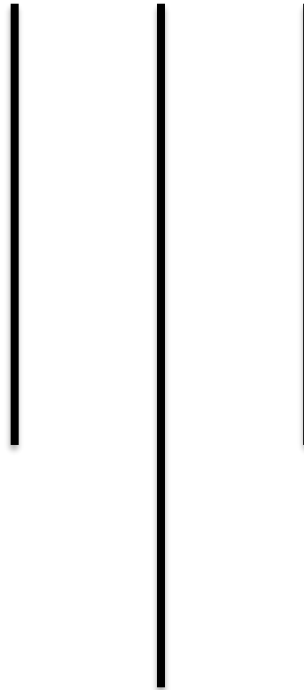




Date: 19/04/2021



**DELHI TECHNOLOGICAL UNIVERSITY,  
(DTU) DELHI**

**SOCIAL MEDIA WEBSITE USING  
JAVASCRIPT/HTML/CSS/REACT.JS  
AND FIREBASE AS THE BACKEND**

**Submitted by: Sonu K. Kushwaha & Ayush Karn**

**Roll: 2K19/CO/383 & 2K19/CO/454**

**Course: Database Management System (CO-202)**

**Submitted to: Mr Manoj Sethi**

## DECLARATION

We hereby declare that project report entitled “SOCIAL MEDIA WEBSITE Using JAVASCRIPT / HTML / CSS / REACT.JS/ FIREBASE “submitted by (**Sonu K Kushwaha & Ayush Karn**) to Delhi Technological University (DTU), Delhi is a record of original work done under the guidance of Mr. **Manoj Sethi** for the course of Database Management System. All the codes and implementations are completely written by us.

Name: Sonu K Kushwaha & Ayush Karn  
Roll No: 2K19/CO/383 & 2K19/CO/454  
Submitted to: Mr. Manoj Sethi

## CERTIFICATE

This is to certify that Sonu K. Kushwaha and Ayush Karn of Computer Science and Engineering Department (COE) having Roll No 2K19/CO/383 & 2K19/CO/454 respectively have successfully completed the project work entitled “SOCIAL MEDIA WEBSITE Using JAVASCRIPT / HTML / CSS / REACT.JS/ FIREBASE” on Database Management System for Fourth Semester which is to be evaluated as the Mid Term Component.

Signature: Sonu K Kushwaha & Ayush Karn

Date: 16-04-2021

## ACKNOWLEDGEMENT

We would like to express my special thanks of gratitude to teacher Mr. Manoj Sethi as well as our college (Delhi Technological University, Delhi) which gave me the golden opportunity to do this wonderful project on the topic “SOCIAL MEDIA WEBSITE Using JAVASCRIPT / HTML / CSS / REACT.JS/ FIREBASE”, which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

## TABLE OF CONTENT

	Page
Chapter 1: Introduction , Languages and Component	6-9
Chapter 2: Related Works in this Field	10-11
Chapter 3: Proposed Work	12
Chapter 4: Implementation and Code	13-27
Chapter 5: Result	28
Chapter 6: Analysis	29
Chapter 7: Conclusion	30
Chapter 8: References	31

# Chapter 1

## Introduction

Social Media plays a crucial role in connecting people and developing relationships, not only with key influencers and journalists covering your company's sector, but also provides a great opportunity to establish customer service by gathering input, answering questions and listening to their feedback.

In this project we have tried to make a social media type of website. In this website we have used React.JS, HTML, CSS for Front end Development and Firebase is used as the Backend for the project.

Following are the functionality that are provided in our project

- Sign Up
- Log In and Log Out
- Posting a Social Media Post
- Add Caption to the Post
- Interactive Interface
- Realtime Comment on a Post
- Firebase Authentication
- Storing Images on Firebase Server

## Introduction to Language and Components Used

### **Firebase:**

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. I have used Firebase in this project for hosting my website and I am planning to use the database feature of Firebase in future for maintaining the database for my app

Complete backend of our project is built upon firebase. From authentication to schema of the database everything is handled by just firebase. Firebase basically provides solution to everything that is needed by a developer for his/her backend. The learning curve of firebase is also not that difficult and that is one of the main reasons we chose firebase as our backend. Firebase also fits perfectly with React.JS making it go hand in hand with each other and making the life much easier. It also gives us a feature to host our website directly on Google's own server and we have used that to host our website on Google's server.

### **CSS3:**

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is used in this project for just the front-end design and plays a very important role in user experience.

