**MINI PROJECT**

**(2020-2021)**

**Real-Time Web Chat App**

**MID-TERM REPORT**

****

**Institute of Engineering & Technology**

**Submitted by :**

**Harshit Verma**

**(181500260)**

**Harshita Katara**

**(181500263)**

***Supervised By: -***

**Harvinder Kaur Mam**

Technical Trainer

**Department of Computer Engineering & Applications**

**Contents**

**Abstract 3**

1. **Introduction 4**

1.1 General Introduction to the topic **4**

1.2 Area of Computer Science **6**

1.3 Hardware and Software Requirements **6**

1. **Problem Definition 7**
2. **Objectives 8**
3. **Implementation Details 9**
4. **Progress till Date & The Remaining work 16**
5. **Some Screenshots 16**
6. **References 24**

**Abstract**

Chat application is a feature or a program on the Internet to communicate directly among Internet users who are online or who are equally using the internet. Chat applications allow users to communicate even though from a great distance. Therefore, this chat application must be real-time and multi-platform to be used by many users.

In this we are developing a Real-Time Web Chat Application using Web-Technology with Firebase Server. Users will always get the latest version of a chat service because no software installation or updates are required. It allows users to communicate even though from a great distance.

Therefore, this chat application must be real-time and multi-platform to be used by many users. To access this Web Chat Application, Users have to take authentication by signing up using Mail Id, Passwords and Username. Using the Firebase server, we can check who is authorised to use this Web Chat Application. Users can update profile photos, send gifs, stickers and images.

Our Project is an example of a chat server. To start a chat client should get connected to a server where they can do private chat and group chat security measures were taken during the last one.

**Introduction**

**1.1 General Introduction to the topic**

A web chat is a system that allows users to communicate in real-time using easily accessible web interfaces. It is a type of Internet online chat distinguished by its simplicity and accessibility to users who do not wish to take the time to install and learn to use specialized chat software.This trait allows users instantaneous access and only a web browser is required to chat. Users will always get the latest version of a chat service because no software installation or updates are required.

Web chat software is sometimes used in a business context as live support software, also called live support, live help or live chat. In this case, the web chat software is integrated with a website to allow for a customer to chat with the business representative or the website owner.

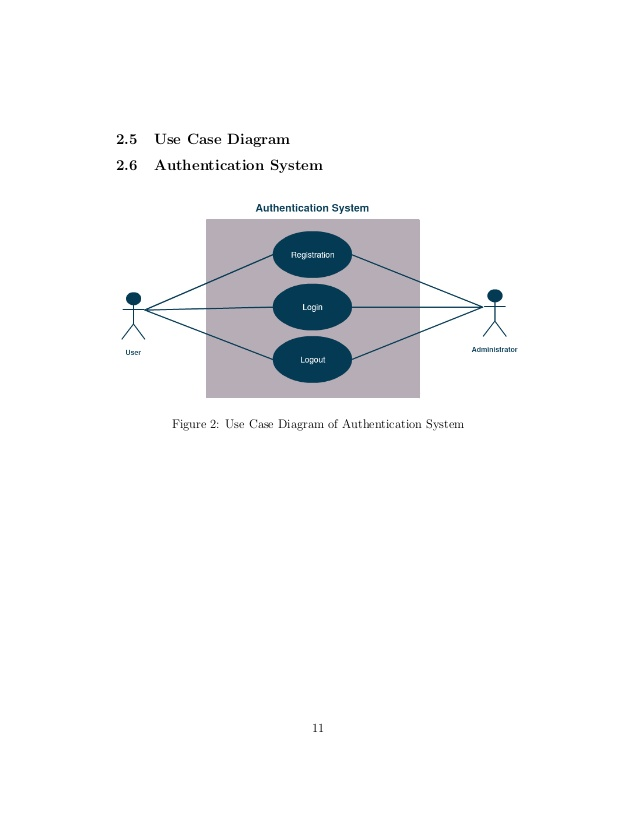
We develop an instant messaging solution using cloud firestore in the backend to enable users to seamlessly communicate with each other.

**About Cloud Firestore :-**

* Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.
* In order to structure your data, you define collections (similar to tables in SQL) which contain documents (similar to rows). Each document contains fields that contain the actual data. You can reference an individual document using its unique path, or you can query a collection for documents whose fields contain the data you’re looking for.
* Using the [Cloud console](https://console.cloud.google.com/), you can browse data in the Cloud Firestore database in your project.
* Like Realtime Database, Cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.

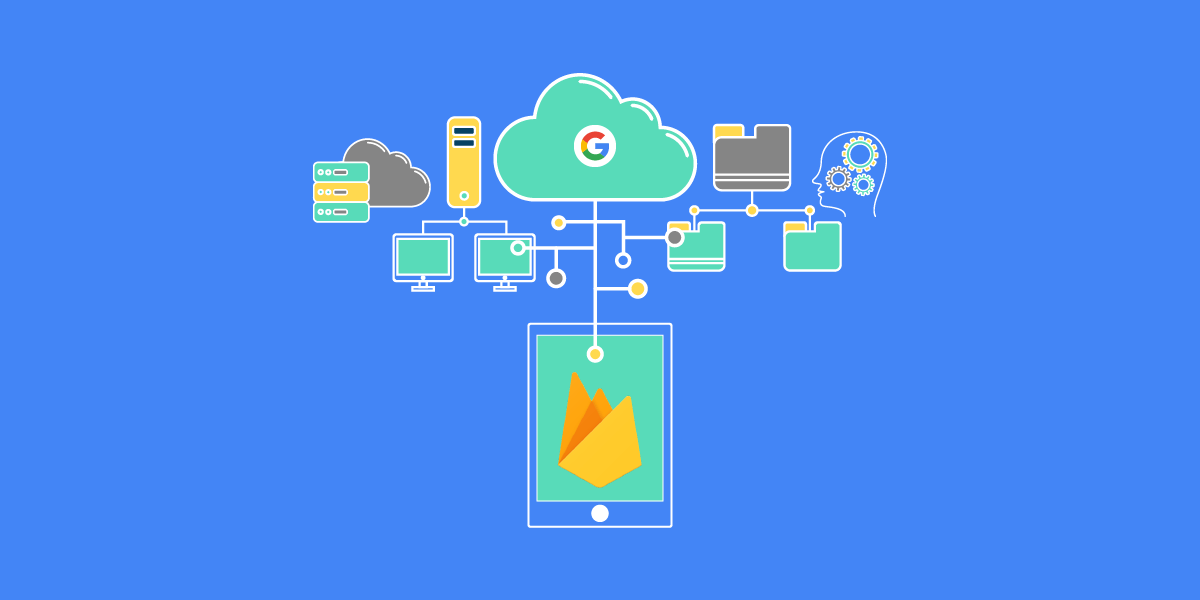
**How Cloud Firestore Works :-**

* Like Realtime Database, Cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.
* Cloud Firestore caches data that your app is actively using, so the app can write, read, listen to, and query data even if the device is offline. When the device comes back online, Cloud Firestore synchronizes any local changes back to Cloud Firestore.
* Capabilities of real-time messaging and notifications.



**1.2 Area of Computer Science**

Firebase is really a good showcase for the capabilities of real-time messaging and notifications. Firebase offers a generous free tier that includes authentication and access to their Realtime Database. The authentication providers we’ll be covering email and password — Google and GitHub — are free on that side as well. The Realtime Database allows up to 100 simultaneous connections and 1 gigabyte storage per month.



**Web Technology:**

Social Networking Site

* HTML
* CSS
* JavaScript
* React - A JavaScript Library

**1.3 Hardware and Software Requirements**

1. **Hardware Requirements :**

* Latest Configuration Laptop

1. **Software Requirements :**

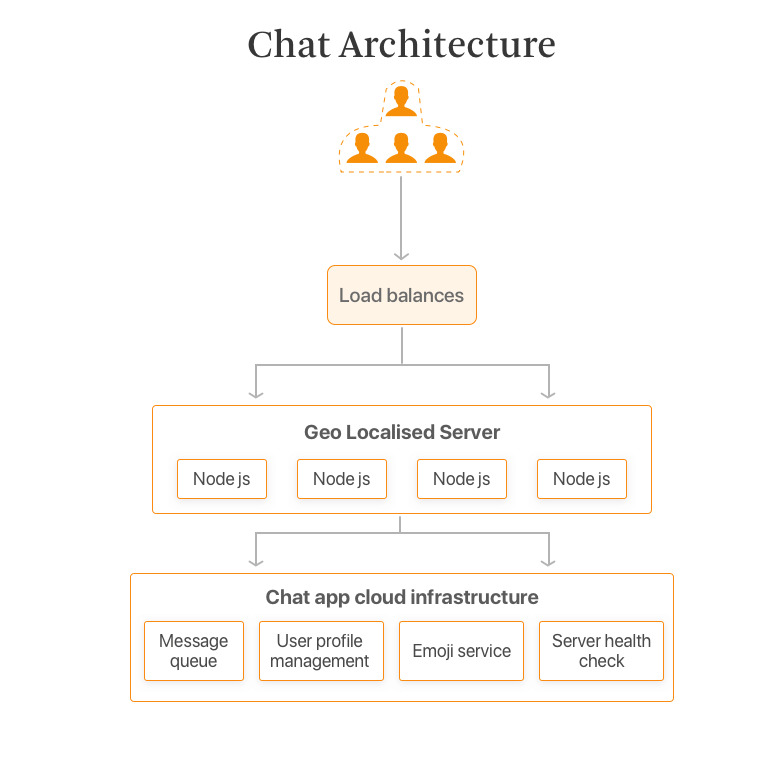
* Visual Studio (Version 1.38)
* Firebase (Backend)

**Problem Definition**

This project is to create a Real-time Chat Application with firebase server and users to enable the users to chat with each other.

To develop an instant messaging solution to enable users to seamlessly communicate with each other.

The project should be very easy to use, enabling even a lay-man to use it.



