a post needs to be rendered we fetch the post by id or the id of the user who posted it and then render the details sent from the server.
Users can share their GitHub repository which will redirect others on clicking it to their repository. This makes it easier to showcase the projects to the community or collaborate or get help from other users. Users can post an image along with the text or the code that they will be posting so as to give more details or to attract more attention to specific parts of codes/programs or to showcase their work better. the image is saved on the server side with the help of "multer" middleware of nodeJs. whenever a post is requested the image name is sent that is stored along with other post details. the client then renders the image with that name that is on the server.
Users can share the post by copying the url given in the alert modal (default), this can be used by non-users to view the content. This is done by fetching the post ID and routing it to new page where the other information is accessed.

1.1.6 Miscellaneous

Users can edit profile, logout and delete it as they wish to, this works by fetching the user route in the frontend and process as required to the functionality. They can also bookmark and like the posts. This is again done by using the end points set up in the backend. Currently, the rank based system works based on the number of login made by the user, for every 100 points an user ranks up. The maximum rank points one can get is 699, top 5 users are displayed in leaderboard where they are sorted in the backend, this is done by fetching the user details which has an object named rank. A randomized course suggestion on

left bar displays different courses on every reload varying from Web development to Data science. Coding practice and course page offer a lots of resources so that the user doesn't have to search them all the time decreasing the productivity rate.

1.2 Product Scenarios

1.2.1 For Developers

This application is very useful for developers. In this application developers can share and showcase their code and developed applications so that others can comment and recommend some improvements. The can chat with other fellow developers as shown in the figure:5.6.

1.2.2 For competitive Programmers

There is a custom IDE in our application which coders can use to practice code as shown in figure:5.9. Coders can also navigate directly to any competitive coding website and can learn to code in any of the courses website as shown in the figure:5.11 and figure:5.10 respectively.

1.2.3 For students

Similar to coders, students can use to practice code as shown in figure:5.9. Students can learn to code through various course sites by directly navigating to them as shown in figure:5.10.

2 Requirement Analysis

2.1 Functional Requirements

- Register/login providing Security.
- Personalization: User can choose the area of interest to be displayed.
- Navigational: Easy to navigate UI so that the user can navigate between different pages.
- Communication: The web application shall make it possible for the users to chat with each other.
- Extensive interactability: Will have code battle feature where users can compete against each other to win.

2.2 Non-functional Requirements

- Performance
- Reliability
- Maintainability
- Efficiency & Scalability
- Interoperability
- Usability, Portability & Compatibility
- UI : Easy to use
- Evolution

2.3 Use Case Scenarios

- For making connections with like minded people with common interests.
- To compete with fellow coders in code battles

2.4 Software Engineering Methodology

- **Agile Model** : Agile Model is a combination of the Iterative and incremental model. This model focuses more on flexibility while developing a product rather than on the requirement.

3 System Design

3.1 Design Goals

- To provide a platform where clients can share their projects.
- To provide a platform where clients can chat and connect with each other.
- To provide an online IDE so that client can run and test project code.
- To provide a platform where client can integrate all other websites like coding practice websites, GitHub, etc.

3.2 System Architecture

Model-Controller-View Architecture (MVC): It is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. As shown in the Figure:3.2.

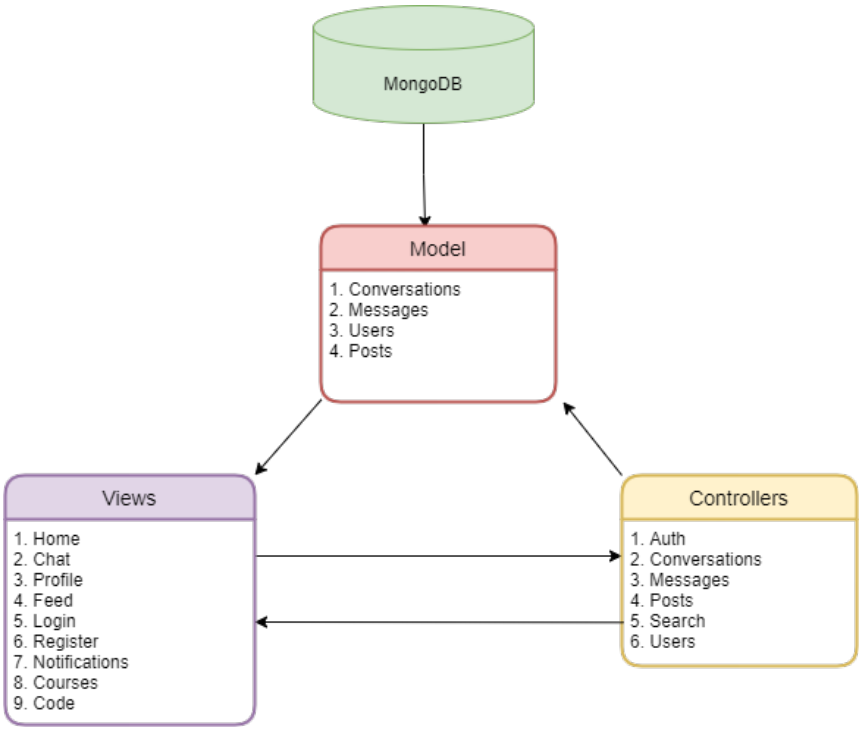


Figure 1: Flow Diagram of System Architecture

3.3 Detailed Design Methodologies (as applicable)

Systems Development Life Cycle (SDLC):

- Requirements Analysis – Establishes a high-level view of the intended project. It includes the analysis of end-user information needs.
- Design – Describes desired features & operations in detail, including screen layouts, mock-ups and style decisions.
- Implementation - Interactive components (Front-end) and Back-end.
- Testing – Stage to checks for errors, interoperability and usability.
- Evolution – Future opportunities are looked at, that then feed back into another cycle starting at an new requirements analysis.

4 Work Done

4.1 By Akshay Bistagond:

- Worked on frontend of some react components
- Set up the Backend express server and MongoDB server hosted on the MongoDB Atlas
- Set up various End points for serving the data from the database and to save data sent from the client to the server onto the mongoDB database along with images using multer middleware.
- Set up state management system using Context API of React
- Made various fetch requests to the server to make the webapp dynamic

4.2 By Abhishek Chavan:

- Set up backend and front end for the online Judge, which executes C++ and python languages using redis and bull queue
- Front end of the entire web application.
- Setup the leftbar, rightbar components with randomized course suggestions
- Code practice and courses page
- Set up rank based system
- Set up various End points for serving the data from the database and to save data sent from the client to the server onto the mongoDB database.(share post, giuthublink etc)

4.3 By Diptesh Banerjee:

- Worked on chat frontend and backend.
- Set up various end points for serving the data
- Set up the models for the mongoDB (Conversation and message)
- Added proper styling components in CSS with responsive design.
- In backend part, added socket.io as websocket API.