## JavaScript:

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

## React.JS:

React is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. I have used react.js for writing my program. In react we have to write code in JSX format and later it gets converted to respective JavaScript by Babel Compiler.

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. We can build encapsulated components that manage their own state, then compose them to make complex UIs.



## **HTML:**

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

## **Node.JS:**

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. I have used node.js for creating a runtime environment for my project

Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

## Chapter 2

### Related Work

During the 3<sup>rd</sup> semester of our college we were introduced to innovative component part of the course and this was completely new area to us. Previously, we didn't have much knowledge about these things and we were surely missing out on so many things. After introduction to the innovative component in our curriculum we were inclined to learn new things and explore new fields that motivated us and we ended up learning about Web Development as it was one of the most common yet most impactful thing related to computer science.

Every time we use Instagram or WhatsApp or any other social media we do that on web. Every time we surf internet to search for something we do that on Web. So, it should be very clear that how big web development is and it is only going to get bigger. During the 3<sup>rd</sup> semester we did a project related to **E Commerce website design** and that is when we got to know about React.JS and Firebase. The learning curve for React and Firebase is not that difficult and anyone with basic coding skill can catch up to it pretty quickly.

During the 4<sup>th</sup> semester we were introduced to DBMS and that is when we had the chance to fully utilize the firebase's backend system. In order to learn the backend implementation using firebase we had to research a bit and learn a lot. Some of the research paper that we read and articles that we read are on the next page.

Singh H S., Singh U. “**Study on Google Firebase for Website Development (The real time database)**”<sup>[1]</sup> International Journal of Engineering Technology Science and Research, Volume 4, Issue 3 March 2017. From this research paper we learnt basics of firebase. Another great paper that we read on firebase was “**Application of Firebase in Android App Development-A Study**”<sup>[2]</sup> by Khawas C., Shah P. published in International Journal of Computer Applications Volume in June 2018

There was also some of the article that we read on firebase. One of the most impactful one was by V. Singh titled “**Introduction to Firebase**”<sup>[5]</sup> published in Medium.com. Firebase’s own documentation was also a great help i.e., Firebase Documentation (<https://firebase.google.com/docs>) A YouTube channel that we learnt other things about react was B. Traversy’s, “React JS Crash Course 2021”<sup>[4]</sup>.

Some of the most famous companies that use React are **Facebook** and its sister company **Instagram**. Firebase is also widely used by companies like **Instacart**, **Twitch** (biggest streaming website), **Alibaba Travel**, **Bitpanda**, etc.

The main reason to use react is because of its ability to refresh specific part of a webpage instead of the entire webpage. For example, when we comment something in Instagram or Facebook only the comment section of the post is refreshed not the entire webpage which is not possible using plain JavaScript. And the one of the main reasons to use Firebase is because of its simplicity and very beginner friendly learning curve.

## Chapter 3

### Proposed Work

In this project we are going to be making a social media website using React.JS/ HTML/ CSS/ JavaScript as the front-end part and we will be using Firebase for all the backend functionality for this project.

Firebase is actually a great choice instead of traditional backend such as SQL, Oracle or other similar Software because it is very new and is managed by google. Firebase is something that is being used in industry. So, learning this would be beneficial to us.

Database will be made inside Firebase's Firestore Database and Authentication will also be done using Firebase only

Firebase actually uses a mixture of SQL and NoSQL based database so it includes goods of both worlds.

We have also hosted the website using Firebase only as Firebase also provides us with a feature to host our webpages on very fast Google's server.

## Chapter 4

### Implementation and Code

This project was implemented using Firebase as our backend to everything that includes Database Design, authentication, storing of the images on Google's firebase server and retrieving the URL that was converted to firebase's URL and fed directly into the Database created by us.

The front-end part of the website was created using React.JS developed by Facebook which is the mixture of JavaScript and HTML also known as JSX.

One of the main benefits of using React.JS as front end is because each of the things can be broken down into simple components and then be used multiple times with just one line of code and changing the values according to the need.

The components that were used in this project are

#### **<Post.js>**

This component that included everything that a Social Media post holds. It included username of the uploader, Caption if there was any, timestamp of when the post was uploaded a PostID to differentiate each post kept in the database.

#### **<ImgUpload.js>**

This component that was used to upload post inside the website and was directly connected to the backend. This component was used at the bottom of the page and by using

this user could upload images to the website as well as videos and then these images were uploaded to firebase's server and then firebase would generate a ImageURL which was stored in the database instead of the image. Using the image URL instead of the image helped us reduce the time taken drastically and made the website more reactive.