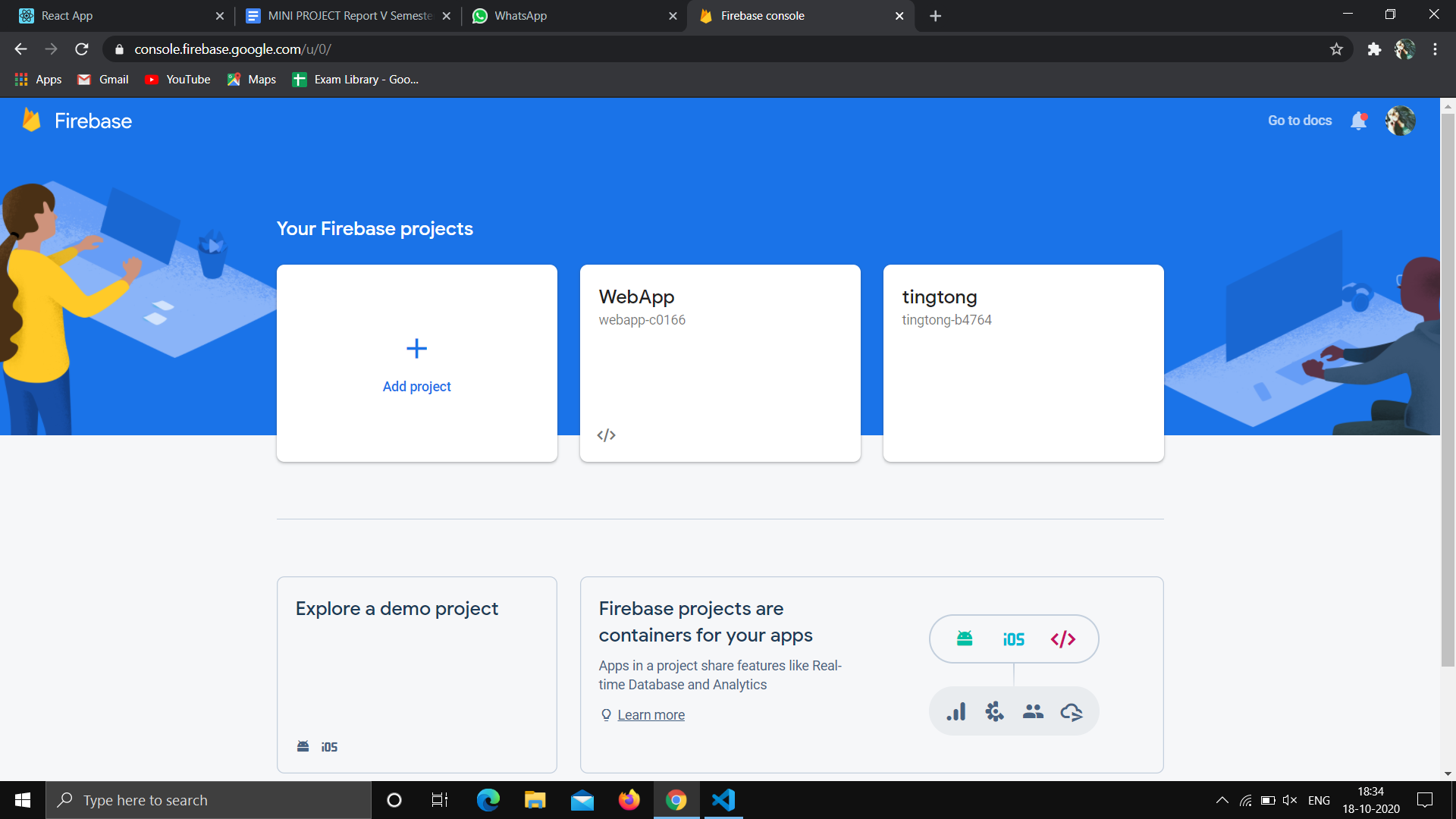
**Objective**

The main objective of the project is to manage the details of Online Chat, Chat Application, Chat History, Chat Profile, users. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build a web application program to reduce the manual work for managing the Online chat,chat History. It tracks all the details about the Chat History, Chat profile ,users.

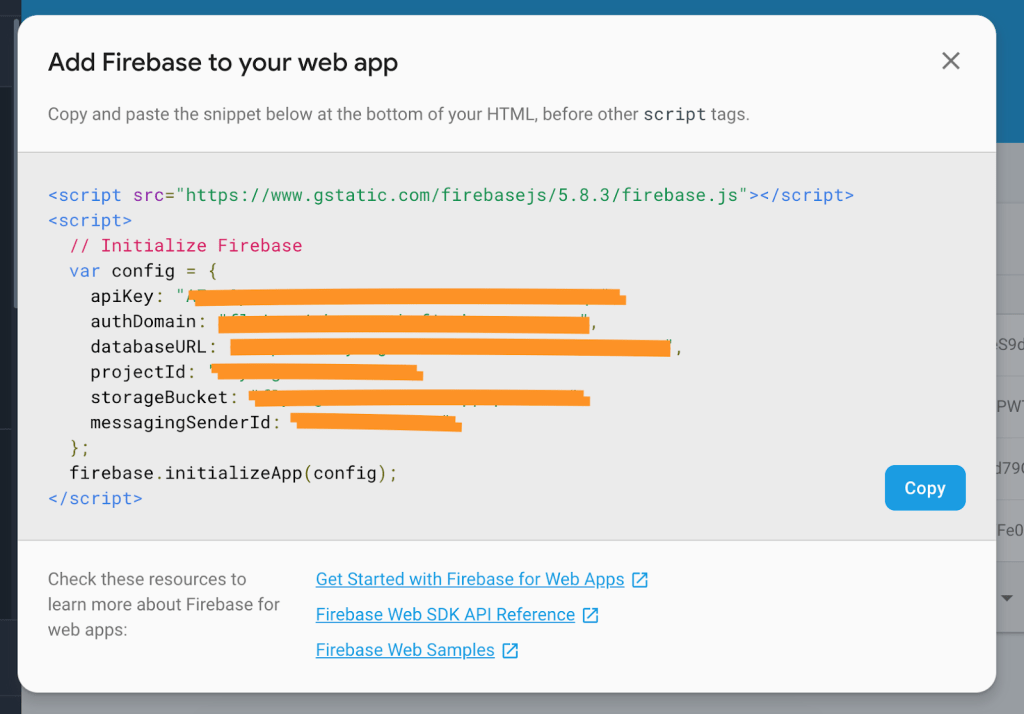
**Implementation Details**

**Part 1:** To develop a react chat app using cloud firestore in the backend.

With the use of firebase console, we have created a project name webApp.

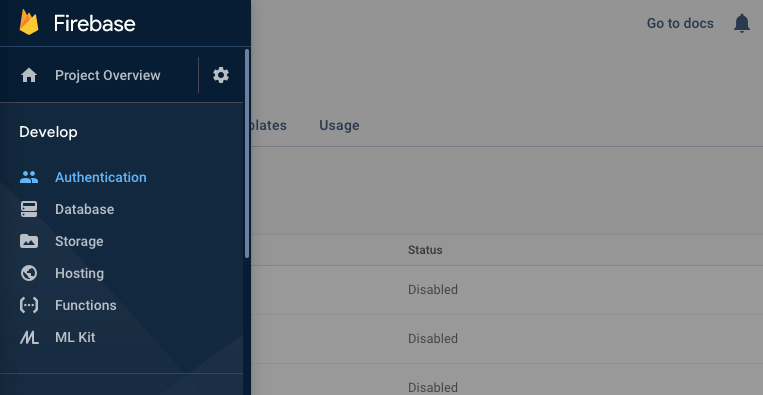


Before we can start using Firebase in our web app, we have to get the configuration details for our project.