5 Results and Discussion

5.1 Login

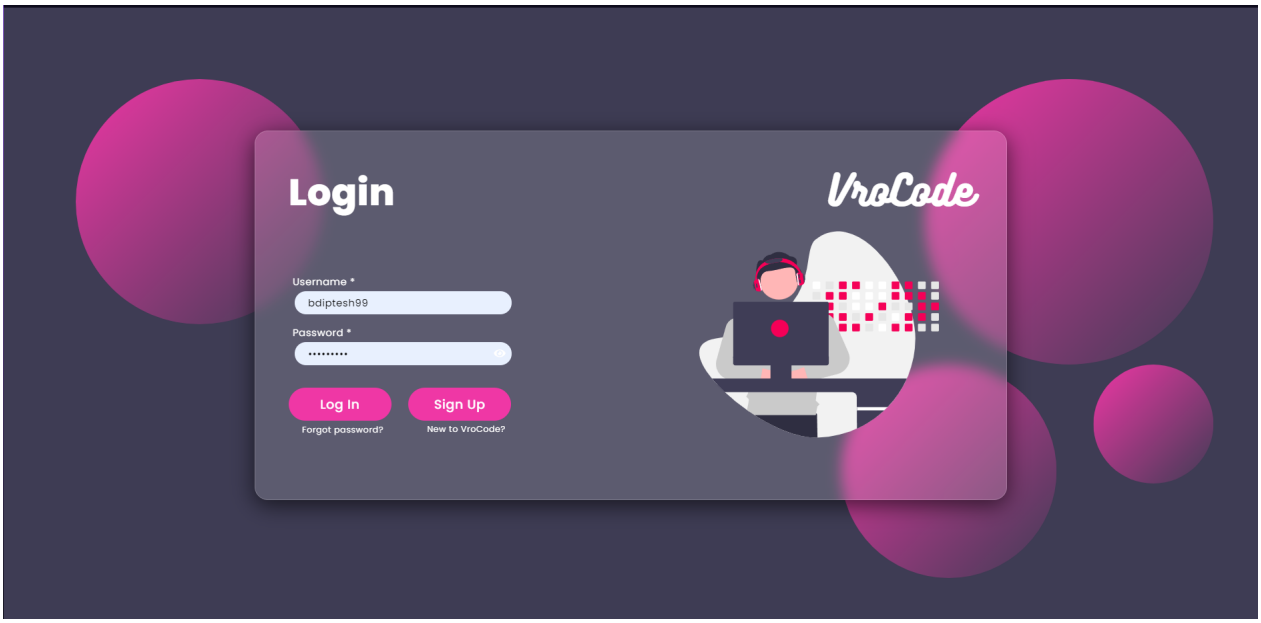


Figure 2: Login

5.2 Forgot Password

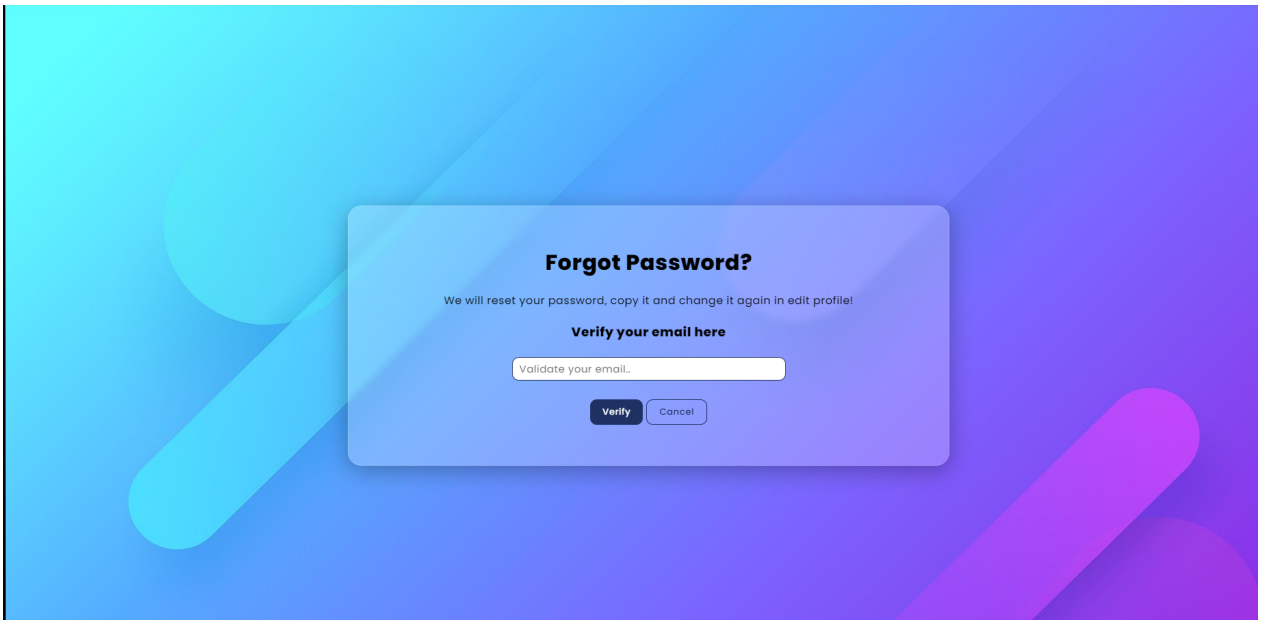


Figure 3: Forgot Password

5.3 Register