### **<App.js>**

This was the main component of our entire project and it is where all other components were called and used. <Post.js> , <ImgUpload.js>, <Index.js> etc were called into this component and used multiple times as desired.

### **<Post.css, ImageUpload.css, App.css, etc>**

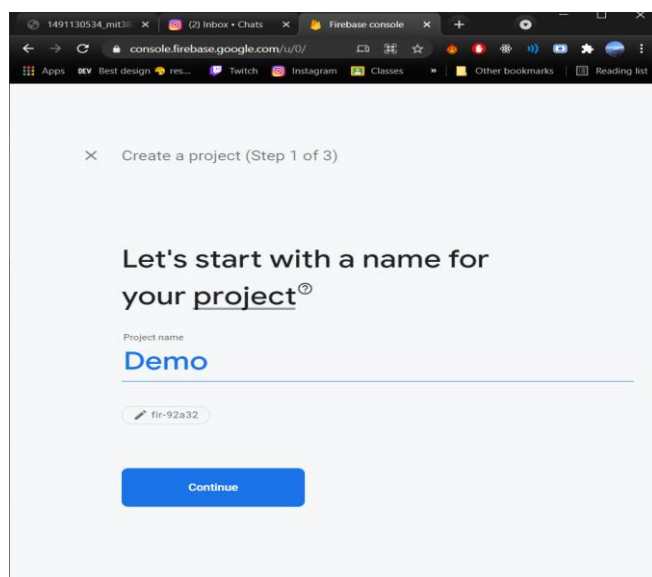
All the CSS components were used just to provide better look to the website which is very important because user interface is very much useful for a website. A website with better UX will always attract more people than with a bad UX one.

Using this component-based implementation with the help of react we were able to drastically decrease the complexity of the code and it was very easier for us to debug as well as modify the code. Using component-based methodology for this project also enabled us to be flexible with the type of schema that we used for this database created inside of the firebase which we will be discussing later on in the report.

## Adding Firebase to Project:

First, we need to add our project to firebase, for which we should follow the below steps and we learnt these steps from “Application of Firebase in Android App Development-A Study” [2]

- 1) Go to <https://console.firebase.google.com/>
- 2) Click on Add Project
- 3) Enter your Project's name
- 4) On next page you can setup Google Firebase Analytics
- 5) On Next page select account for Google Analytics
- 6) Click Create Project



- 7) Click Create Project and Your project will be created
- 8) Go to </> Icon on the top and give your App a nickname
- 9) Check the box for setup firebase hosting
- 10) Click Next, Next and Continue to Console and your project is Finally Created
- 11) Copy the Codes from Config page into Firebase.js and you're good to go.

## Authentication

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more. Firebase Authentication integrates tightly with other Firebase services.

There are basically two ways we can implement authentication either by using FirebaseUI as a complete drop-in auth solution or by using the Firebase Authentication SDK to manually integrate one or several sign-in methods into your app. We have opted for the manual sign-in integration using Authentication SDK in this project.

For other backend it requires weeks even months and 100's of lines of difficult code in order to provide a user-based authentication, Firebase can do this in just 10 lines of code and we just have to use when and where to ask for the user to authenticate them and everything else is handled by Firebase. We just have to import the <Firebase.js> component where we want to provide the authentication feature and we are ready to go.

A part of code that handles authentication is given below



```

const signUp = (event) => {
  event.preventDefault();
  auth.createUserWithEmailAndPassword(email , password)
    .then((authUser) => {
      return authUser.user.updateProfile({
        displayName: username
      })
    })
    .catch((error) => alert(error.message));
  setOpen(false);
}

const signIn = (event) => {
  event.preventDefault();
  auth.signInWithEmailAndPassword(email , password)

    .catch((error) => alert(error.message));
  setOpenSignIn(false);
}

```

## Database Schema:

We have used firebase's Firestore Database to implement the Database for our project. It is a NoSQL based Database and contains these components inside of them.

A Collection called **Post** that holds all the posts that are uploaded to the website.