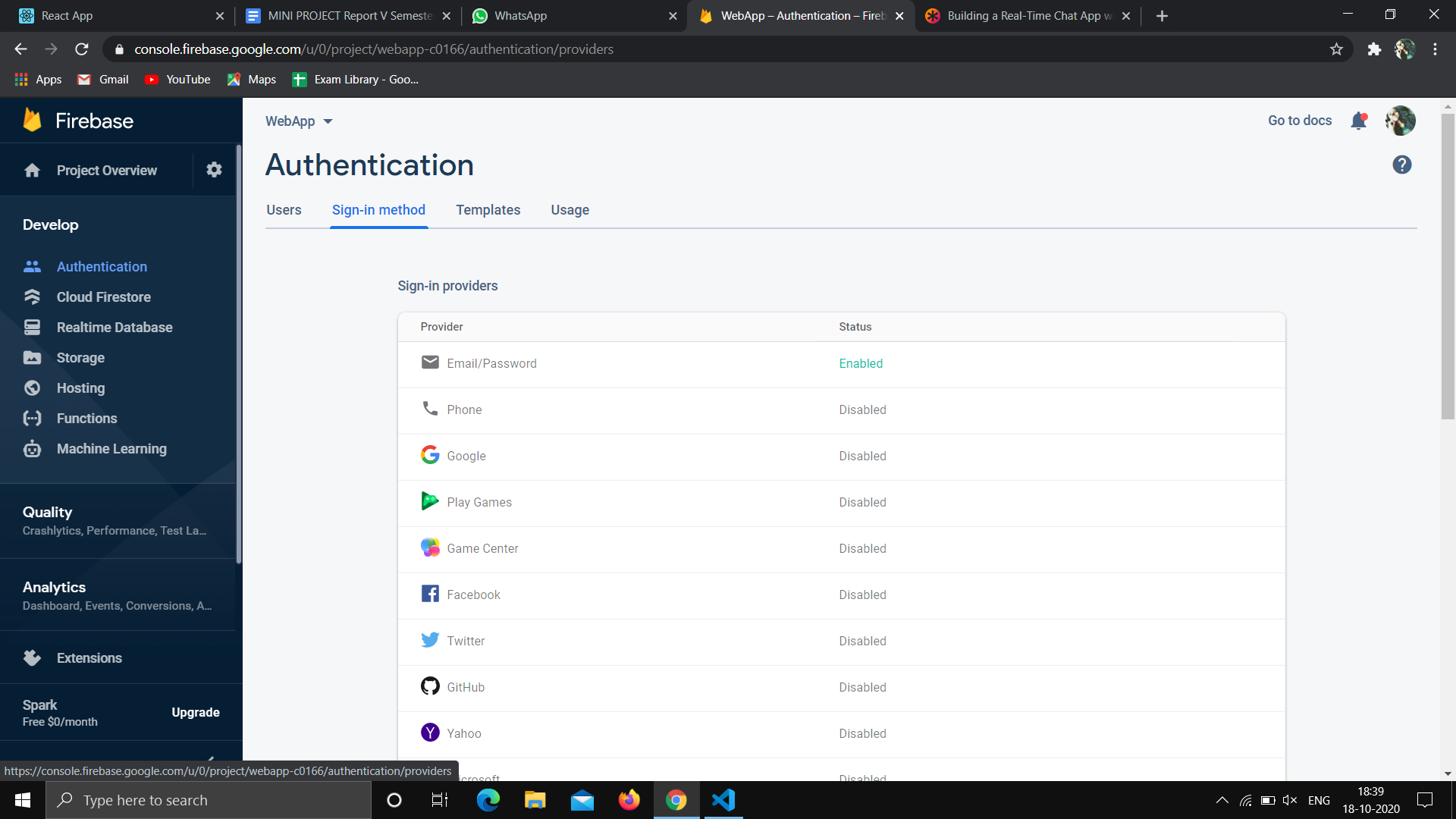


Now we have copied and stored the configuration details on the next screen in a safe place. That will come in handy in the next step.

To access this app users should have to first sign-up by email and password. So, we have to authenticate users via email and password. We need to enable these from the Authentication tab in the dashboard.



There’s a **Sign-in method** tab in the Firebase dashboard. We have enabled it.



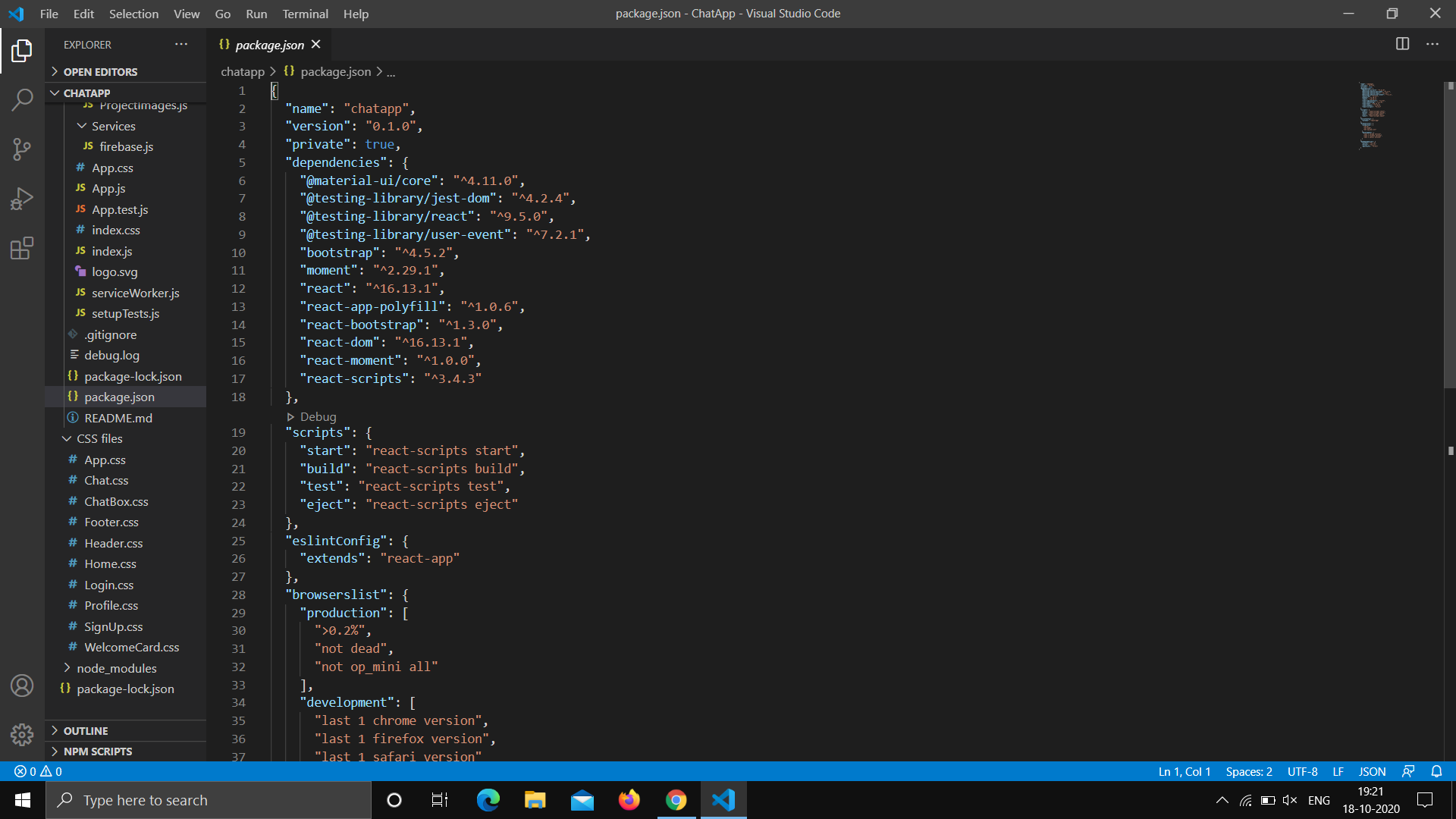
Now we have set up a firebase and we can use it in our App.

**Part 2 :** Setting Up the Web App

For our web app, we are using React but most of the concepts can be applied to any other framework. We have used Node.js for a React setup.

We have used [create-react-app](https://github.com/facebook/create-react-app) to bootstrap a new React project. This downloads and installs the necessary packages required for a React application. In the terminal, cd into where our project folder like our project to go and run this to initialize it.

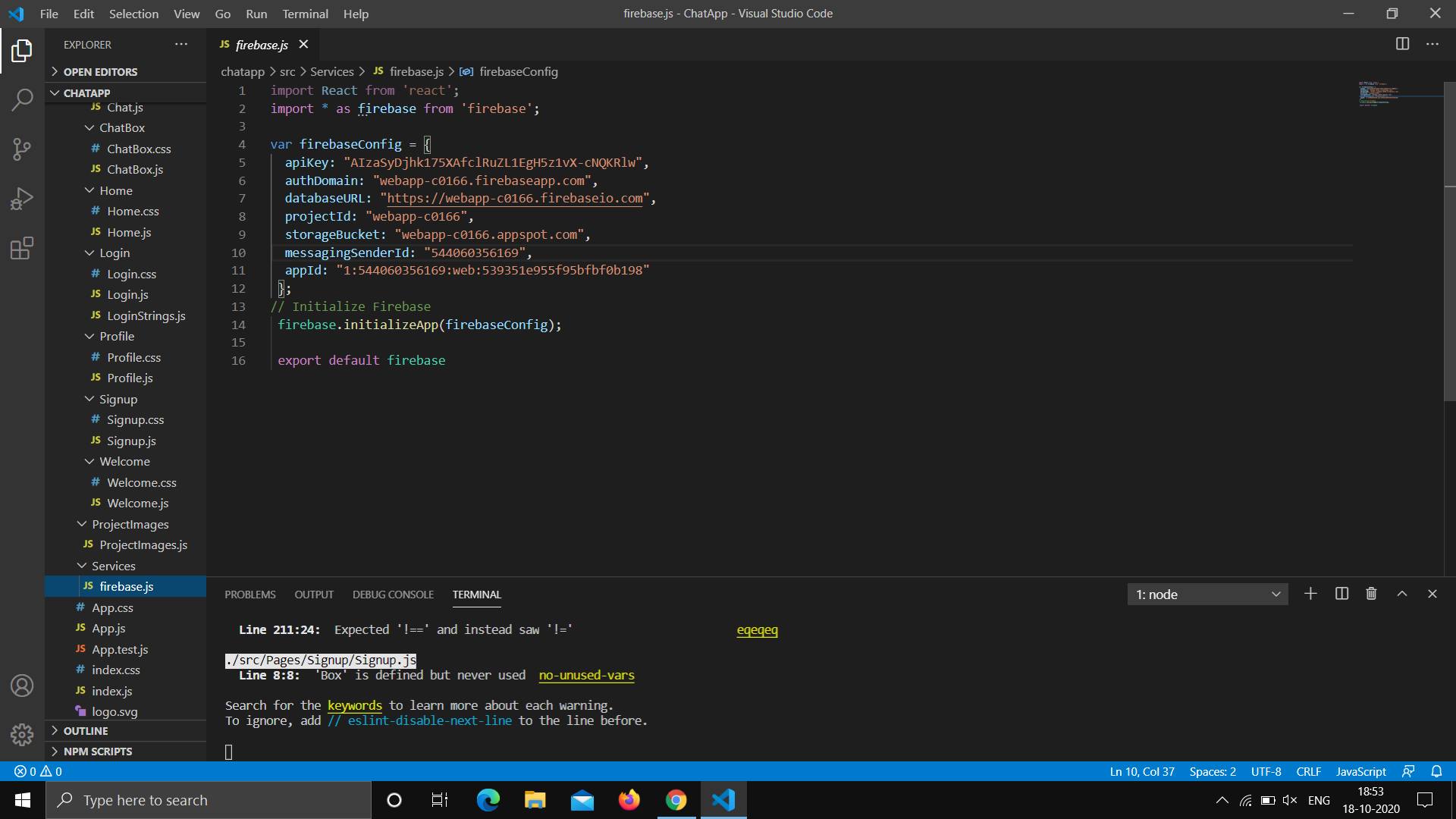
Using various react commands for the initial setup for our react app and installs the dependencies in package.json.