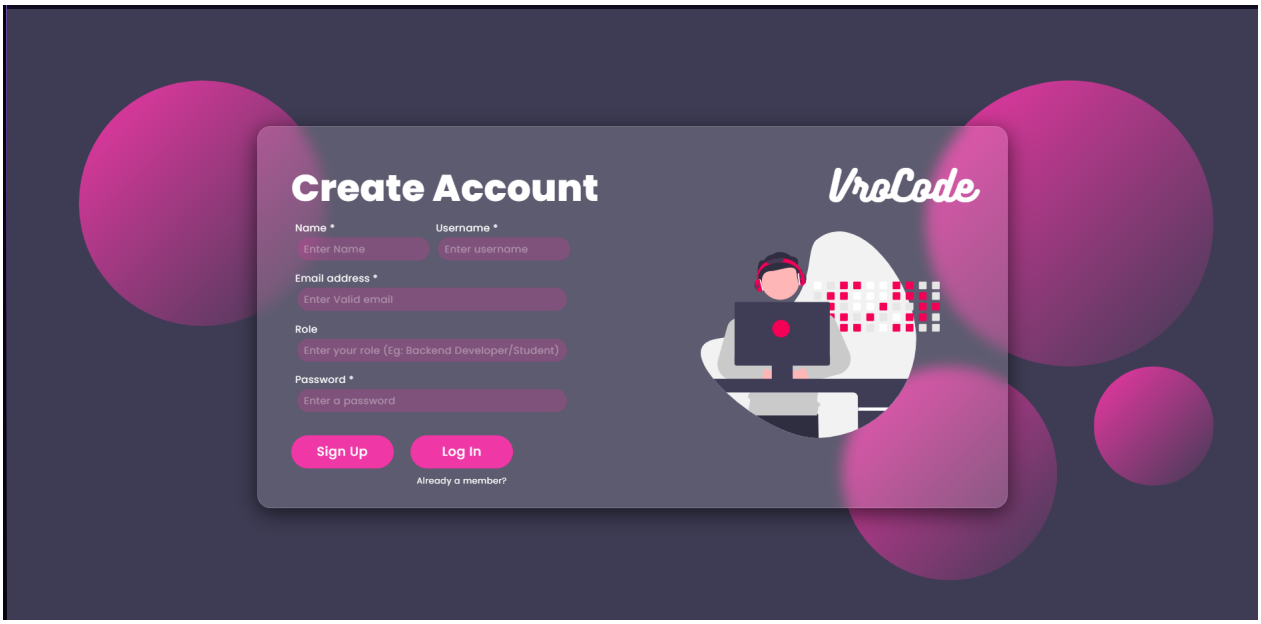


Figure 4: Register

5.4 Profile

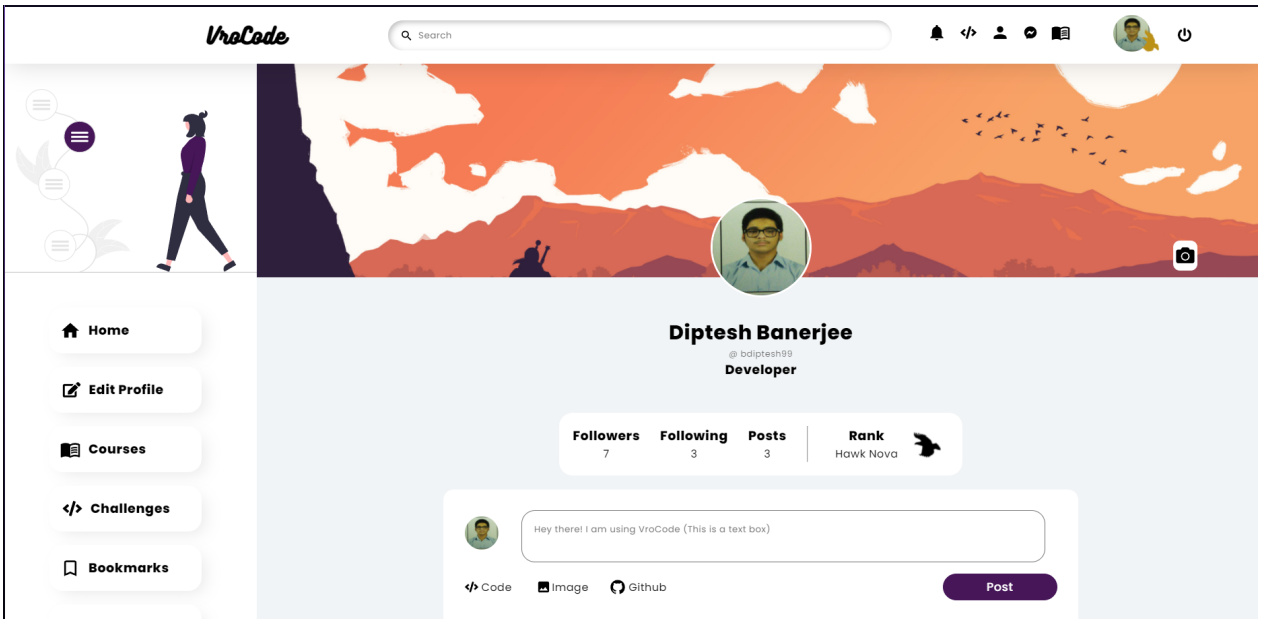


Figure 5: profile

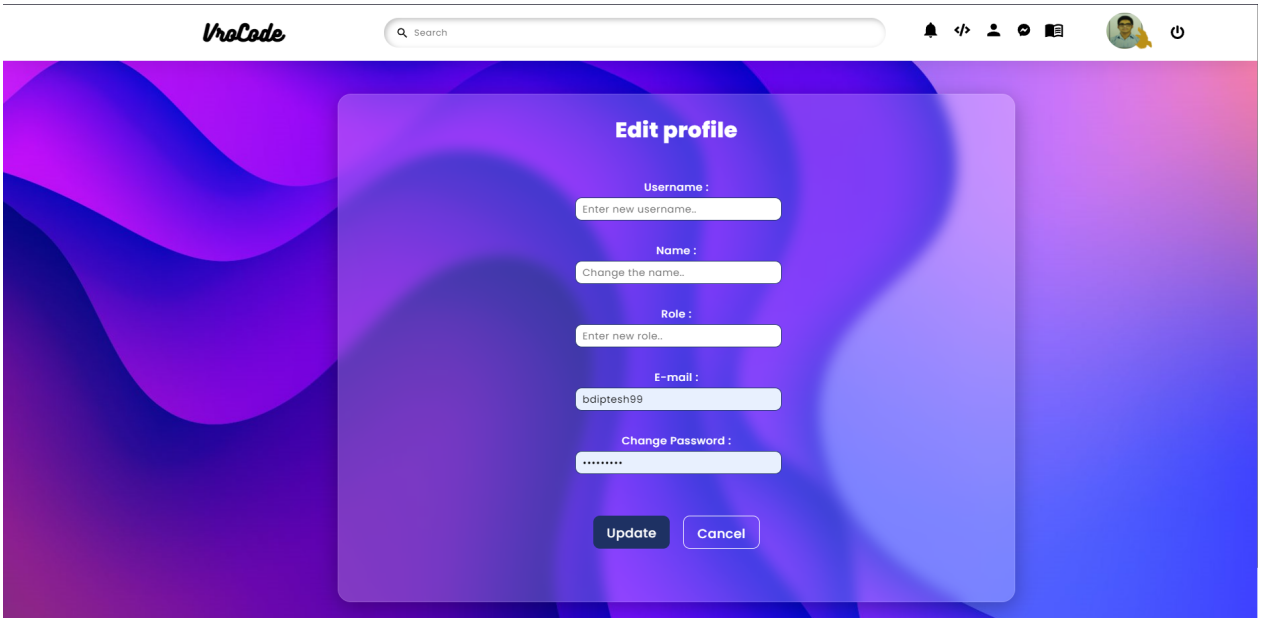


Figure 6: Edit Profile

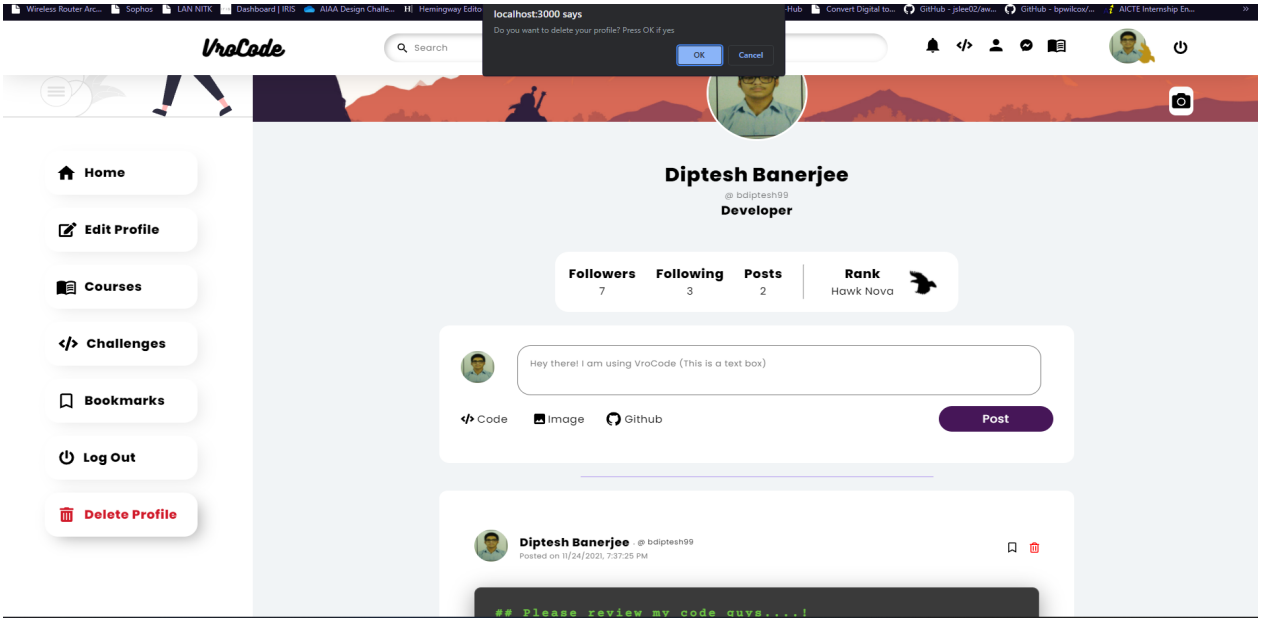