Each **Post** has following documents in them:

**caption:** "string" this stores the caption for that post

**imgUrl:** "string" ImageURL is stored in this part

**timestamp:** "time" Firebase's server time is stored for ordering the posts and maintaining uniformity.

**username:** "string" username of the uploader is stored

Each **post** also has another **sub collection** in order to store all the comments that are done on that specific **post**. This comment section is further divided into sub documents which are as follows:

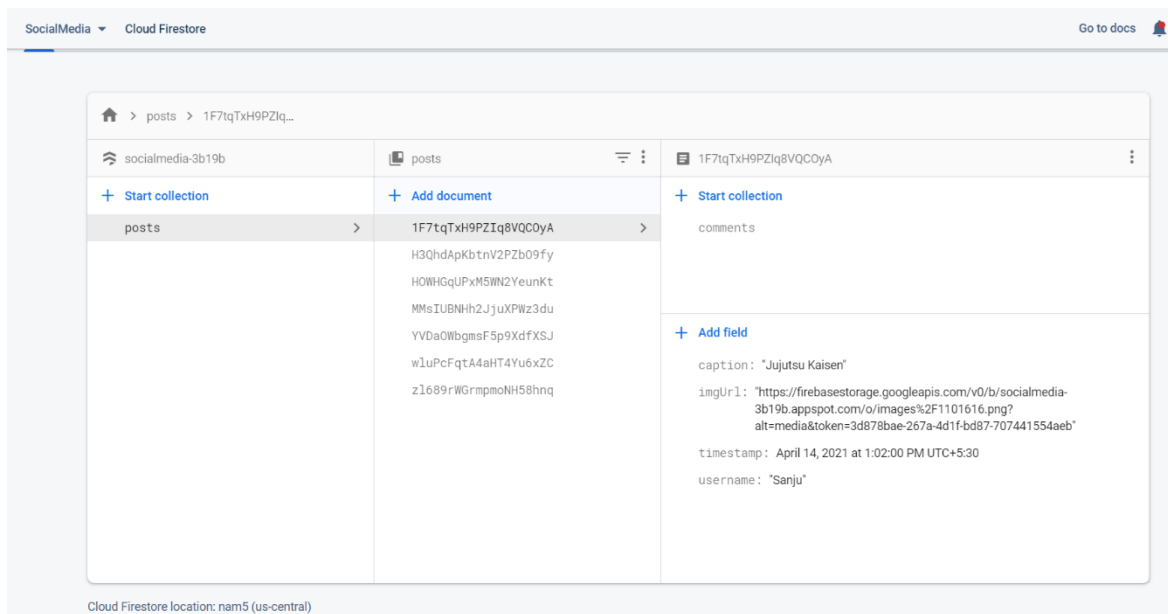
**text:** “string” for storing the comment on the post.

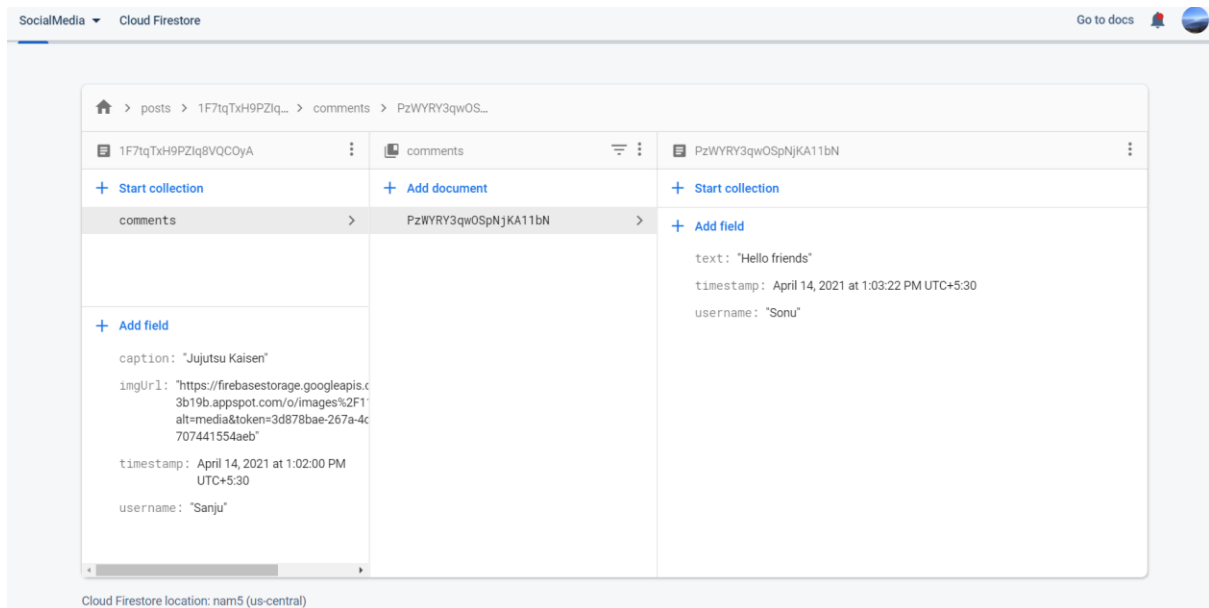
**timestamp:** “time” for storing the time of uploading the comment and for ordering purpose this concept was used from “A Research Paper on a Pet-Friendly Application using Flutter and Firebase” [3]