If everything got installed correctly, see a screen like this:



**Part 3 :** Now We have imported and initialized Firebase using the configuration details we copied earlier when registering the app in the Firebase dashboard. Then, we have exported the authentication and database modules.



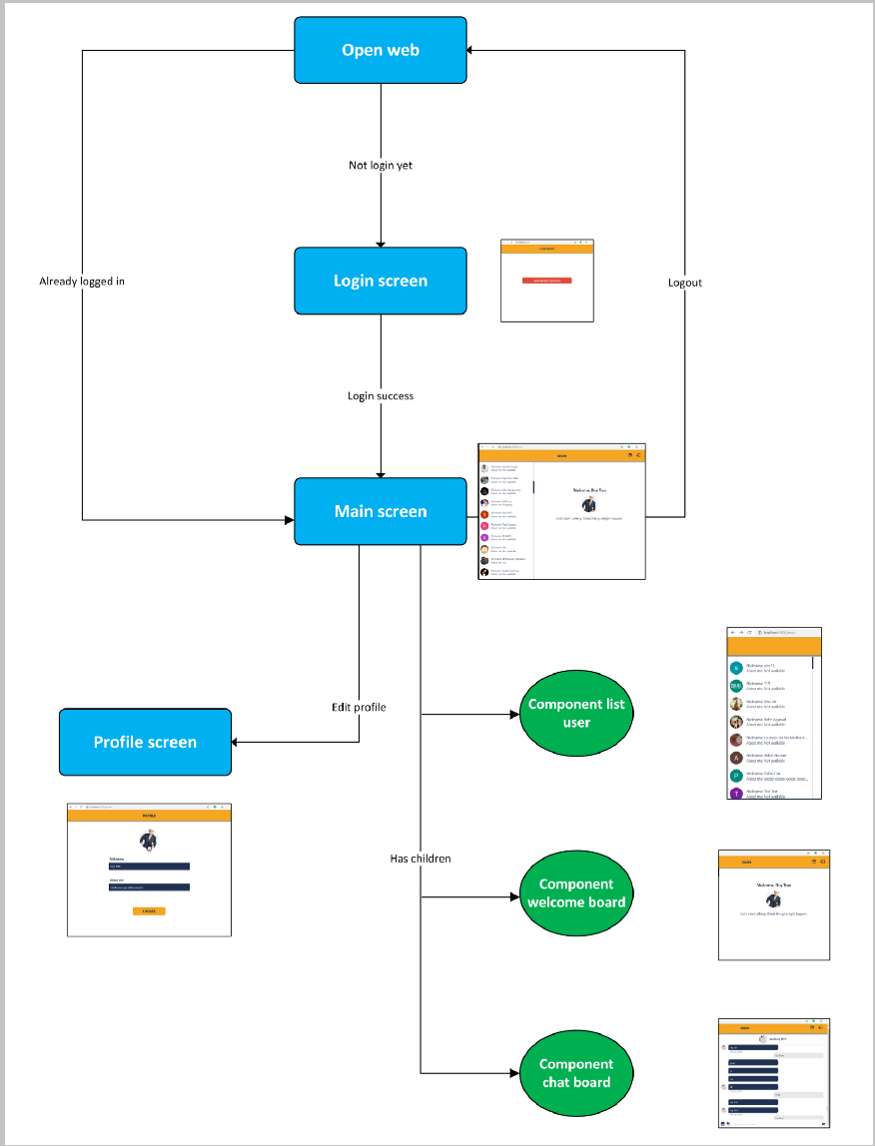
**Part 4 :** We have written the logic to check if the user is indeed authenticated.

**Part 5 :** We have authenticating users with email and password.

Users can update their profile by clicking on the profile icon and mentioning his/her name with bio.

So, whoever in users contact will be loaded and users can search anyone by just typing name and can see the history of their chats too and time will also display that at what time users had a chat with someone.

In this application, we have also added a feature of message notification and feature of sending messages, stickers and images also have been implemented.



**Progress**

1. Part 1 is completed

2. Part 2 is completed

3. Part 3 is completed

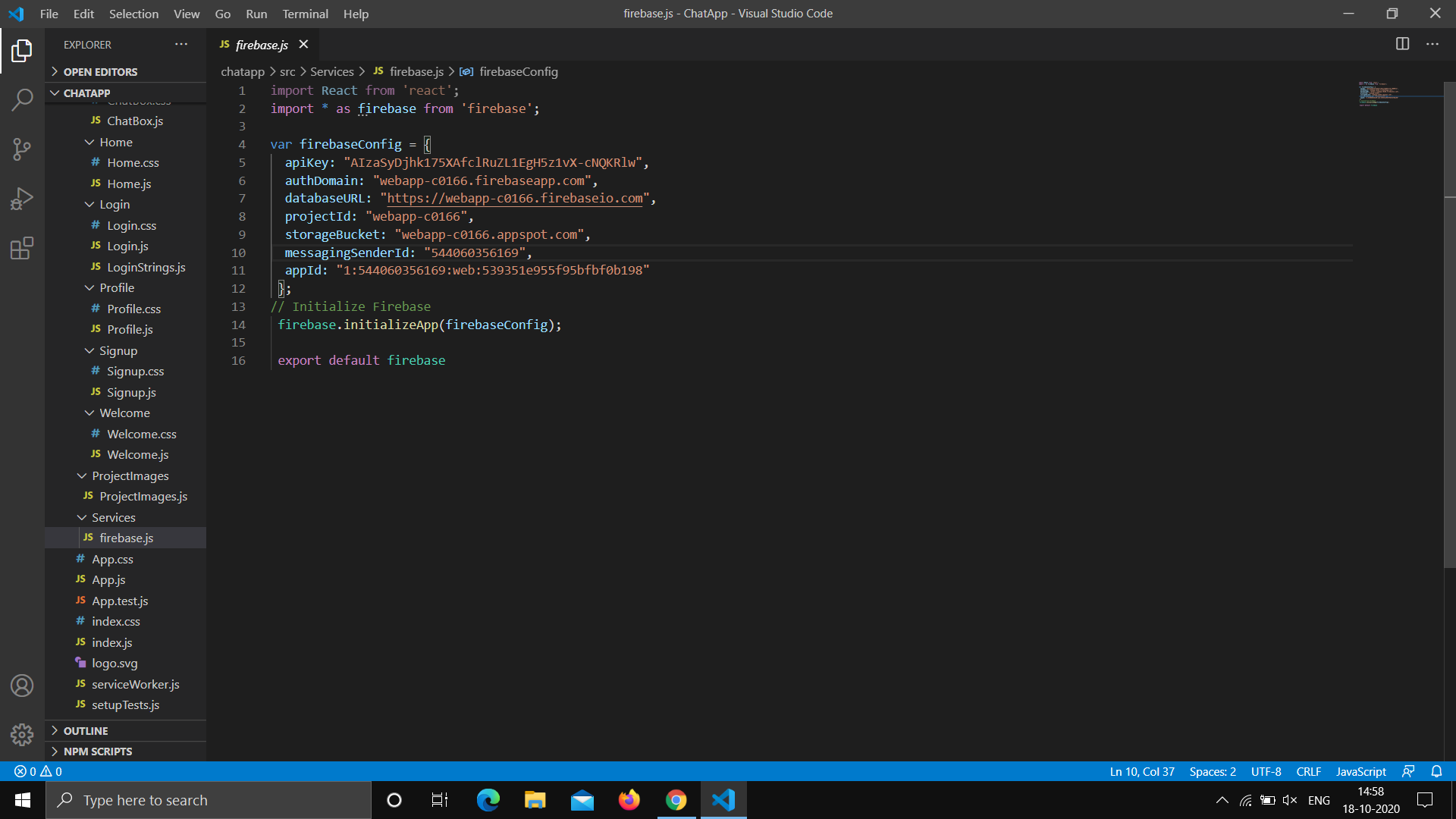
4. Part 4 is completed

**Remaining work :**

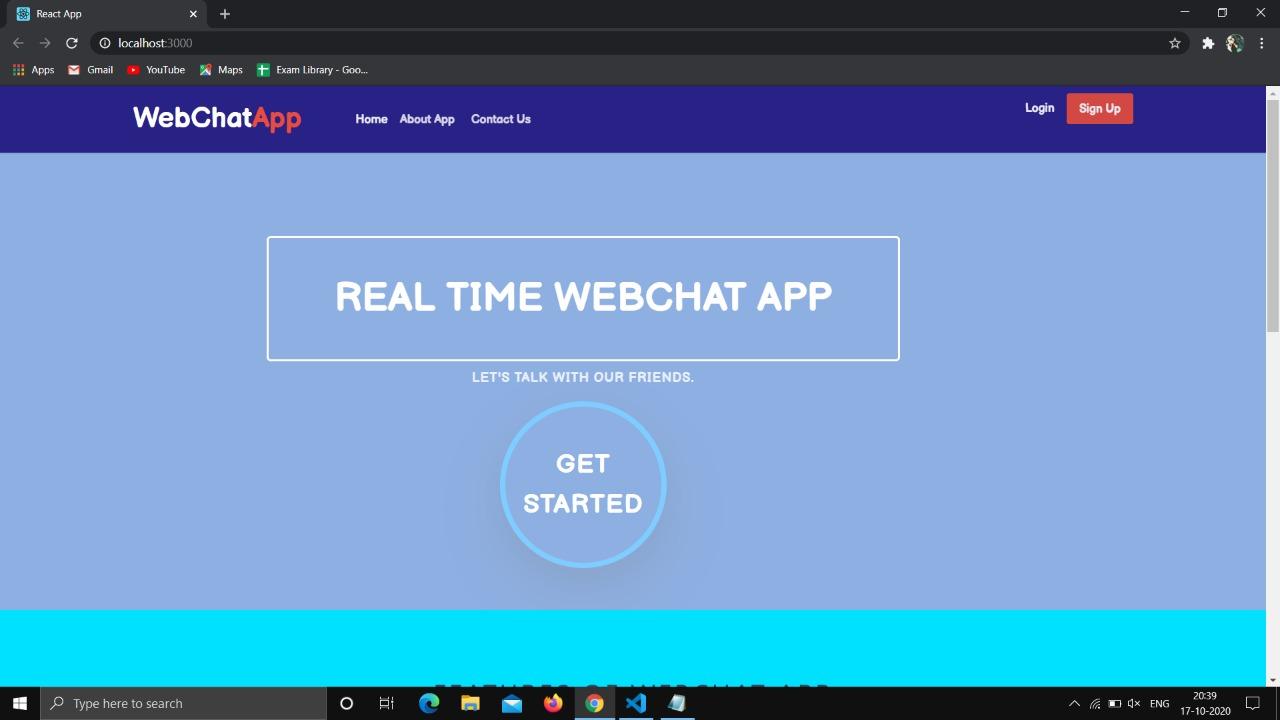
* Some Task in Part 5.
* Add some CSS and Animation.