Figure 7: Delete profile

5.5 Following and followers

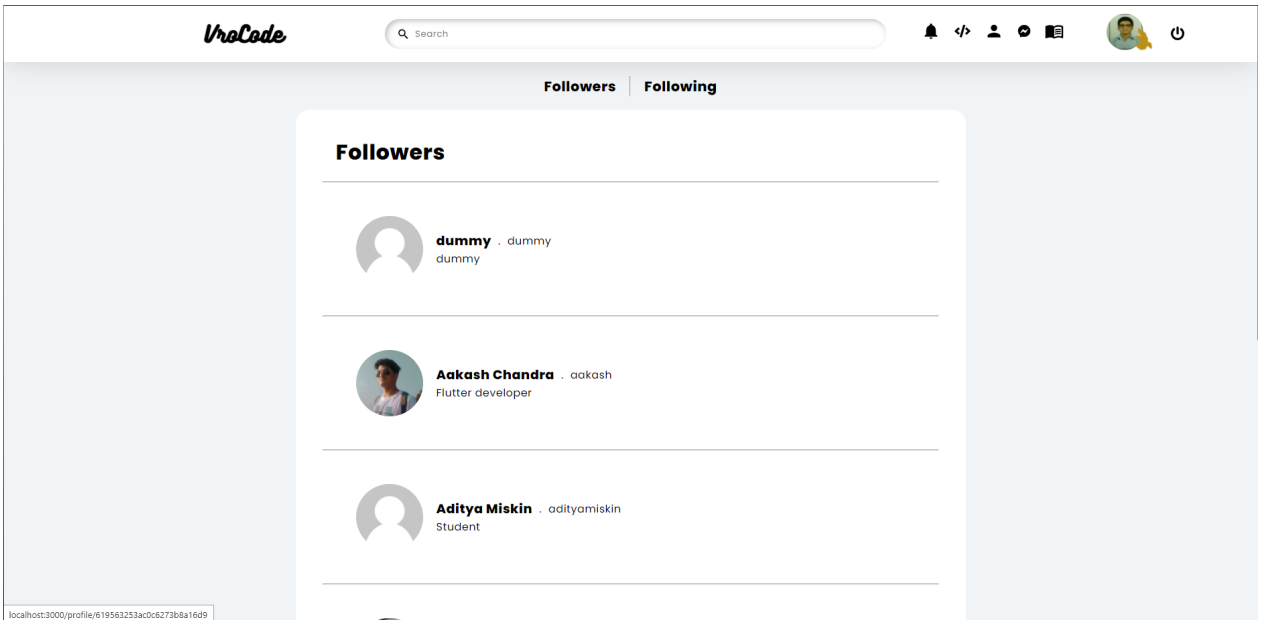


Figure 8: Followers

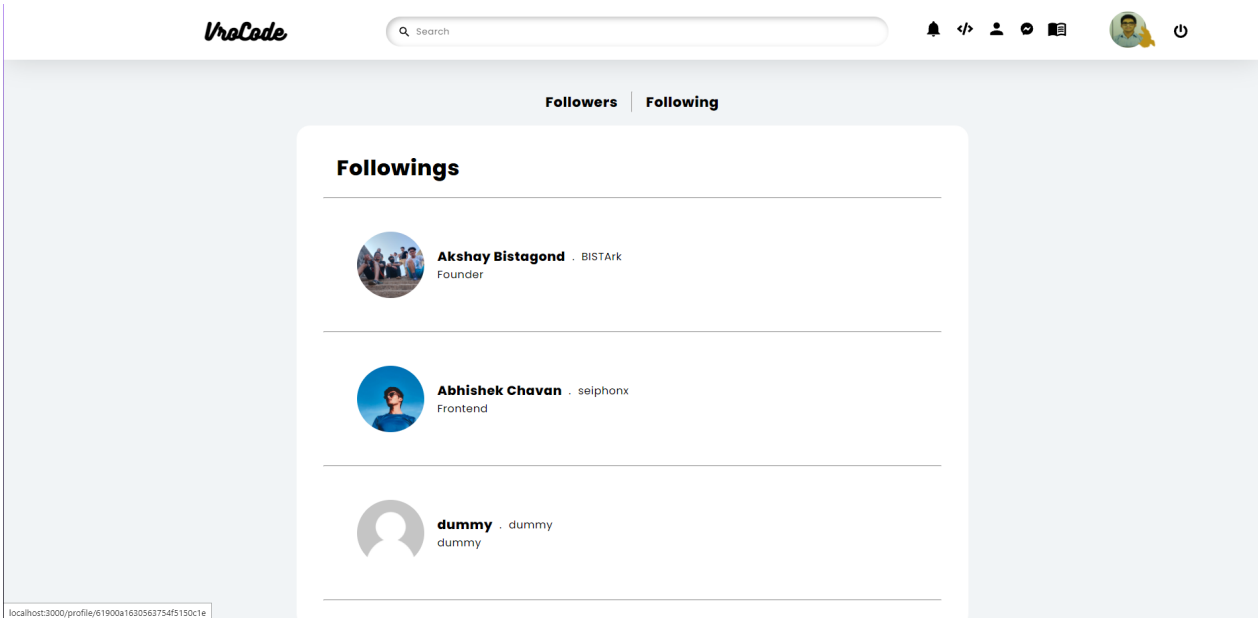


Figure 9: Following

5.6 Chat