**username:** “string” for storing username of the person commenting on the post.

Along with all of these each of the document is automatically provided a DocumentID in order to be made unique and this DocumentID is used as a primary key for the database and also as a foreign key for relating to some other part of database.

A picture of this database is attached below for the reference.





Automatically All of the code is not possible to be given in this report so I am attaching most important component's codes

## Code:

### <App.js>

```
import React, {useState, useEffect} from "react";
import './App.css';
import Post from './post.js';
import {db, auth} from './FireBase.js';
import {makeStyles} from '@material-ui/core/styles';
import Modal from '@material-ui/core/Modal';
import {Button, Input} from '@material-ui/core';
import ImgUpload from './ImgUpload';
import InstagramEmbed from 'react-instagram-embed';
//import firebase from 'firebase';

function getModalStyle() {
  const top = 50;
  const left = 50;

  return {
    top: `${top}%`,
    left: `${left}%`,
    transform: `translate(-${top}%, -${left}%)`,
  };
}
```

```

    };
  }

const useStyles = makeStyles((theme) => ({
  paper: {
    position: 'absolute',
    width: 400,
    backgroundColor: theme.palette.background.paper,
    border: '2px solid #000',
    boxShadow: theme.shadows[5],
    padding: theme.spacing(2, 4, 3),
  },
}));

function App() {
  const classes = useStyles();
  const [modalStyle] = useState(getModalStyle);
  const [posts, setPosts] = useState([]);
  const [open, setOpen] = useState(false);
  const [openSignIn, setOpenSignIn] = useState(false);
  const [email, setEmail] = useState('');
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [user, setUser] = useState(null);
  useEffect(() => {
    const unsubscribe= auth.onAuthStateChanged((authUser) => {
      if(authUser){
        console.log(authUser);
        setUser(authUser);
        if(authUser.displayName){

        }else{
          return authUser.updateProfile({
            displayName: username,
          });
        }
      }else{
        setUser(null);
      }
    })
    return () => {
      unsubscribe();
    }
  },[user, username]);
  useEffect(() => {
    db.collection('posts').orderBy('timestamp','desc').onSnapshot((snapshot) => {
      setPosts(

```

```

        snapshot.docs.map(doc=>({
            id: doc.id,
            post : doc.data()
        }))
    ));
})
}, []);

const signUp = (event) => {
    event.preventDefault();
    auth.createUserWithEmailAndPassword(email , password)
    .then((authUser) => {
        return authUser.user.updateProfile({
            displayName: username
        })
    })
    .catch((error) => alert(error.message));
    setOpen(false);
}

const signIn = (event) => {
    event.preventDefault();
    auth.signInWithEmailAndPassword(email , password)

    .catch((error) => alert(error.message));
    setOpenSignIn(false);
}

return (
    <div className="app">

        <Modal
            open={open}
            onClose={()=> setOpen(false)}>
            <div style={modalStyle} className={classes.paper}>
                <form className = "app__signUp">
                    <center>
                        <div className="header__title">
                            <p>Social Media</p>
                        </div>
                    </center>
                    <Input
                        type='text'
                        placeholder='username'
                        value={username}
                        onChange={(e)=>setUsername(e.target.value)}
                    />
                    <Input