**SCREENSHOTS**

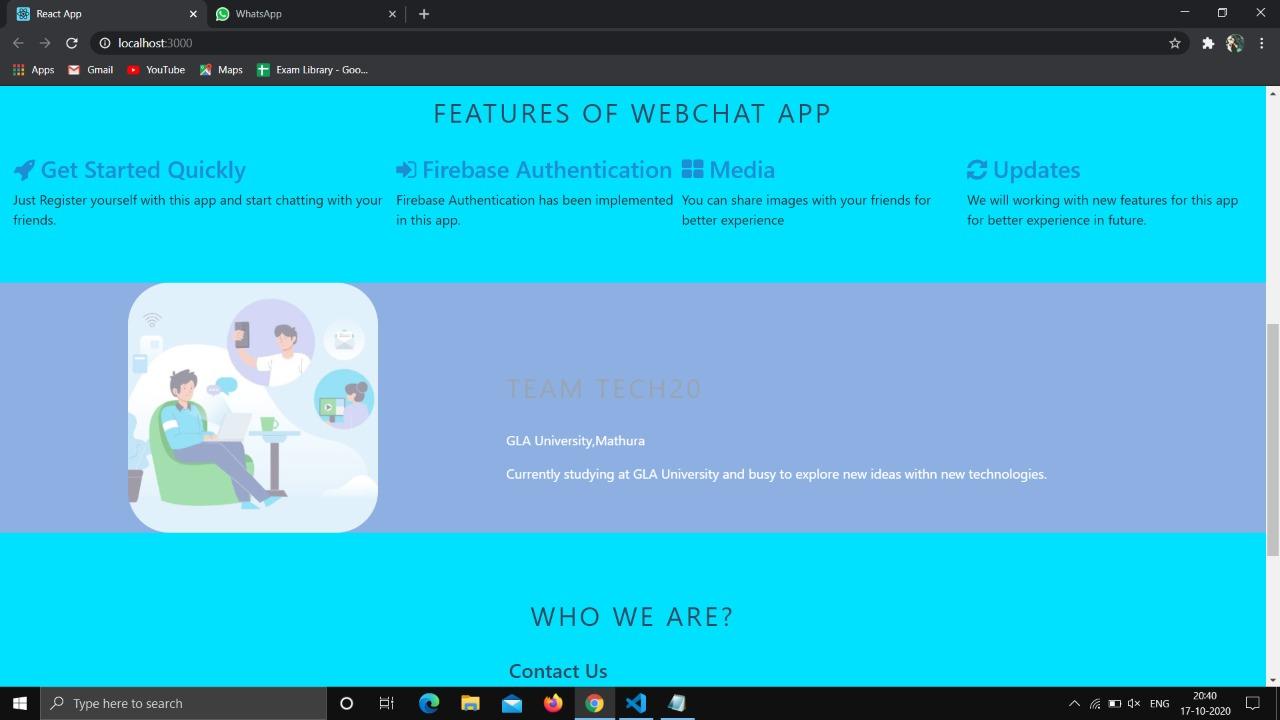
### Get Firebase into the app :

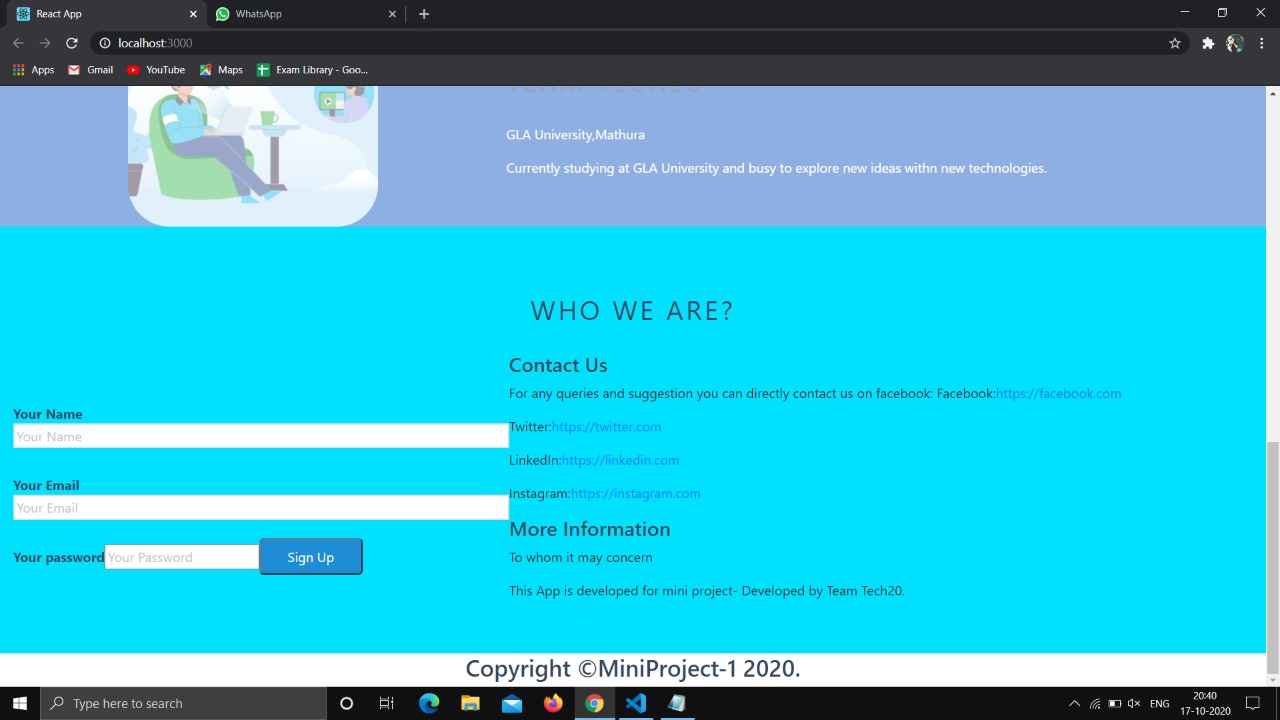


Header of Front-End :

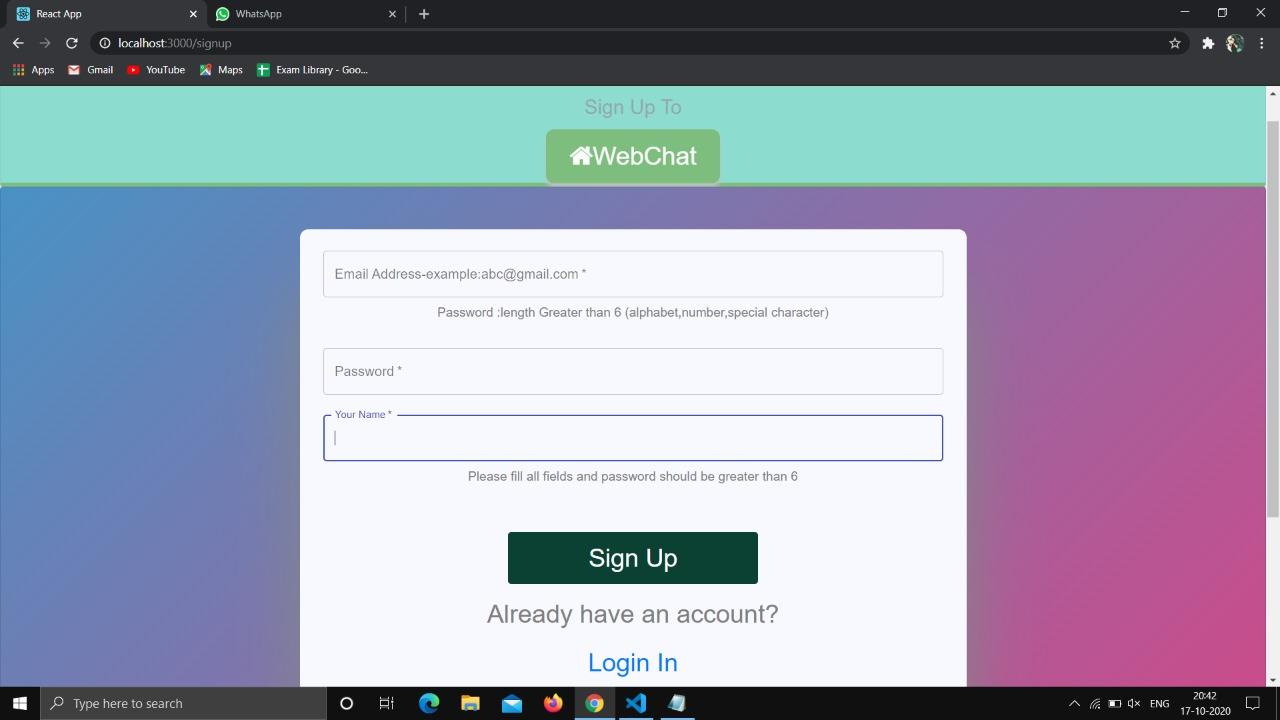


Footer of Front-End :