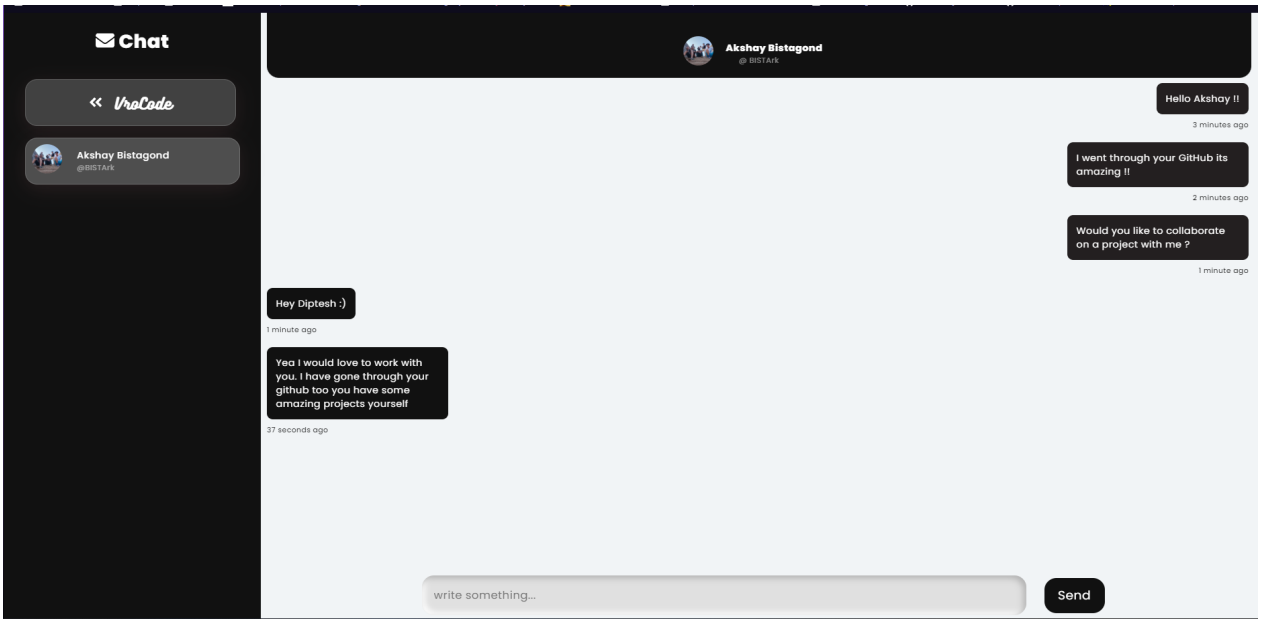


Figure 10: Chat application interface

5.7 Post

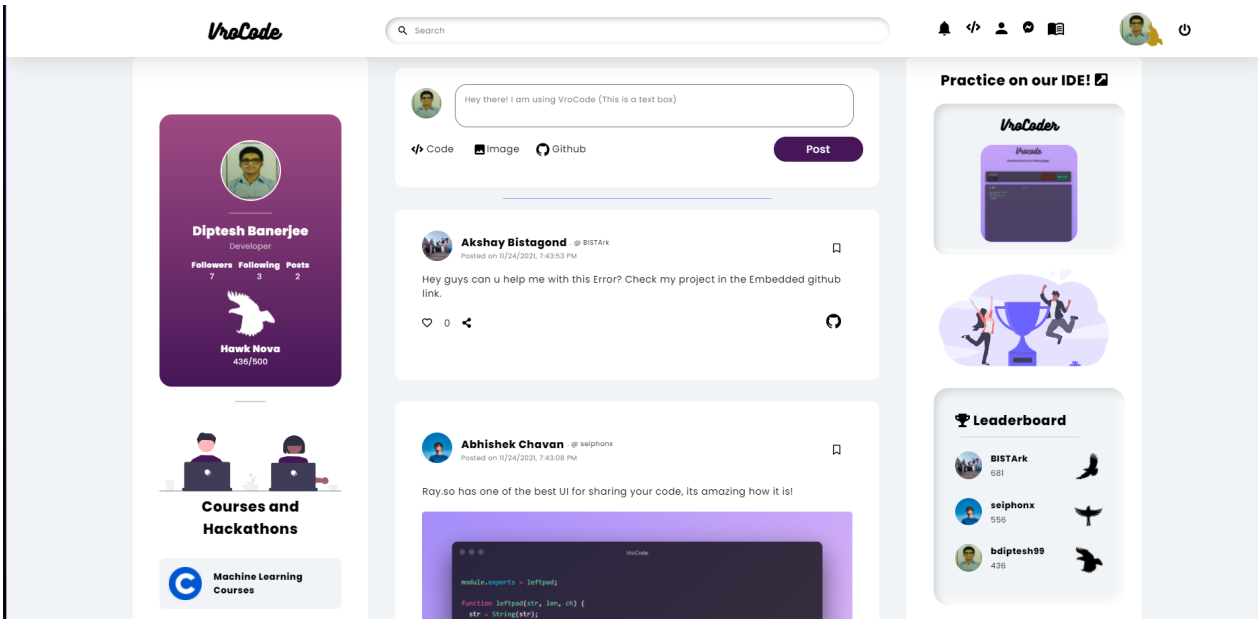


Figure 11: Home page UI

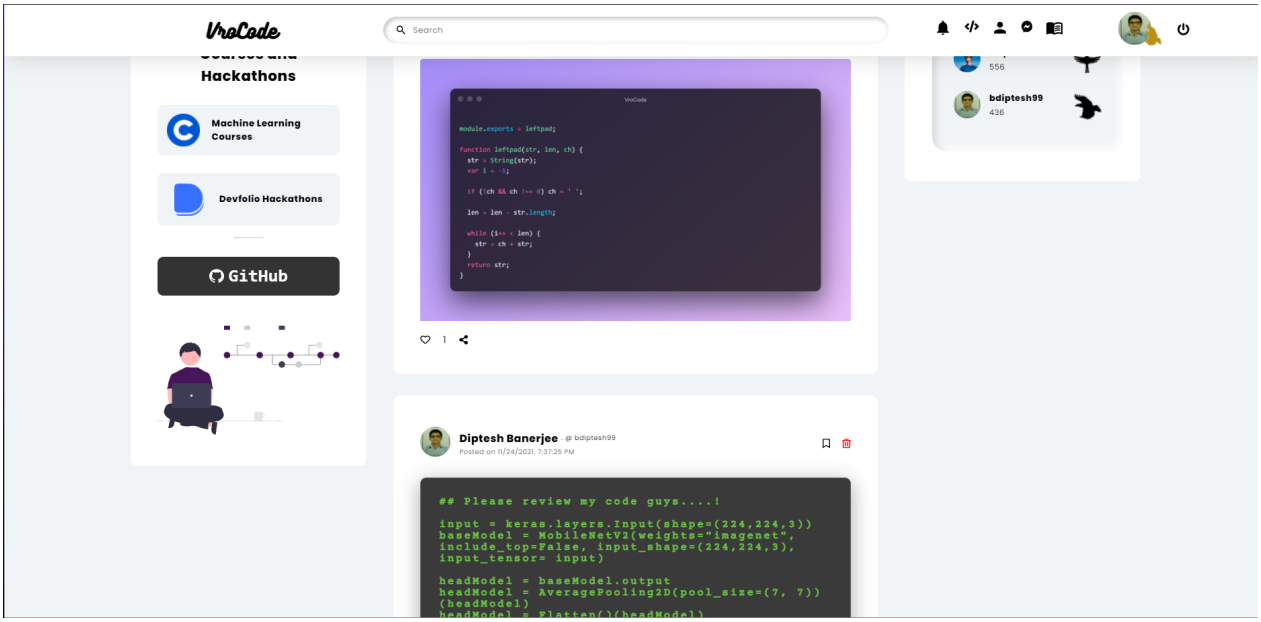