```

```

        type='email'
        placeholder='email'
        value={email}
        onChange={(e)=>setEmail(e.target.value)}
      />
      <Input
        type='password'
        placeholder='password'
        value={password}
        onChange={(e)=>setPassword(e.target.value)}
      />
      <Button type="submit"   onClick={signUp}>Sign Up</Button>

    </form>
  </div>

</Modal>
<Modal
  open={openSignIn}
  onClose={()=> setOpenSignIn(false)}>
  <div style={modalStyle} className={classes.paper}>
    <form className = "app__signUp">
      <center>
        <div className="header__title">
          <p>Social Media</p>
        </div>

        </center>
        <Input
          type='email'
          placeholder='email'
          value={email}
          onChange={(e)=>setEmail(e.target.value)}
        />
        <Input
          type='password'
          placeholder='password'
          value={password}
          onChange={(e)=>setPassword(e.target.value)}
        />
        <Button type="submit"   onClick={signIn}>Sign In</Button>

      </form>
    </div>

  </Modal>
  <div className="app__header">

```

```

<div className="header__title">
  <p>Social Media</p>
</div>
{user ? (
  <Button onClick={()=>auth.signOut()}>Logout</Button>
): (
  <div className = "app__loginContainer">
    <Button onClick={()=>setOpen(true)}>Sign Up</Button>
    <Button onClick={()=>setOpenSignIn(true)}>Sign In</Button>
  </div>
)}
</div>

<div className= 'app__posts'>
  <div className = "app__postsLeft">

    {posts.map( ({id , post}) => (
      <Post
        key = {id}
        postId={id}
        user={user}
        username={post.username}
        caption={post.caption}
        imgUrl={post.imgUrl}
      />
    ))}
  </div>
  <div className = "app_postsRight">
    <InstagramEmbed
      url='https://instagr.am/p/Zw9o4/'
      maxWidth={320}
      hideCaption={false}
      containerTagName='div'
      protocol=''
      injectScript
      onLoading={() => {}}
      onSuccess={() => {}}
      onAfterRender={() => {}}
      onFailure={() => {}}
    />

  </div>
</div>

  {user?.displayName ? (<ImgUpload username= {user.displayName} />):( <h3>Sorry you need to login !!</h3>)}
</div>

```

```

    );
  }
  export default App;

```

## <Firebase.js>

```

// For Firebase JS SDK v7.20.0 and later, measurementId is optional
import firebase from 'firebase';

const FirebaseApp = firebase.initializeApp({
  apiKey: "AIzaSyDHHjcUcQYc5IzHFMdLSZ2KmJ30QT92ERU",
  authDomain: "socialmedia-3b19b.firebaseio.com",
  projectId: "socialmedia-3b19b",
  storageBucket: "socialmedia-3b19b.appspot.com",
  messagingSenderId: "259452570231",
  appId: "1:259452570231:web:eb2d8affc7ed87a8581472",
  measurementId: "G-Y8Z78YET77"
});

const db = FirebaseApp.firestore();
const auth = firebase.auth();
const storage = firebase.storage();

export { db, auth, storage };

```

## <ImgUpload.js>

```

import React , { useState } from 'react';
import {Button} from '@material-ui/core';
import {db , storage} from './FireBase.js';
import firebase from 'firebase';
import './ImageUpload.css';

function ImgUpload({username}) {
  const [caption , setCaption] = useState('');
  const [image, setImage] = useState(null);
  //const [url, setUrl] = useState(" ");
  const [progress,setProgress] = useState(0);

  const handleChange = (e) =>{
    if(e.target.files[0]){
      setImage(e.target.files[0]);
    }
  }

  const handleUpload = () =>{

```



```

const uploadTask = storage.ref(`images/${image.name}`).put(image);
uploadTask.on(
  "state_changed",
  (snapshot) => {
    const progress = Math.round(
      (snapshot.bytesTransferred / snapshot.totalBytes) * 100
    );
    setProgress(progress);
  },
  (error) =>{
    console.log(error);
    alert(error.message);
  },
  () => {
    storage
      .ref("images")
      .child(image.name)
      .getDownloadURL()
      .then(url => {
        db.collection("posts").add({
          timestamp : firebase.firestore.FieldValue.serverTimestamp(),
          caption : caption,
          imgUrl : url,
          username : username
        });

        setProgress(0);
        setCaption('');
        setImage(null);
      });
  }
)
}
return (
  <div className='ImageUpload'>
    <progress className= 'ImageUpload__progress' value ={progress} max
    ="100" />
    <input type = "text" placeholder = "Enter a Caption" onChange={event => setCaption(event.target.value)} />
    <input type = "file" onChange = {handleChange} />
    <Button onClick={handleUpload} variant='contained' color='secondary' >
      Upload
    </Button>
  </div>
)
}