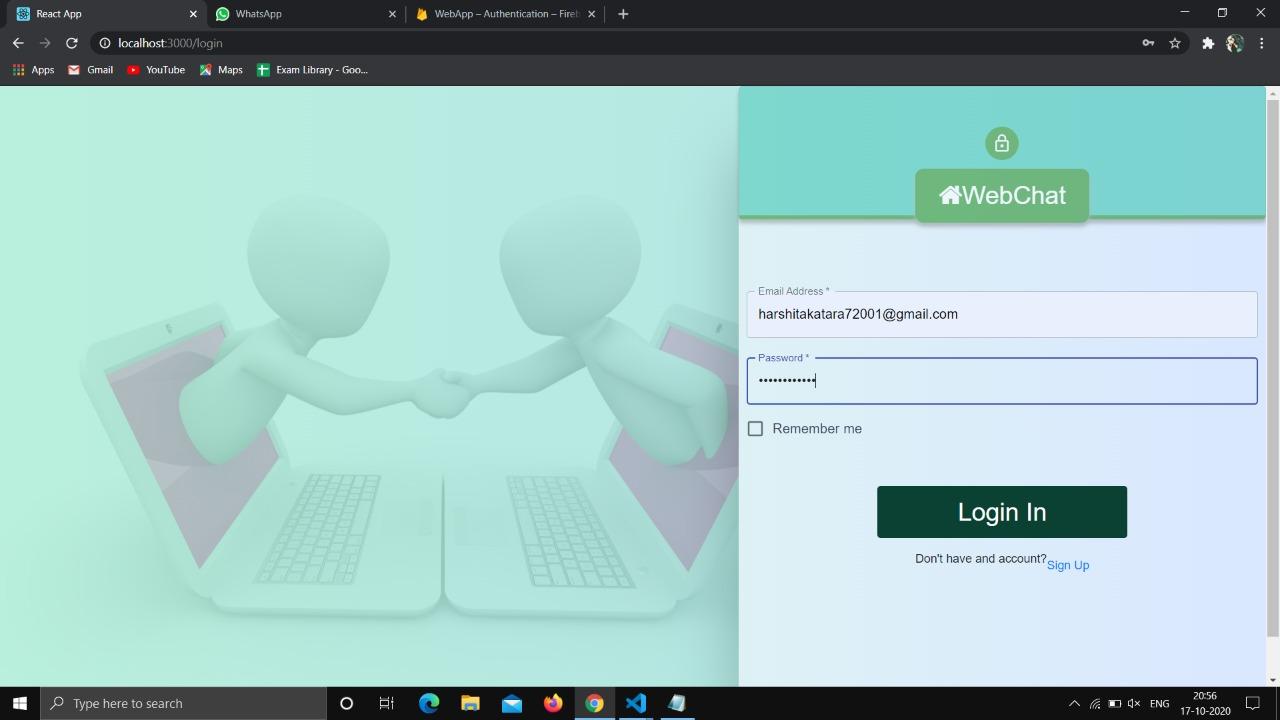




Sign - Up Page :

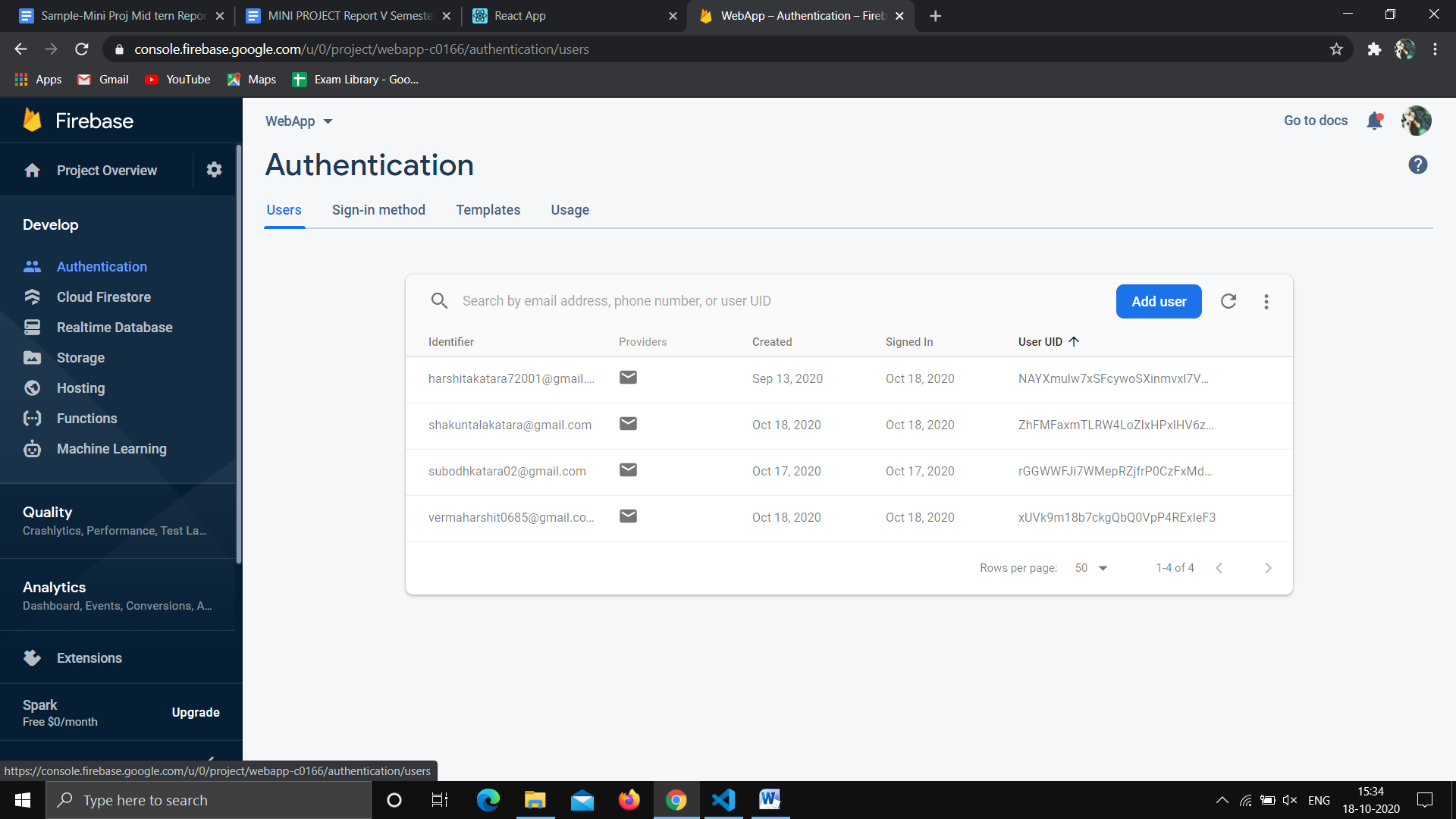


Login Page :

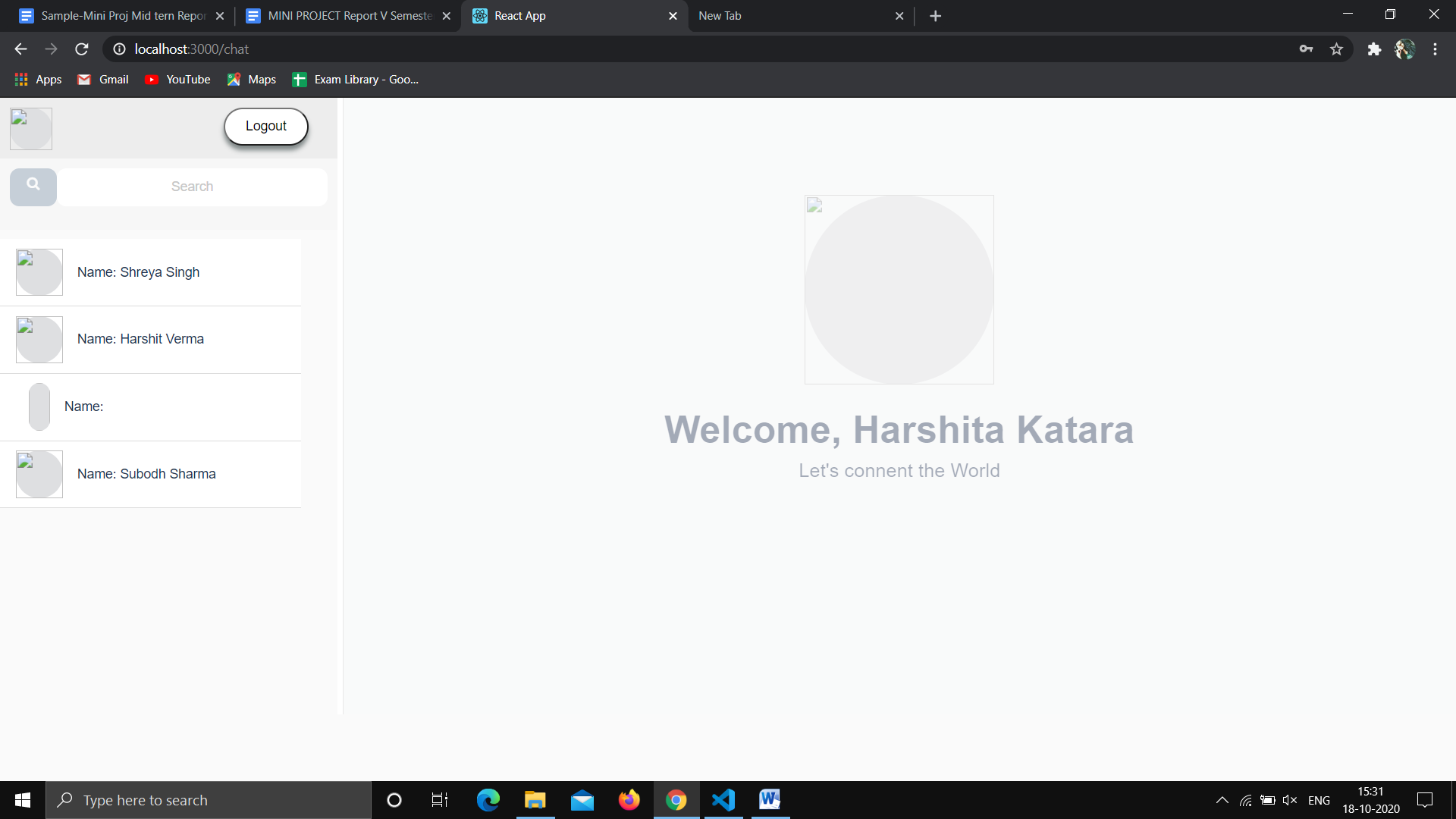


Get Authentication using Firebase :

