Instagram->

If text extended

text-overflow: ellipsis;

overflow: hidden;

white-space: nowrap;

Firebase store

Hosting/deploy

Login

**Firbase db->**

Create on firebase

Config => firebase.js

import { db,auth} from "./firebase";

useEffect(() => {

db.collection("posts")

.orderBy("timestamp", "desc")

.onSnapshot((snapshot) =>

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

);

}, []);

{posts.map(({ id, post }) => (

<Post

user={user}

key={id}

postId={id}

username={post.username}

caption={post.caption}

imageUrl={post.imageUrl}

>Useeffect condition if)

>cleanupaction

**Login->**

Material ui => modal

import { Button, Avatar, makeStyles, Modal, Input } from "@material-ui/core";

Styling=>

function getModalStyle() {

const top = 50;

const left = 50;

return {

height: "300px",

top: `${top}%`,

left: `${left}%`,

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

};

}

const useStyles = makeStyles((theme) => ({

paper: {

position: "absolute",

width: 400,

height: 200,

backgroundColor: theme.palette.background.paper,

border: "2px solid #000",

boxShadow: theme.shadows[5],

padding: theme.spacing(2, 4, 3),

},

}));

const classes = useStyles();

const [modalStyle] = useState(getModalStyle);

const [posts, setPosts] = useState([]);

const [username, setUsername] = useState("");

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const [open, setOpen] = useState(false);

const [user, setUser] = useState(null);

useEffect(() => {

const unsubscribe = auth.onAuthStateChanged((authUser) => {

if (authUser) {

// user is logged in...

console.log(authUser);

setUser(authUser);

if (authUser.displayName) {

// dont update username

} else {

return authUser.updateProfile({

displayName: username,

});

}

} else {

setUser(null);

}

});

return () => {

unsubscribe();

};

}, [user, username]);

const handleLogin = (e) => {

e.preventDefault();

auth

.signInWithEmailAndPassword(email, password)

.catch((error) => alert(error.message));

setOpen(false);

};

const handleRegister = (e) => {

e.preventDefault();

auth

.createUserWithEmailAndPassword(email, password)

.catch((error) => alert(error.message));

setRegisterOpen(false);

};

<Modal open={open} onClose={() => setOpen(false)}>

<div style={modalStyle} className={classes.paper}>

<form className="app\_\_login">

<center>

<img

className="app\_\_headerImage"

src="instanger.png"

alt=""

/>

</center>

<Input

placeholder="email"

type="text"

value={email}

onChange={(e) => setEmail(e.target.value)}

/>

<Input

placeholder="password"

type="password"

value={password}

onChange={(e) => setPassword(e.target.value)}

/>

<Button onClick={handleLogin}>Login</Button>

</form>

</div>

</Modal>

<Modal open={registerOpen} onClose={() => setRegisterOpen(false)}>

<div style={modalStyle} className={classes.paper}>

<form className="app\_\_login">

<center>

<img

className="app\_\_headerImage"

src="instanger.png"

alt=""

/>

</center>

<Input

type="text"

placeholder="username"

value={username}

onChange={(e) => setUsername(e.target.value)}

/>

<Input

placeholder="email"

type="text"

value={email}

onChange={(e) => setEmail(e.target.value)}

/>

<Input

placeholder="password"

type="password"

value={password}

onChange={(e) => setPassword(e.target.value)}

/>

<Button onClick={handleRegister}>Register</Button>

</form>

</div>

</Modal>

=>signin signout

const [open, setOpen] = useState(false);

const [registerOpen, setRegisterOpen] = useState(false);

{user?.displayName ? (

<div className="app\_\_headerRight">

<Button onClick={() => auth.signOut()}>Logout</Button>

<Avatar

className="app\_\_headerAvatar"

alt={user.displayName}

src={`https://avatars.dicebear.com/api/human/${urlUsername}.svg?mood[]=happy`}

/>

</div>

) : (

<form className="app\_\_loginHome">

<Button onClick={() => setOpen(true)}>Login</Button>

<Button onClick={() => setRegisterOpen(true)}>Sign Up</Button>

</form>

)}

**Drupal**

Image upload->

ImageUpload.js

import { storage, db } from "./firebase";

const [image, setImage] = useState(null);

const [progress, setProgress] = useState(0);

const [caption, setCaption] = useState("");

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) => {

// progress function ...

const progress = Math.round(

(snapshot.bytesTransferred / snapshot.totalBytes) \* 100

);

setProgress(progress);

},

(error) => {

// Error function ...

console.log(error);

},

() => {

// complete function ...

storage

.ref("images")

.child(image.name)

.getDownloadURL()

.then((url) => {

setUrl(url);

// post image inside db

db.collection("posts").add({

imageUrl: url,

caption: caption,

username: username,

timestamp: firebase.firestore.FieldValue.serverTimestamp(),

});

setProgress(0);

setCaption("");

setImage(null);

});

}

);

};

<div className="imageupload">

<progress className="imageupload\_\_progress" value={progress} max="100" />

<Input

placeholder="Enter a caption"

value={caption}

onChange={(e) => setCaption(e.target.value)}

/>

<div>

<input type="file" onChange={handleChange} />

<Button className="imageupload\_\_button" onClick={handleUpload}>

Upload

</Button>

</div>

<br />

</div>

Instagram enbed->

npm i react-instagram-embed

comment functionality->

we should have individual post id

+ start collection

import { db } from "./firebase";

import firebase from "firebase";

const [comments, setComments] = useState([]);

const [comment, setComment] = useState("");

useEffect(() => {

let unsubscribe;

if (postId) {

unsubscribe = db

.collection("posts")

.doc(postId)

.collection("comments")

.onSnapshot((snapshot) => {

setComments(snapshot.docs.map((doc) => doc.data()));

});

}

return () => {

unsubscribe();

};

}, [postId]);

<form className="post\_\_commentBox">

<input

className="post\_\_input"

type="text"

placeholder="Add a comment..."

value={comment}

onChange={(e) => setComment(e.target.value)}

/>

<button

disabled={!comment}

className="post\_\_button"

type="submit"

onClick={postComment}

>

Post

</button>

</form>

<div className="post\_\_comments">

{comments.map((comment) => (

<p>

<b>{comment.username}</b> {comment.text}

</p>

))}

</div>

const postComment = (e) => {

e.preventDefault();

db.collection("posts").doc(postId).collection("comments").add({

text: comment,

username: user.displayName,

timestamp : firebase.firestore.FieldValue.serverTimestamp()

});

setComment("");

};