```

```
export default ImgUpload
```

<post.js>

```
import React , {useState ,useEffect} from 'react';
import './post.css';
import Avatar from "@material-ui/core/Avatar";
import {db } from './FireBase.js';
import firebase from 'firebase';

function Post({postId,user,username,caption,imgUrl}) {
  const [comments, setComments] = useState([]);
  const [comment, setComment] = useState('');
  useEffect(() => {
    let unsubscribe;
    if(postId){
      unsubscribe = db
        .collection("posts")
        .doc(postId)
        .collection("comments")
        .orderBy("timestamp","asc")
        .onSnapshot((snapshot)=>{
          setComments(snapshot.docs.map((doc)=>doc.data()))
        })
    }
    return () => {
      unsubscribe();
    }
  }, [postId]);

  const postComment = (event) =>{
    event.preventDefault();
    db.collection("posts").doc(postId).collection("comments").add({
      text: comment,
      username: user.displayName,
      timestamp: firebase.firestore.FieldValue.serverTimestamp()
    });
    setComment('');
  }

  return (
    <div className="post">
      <div className="post__header">
        <Avatar className="post__avatar"
          src="/static/images/avatar/jpg"
          alt={username}/>
        <h3>{username}</h3>
      </div>
    </div>
  )
}
```

```

    </div>

    <img className="post__image" alt="abc" src={imgUrl}></img>

    <h4 className="post__text"><strong>{username}</strong> {caption}<
/h4>
    <div className = "post__comments">
      {
        comments.map((comment) => (
          <p>
            <strong>{comment.username}</strong> {comment.text}
          </p>
        ))
      }
    </div>
    {user && (
      <form className = "post__commentsBox">
        <input
          className="post__input"
          type="text"
          placeholder="Add a comment.."
          value={comment}
          onChange={(e) => setComment(e.target.value)}
        />
        <button
          className="post__button"
          disabled={!comment}
          type="submit"
          onClick={postComment} >Post
        </button>

        </form>
      )}
    </div>
  )
}

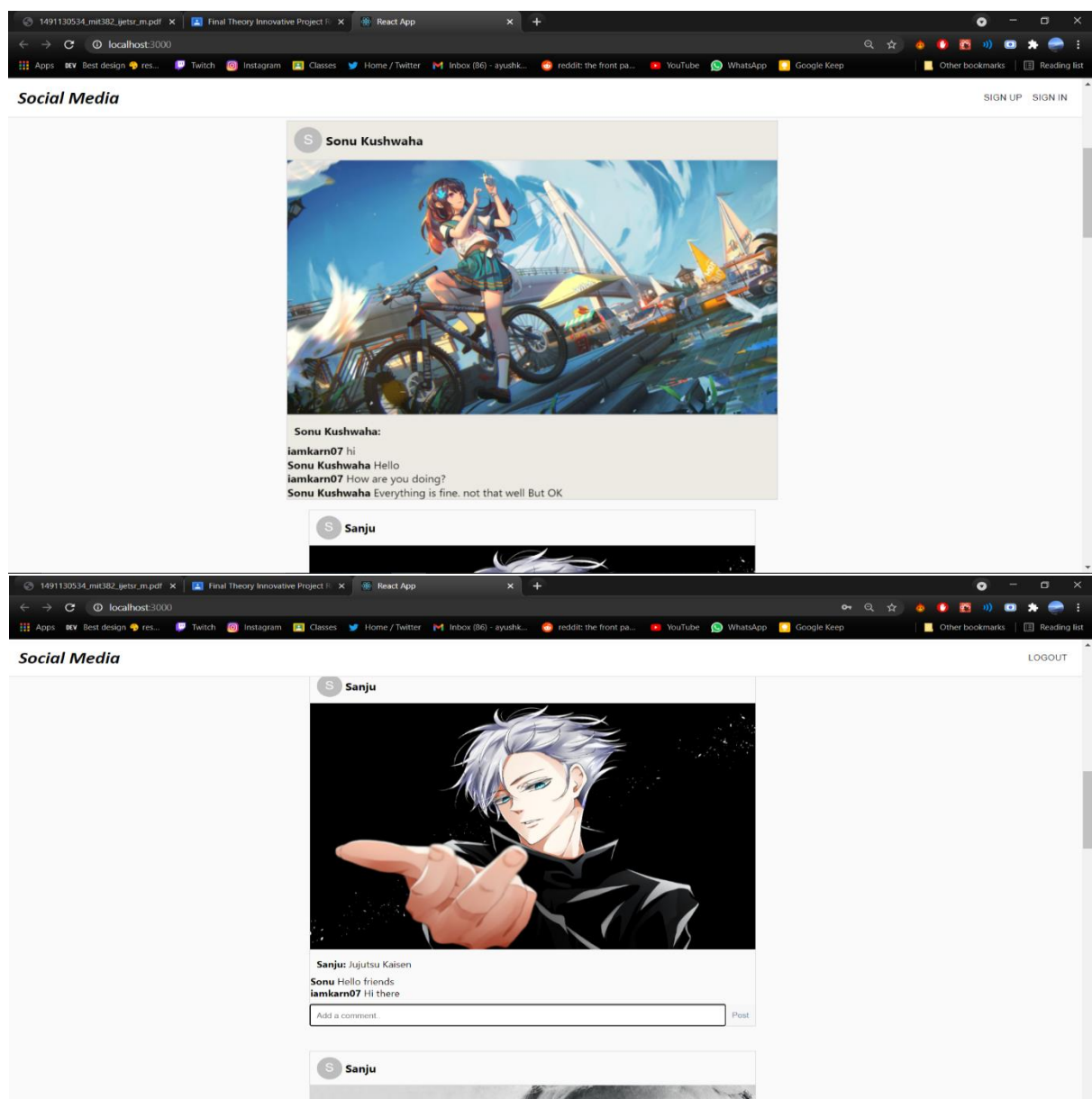
export default Post;