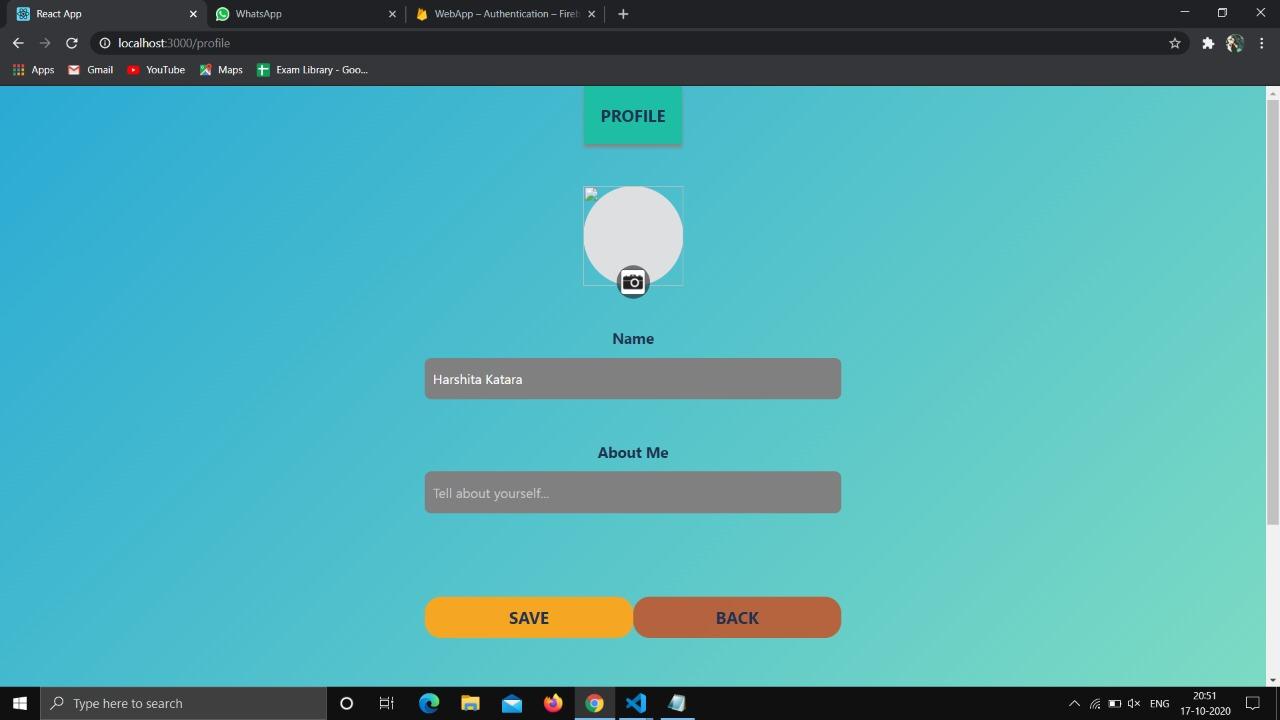




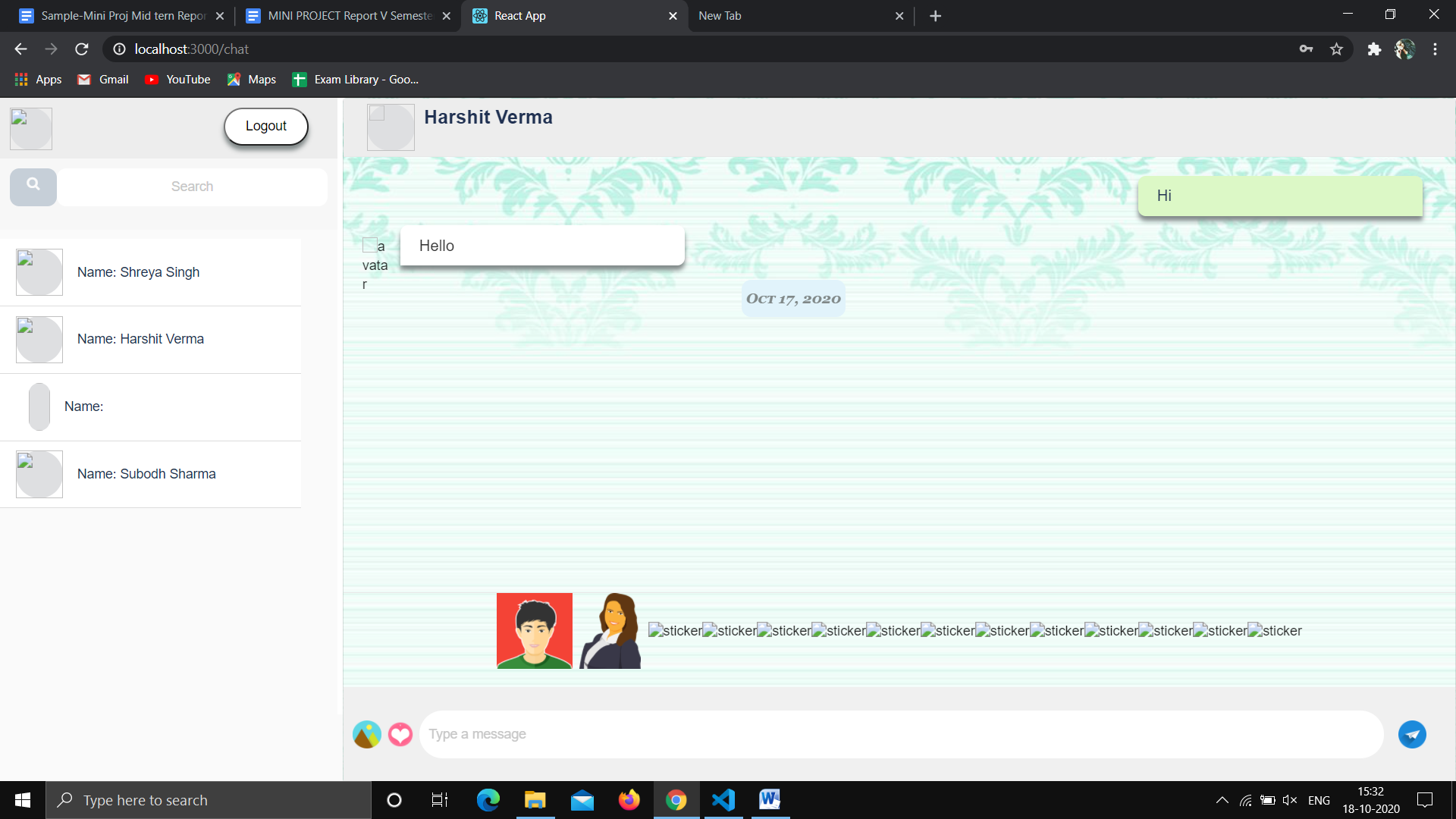
Welcome Page :



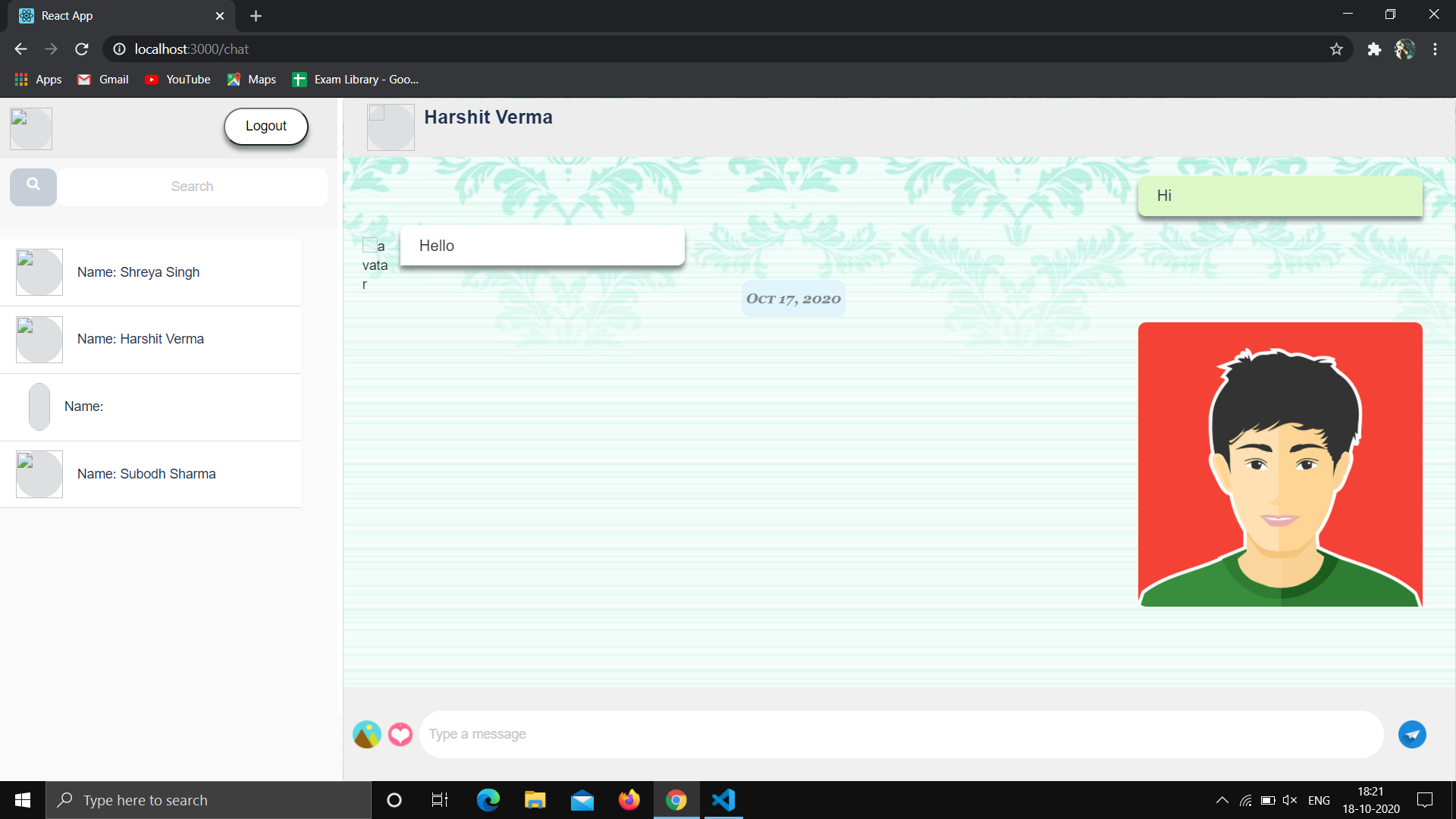
Profile Page :