Figure 12: Random course suggestion, Github and Leader board UI

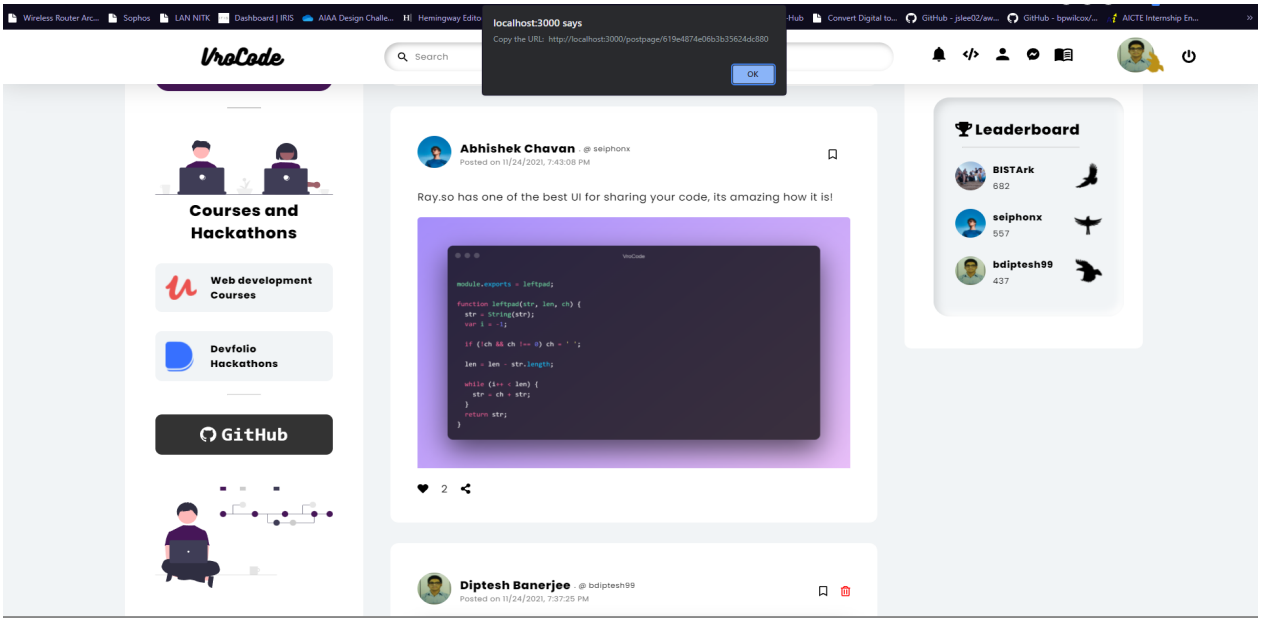


Figure 13: Share Post alert

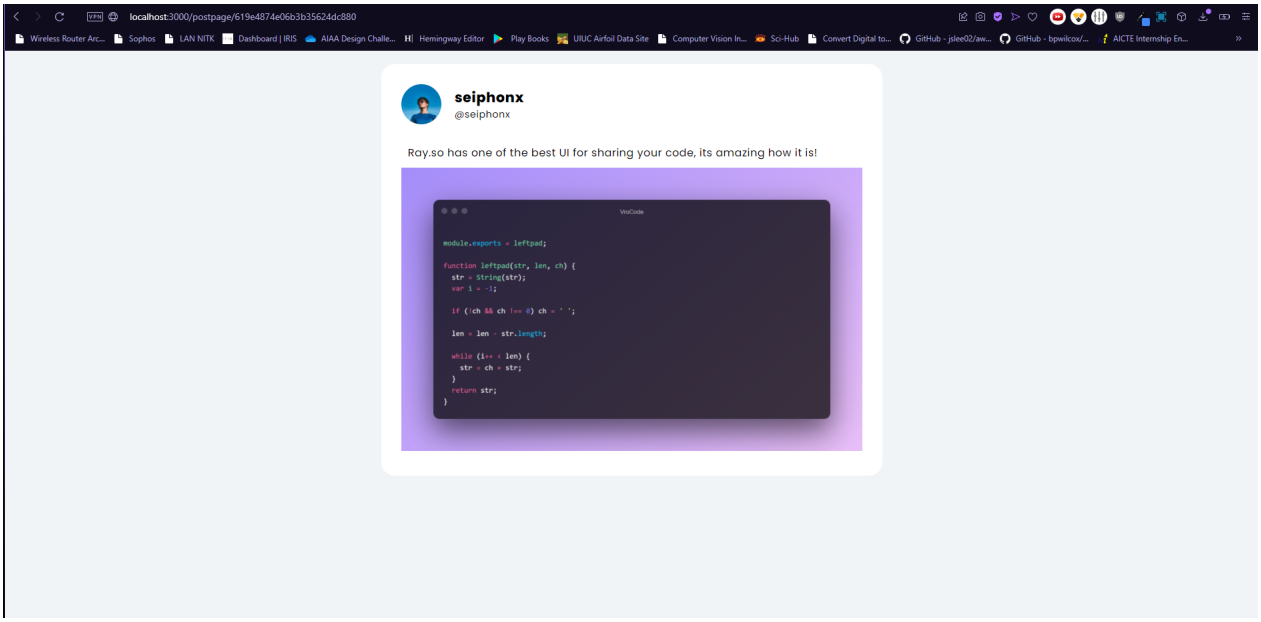


Figure 14: Share Post window (can be accessed by non-user)