```

## Chapter 5

### Result

The results that we got from learning React.JS and Firebase for just 3-4 months were very promising and we definitely could learn a lot more if we had more time up our sleeves. But because we had 5 other projects to work on we couldn't do everything that we wanted to. Some of the results screenshots are attached below. We were also able to Host our website on <https://socialmedia-3b19b.web.app> if you want to check this out.



## Chapter 6

### Analysis

After analyzing what we have achieved from just 3-4 months of learning React.JS and Firebase we are pretty sure we could now work on bigger projects or we could just continue to add more features in this project. We could add chatting functionality to this project. We could add more types of post that could be posted on the website that includes Audio, Video, text-based posts, etc. We could also add more things such as following/ adding people as friends, keeping track of likes, dislikes, etc.

One innovative thing that we did in this project is that, by using react instead of other similar framework we saved a lot of data while browsing the website. For example: when a user comments on a post or when a user uploads a post with the help of react we are just refreshing that specific part of DOM and not the whole DOM. Because of this we don't have to fetch everything in the database but we could just fetch the part that is recently modified or added and we can save a lot of time and processing power.

Another innovative thing that we did in this project was using URL for image instead of the actual image inside of the database. This helped us reduce the size of the database. We also used firebase compression algorithm in order to reduce the size of the image and then stored that reduced sized image with no noticeable change in quality inside Firebase and used the URL for the same inside our database.

## Chapter 7

### Conclusion

From this project we would like to conclude that we have actually learnt a great deal about web development and a guy who was not at all interested in building web-based application I have actually come to like it and I will be making a few more projects on the web development. This innovative part that was added to our curriculum could prove to be very useful in future as we are trying to learn new things on our own and self-taught things are proven to be more effective as well as more useful for a person.

Not to forget that this Social Media website also has very good uses in today's world.

- 1) Can Broaden Your Brand if used properly for advertisement
- 2) Helps us connect with friends and family
- 3) Helps us meet new people.
- 4) Helps us stay in touch with everyone in a socially distant way and we all know how much necessary that is especially because of the pandemic we are in.

Last but not the least is that this opportunity to make something helped us learn a lot of things that we would continue to learn more about.

## Chapter 8

### References

- [1] Singh H S., Singh U. “Study on Google Firebase for Website Development (The real time database)” International Journal of Engineering Technology Science and Research, Volume 4, Issue 3 March 2017
  
- [2] Khawas C., Shah P. “Application of Firebase in Android App Development-A Study” International Journal of Computer Applications Volume 179(46): June 2018
  
- [3] Yeshwin A., Sahoo A., Shoby P. “A Research Paper on a Pet-Friendly Application using Flutter and Firebase” International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 8 Issue XII Dec 2020
  
- [4] B. Traversy, “React JS Crash Course 2021” Jan 2021. [YouTube]
  
- [5] V. Singh “Introduction to Firebase” Medium.com Dec 2018 [Medium.com]
  
- [6] Firebase Documentation (<https://firebase.google.com/docs>)