ChatBox Page :



We can send Stickers:



We can Send Images too:-

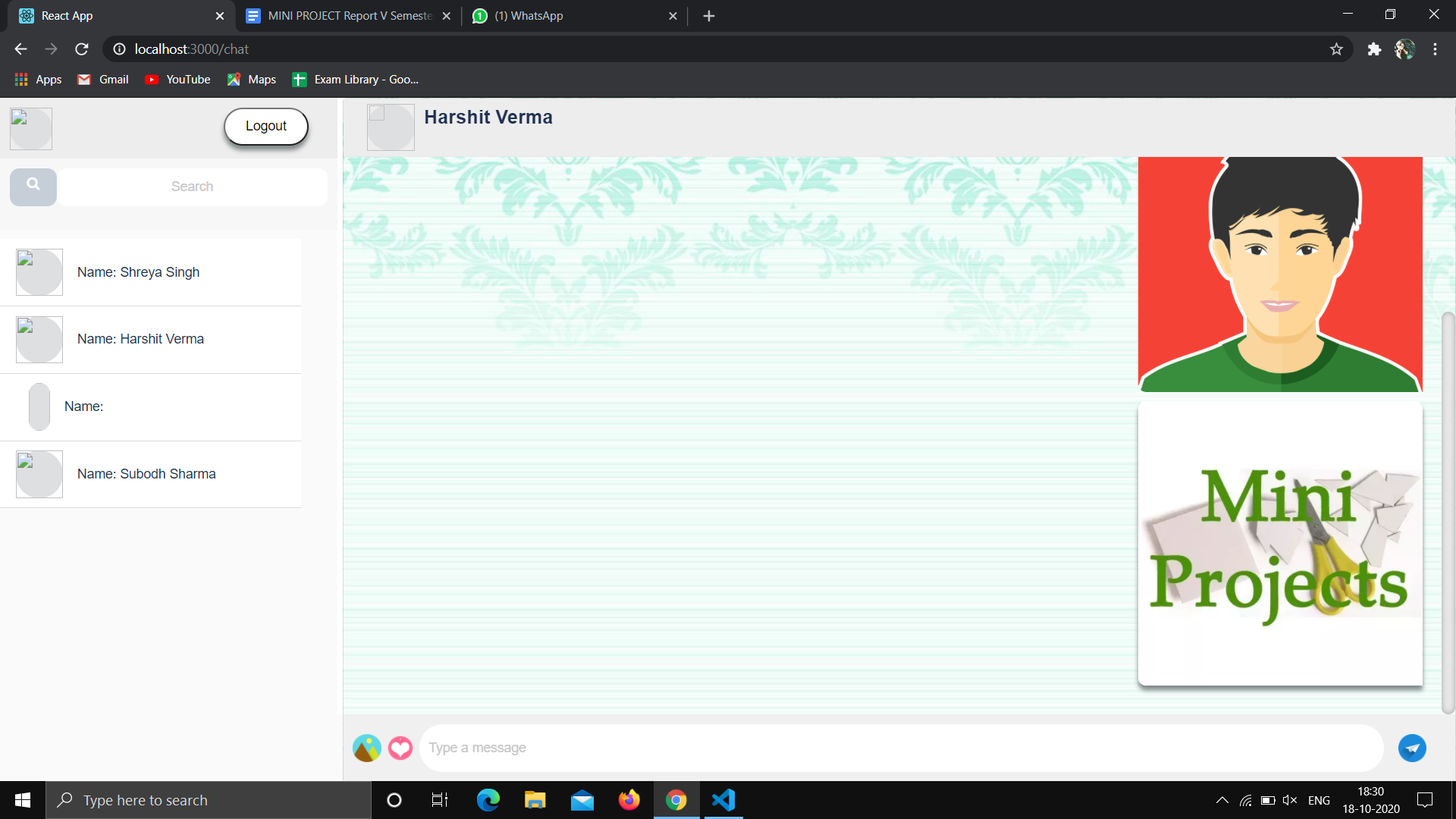
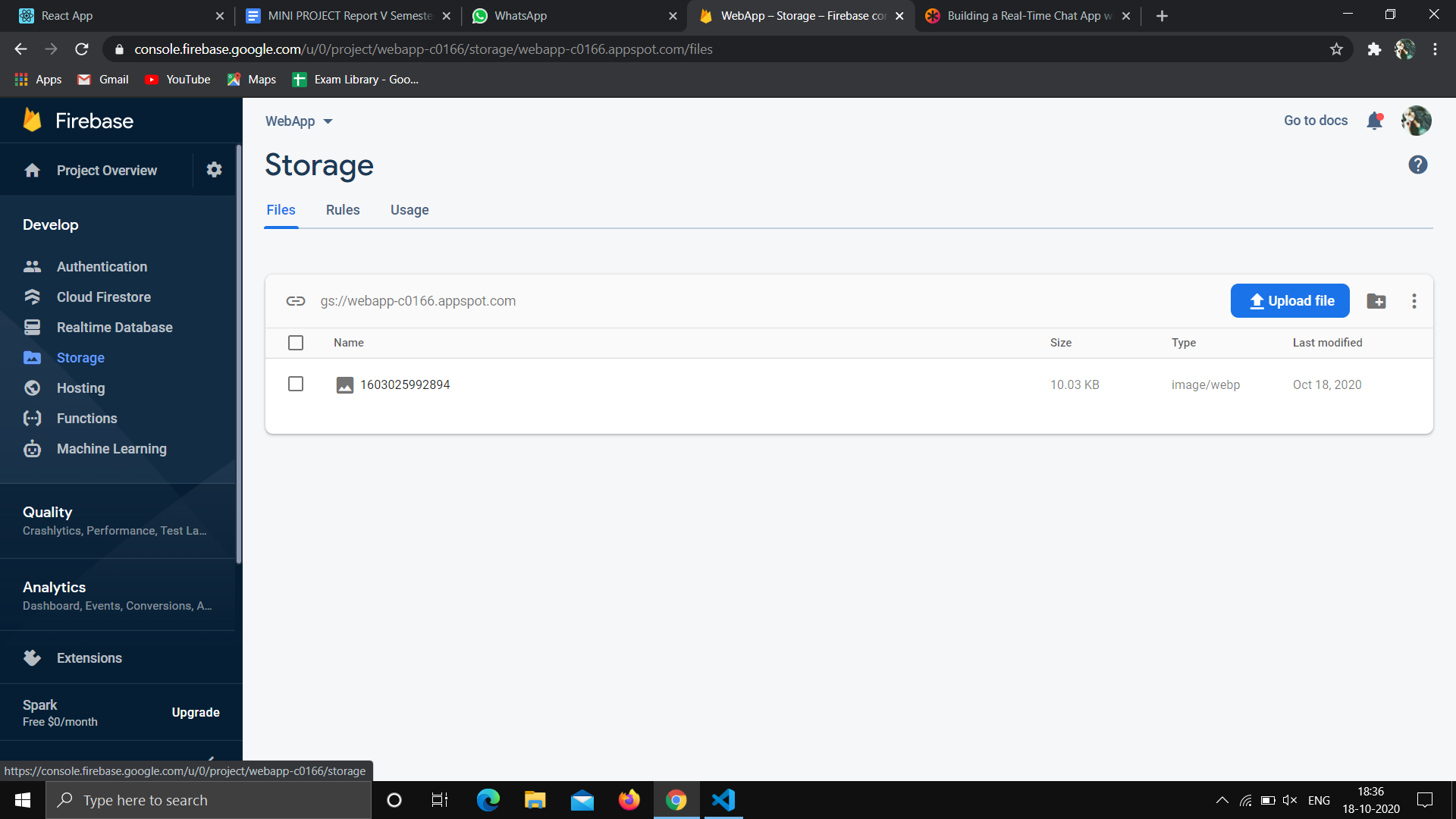


Image which has been sent storing on google firebase.



We can search anyone in contact by typing name:



**References**

* <https://css-tricks.com/building-a-real-time-chat-app-with-react-and-firebase/>
* <https://medium.com/@duytq94/building-web-chat-with-reactjs-and-firebase-e7f9654b661>
* <https://www.youtube.com/playlist?list=PLC3y8-rFHvwgg3vaYJgHGnModB54rxOk3>

**Github Link :**

**Harshit Verma:** [**https://github.com/Harshit8126**](https://github.com/Harshit8126)

**Harshita Katara added as a contributor on Harshit Verma’s Github Mini-Project-1 Repository.**