5.8 IDE - Online judge

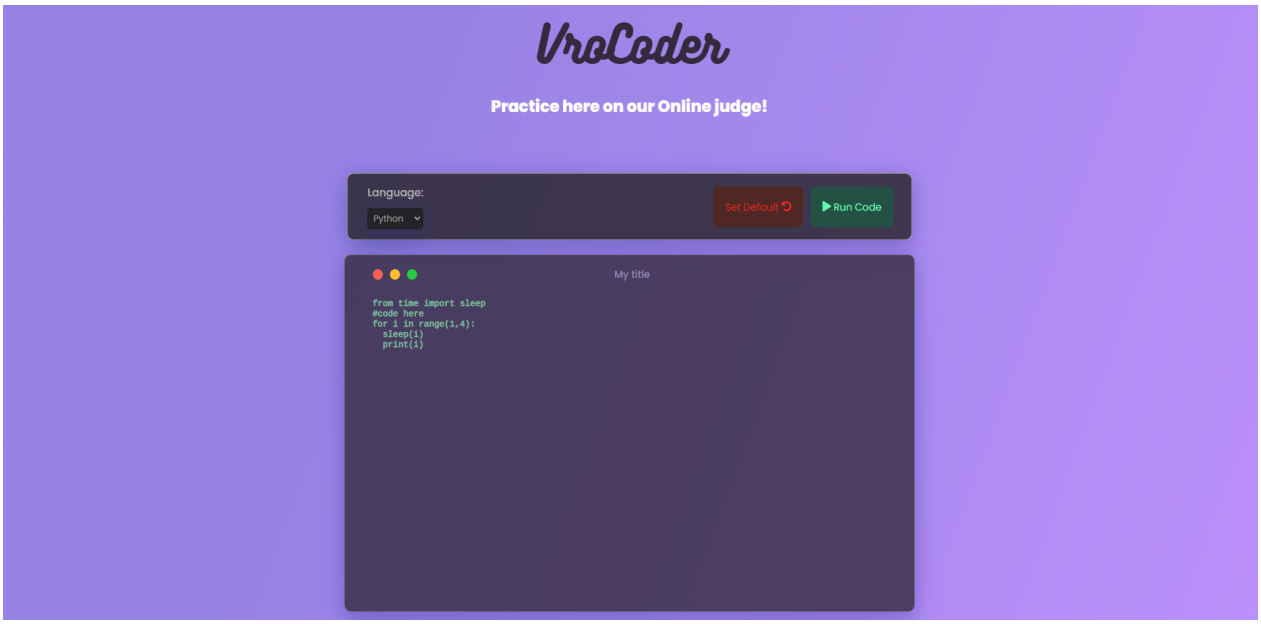


Figure 15: IDE interface

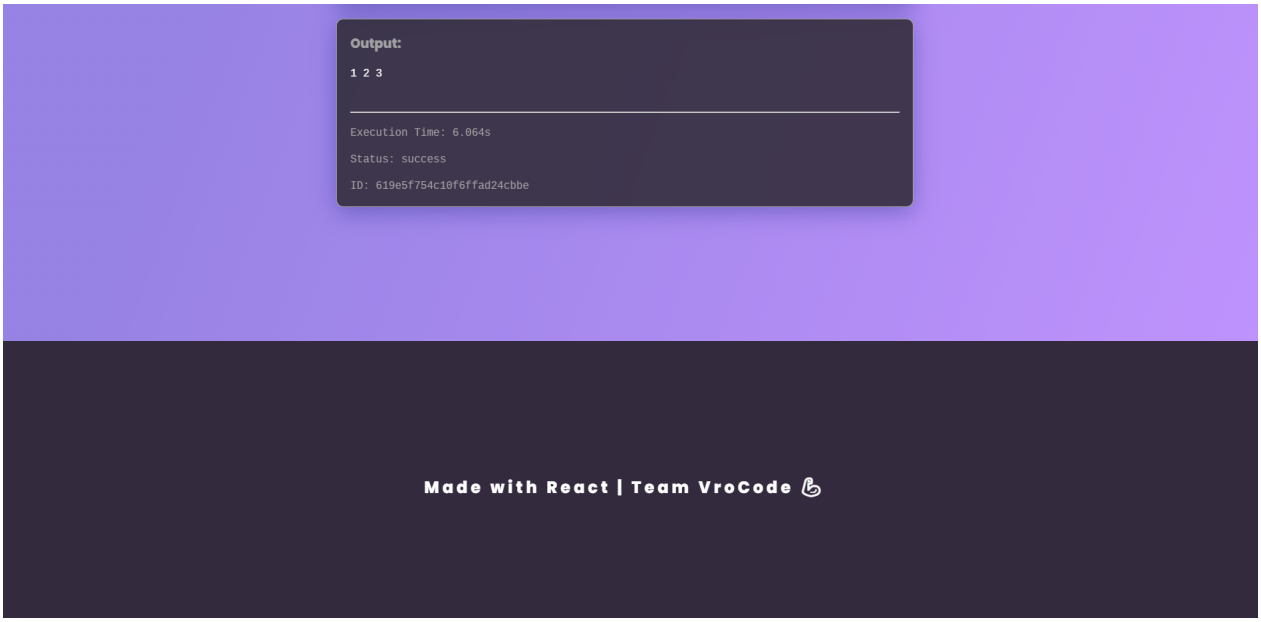


Figure 16: Execution of the program

5.9 Notifications

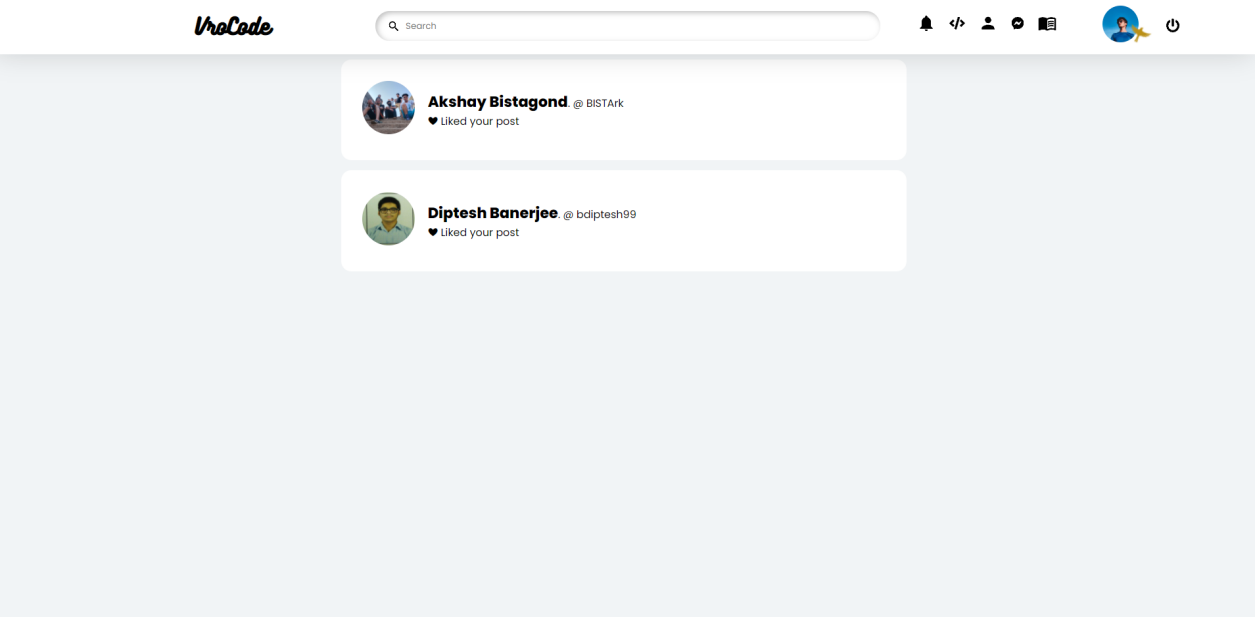


Figure 17: Notifications when other user likes the post

5.10 Course sites

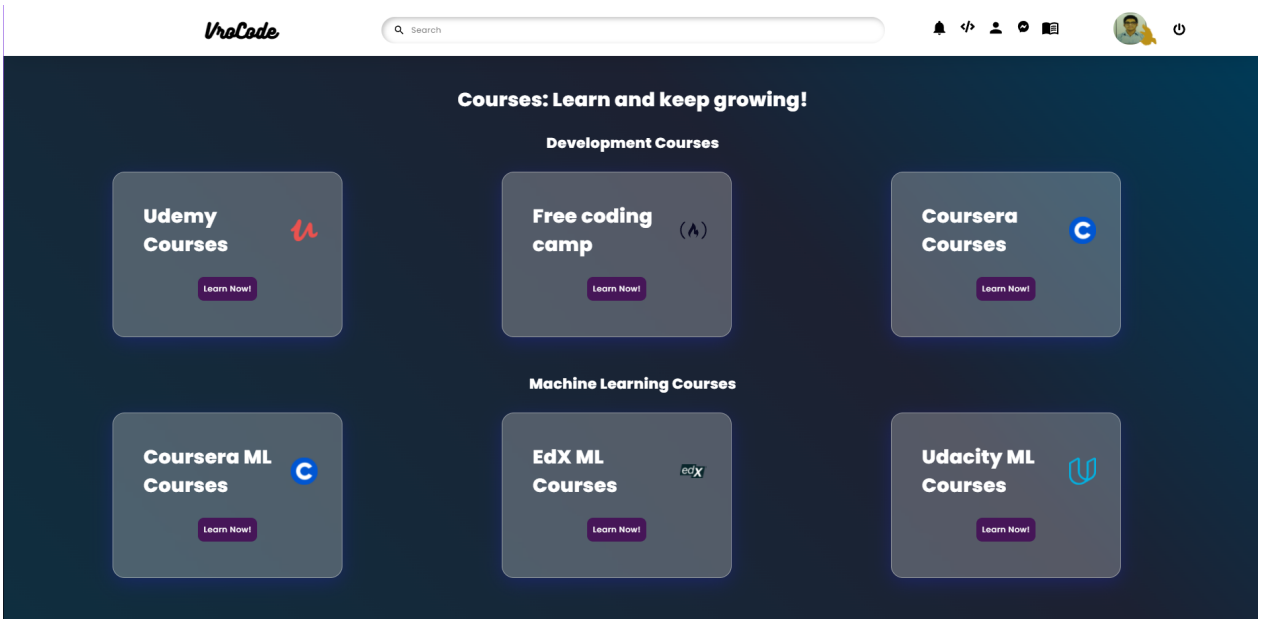


Figure 18: Courses List

5.11 Competitive sites

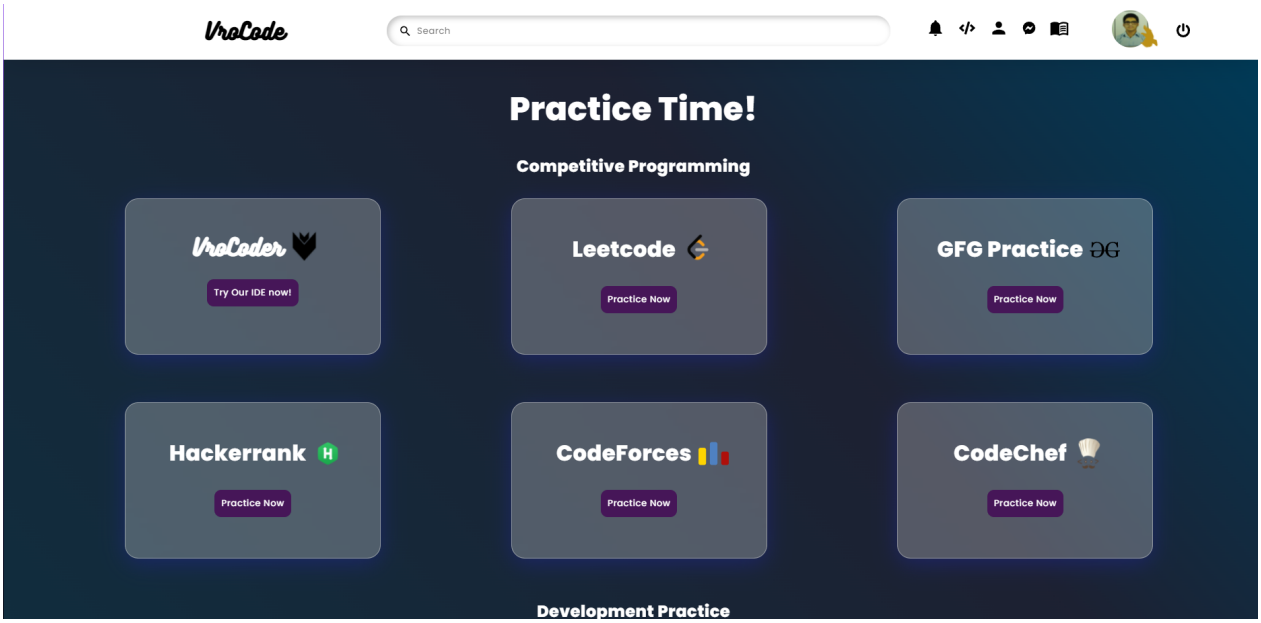


Figure 19: Coding and programming practice List

6 Conclusion and Future Work

At present our website has all the features that are required for a social media website specially designed for programmers and coders. There are minor bugs that need to be fixed and some feature components can be improved based on user feedback. Our future prospect is to use openAI GPT-3 in the search so that users can search almost anything with any keyword and there is auto complete and suggestion feature. In chat, a group chat system will be introduced so that multiple people can chat together.
We will also work on google and GitHub login integration.

References

[1] Latex Template for NITK, Adithya Bhat, Dept. of Information Technology, NITK, https://github.com/adithyabhatkajake/latex-template-nitk

[2] https://github.com/safak/youtube/tree/mern-social-app

[3] https://dev.to/yosraskhiri/how-to-upload-an-image-using-mern-stack-1j95

[4] https://notion.so

[5] https://optimalbits.github.io/bull/