



Blog Online

Author: Cao Hoang Quy




Tech/Tools

- Visual Studio Code.
- Firebase Authentication.
- Firestore.
- Firebase Storage.
- ReactJS.
- Material UI.

package.json M X

blog-react-app > package.json > {} dependencies

```
1 {
2   "name": "blog-react-app",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@material-ui/core": "^4.11.3",
7     "@material-ui/icons": "^4.11.2",
8     "@testing-library/jest-dom": "^5.11.10",
9     "@testing-library/react": "^11.2.6",
10    "@testing-library/user-event": "^12.8.3",
11    "draft-convert": "^2.1.10",
12    "draft-js": "^0.11.7",
13    "draftjs-to-html": "^0.9.1",
14    "firebase": "^8.3.2",
15    "html-to-draftjs": "^1.5.0",
16    "react": "^17.0.2",
17    "react-dom": "^17.0.2",
18    "react-draft-wysiwyg": "^1.14.5",
19    "react-html-parser": "^2.0.2",
20    "react-router-dom": "^5.2.0",
21    "react-scripts": "4.0.3",
22    "web-vitals": "^1.1.1"
23  },
24  "scripts": {
25    "start": "react-scripts start",
26    "build": "react-scripts build",
27    "test": "react-scripts test",
28    "eject": "react-scripts eject"
29  },
30  "eslintConfig": {
31    "extends": [
```



Module of ReactJS

Upload A image

```
const handleUpload = () => {
  if (image) {
    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) => {
        alert(error.message);
      },
      () => {
        storage
          .ref("images")
          .child(image.name)
          .getDownloadURL()
          .then(url => {
            setLinkImage(url);
            setImage(null);
            setProgress(0);
          });
      }
    );
  } else {
    alert('Pls Input image!');
  }
}
```

Create a body of Post

```
const onEditorStateChange = (editorState) => {  
  setEditorState(editorState);  
  convertContentToHTML();  
}  
  
const convertContentToHTML = () => {  
  let currentContentAsHTML = draftToHtml(convertToRaw(editorState.getCurrentContent()));  
  setConvertedContent(currentContentAsHTML);  
}
```



Add a Post to collection in Firestore

```
const postBlog = () => {  
  db  
    .collection("posts").add({  
      username: user.displayName,  
      title: title,  
      body: convertedContent,  
      timestamp: firebase.firestore.FieldValue.serverTimestamp(),  
      date: new Date()  
    })  
    .catch((error) => alert(error.message));  
  setTitle('');  
  setEditorState(() => EditorState.createEmpty());  
  setConvertedContent(null);  
  setImage(null);  
  setLinkImage('');  
}
```



Get List Post from DB

```
useEffect(() => {  
  function getData() {  
    db.collection('posts').orderBy('timestamp', 'desc').onSnapshot(snapshot => {  
      setPosts(snapshot.docs.map(doc => ({  
        id: doc.id,  
        post: doc.data()  
      })))  
    })  
  }  
  getData();  
}, []);
```



Get all comments and add a comment to a post



```
useEffect(() => {
  let unsubscribe;
  if (postId) {
    unsubscribe = db
      .collection("posts")
      .doc(postId)
      .collection("comments")
      .orderBy('timestamp', 'desc')
      .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('');
}
```


Sign Up and Sign In

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

```
const signIn = (event) => {
  event.preventDefault();
  auth.signInWithEmailAndPassword(email, password)
    .catch((error) => alert(error.message))
  setOpenSignIn(false);
}
```

Authentication

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

useEffect(() => {
  auth.onAuthStateChanged((authUser) => {
    if (authUser) {
      console.log(authUser);
      setUser(authUser);
    } else {
      setUser(null);
    }
  })
}, [user, username]);
```