TUGAS MINGGU 11

Pemrograman Berbasis Framework

“Firebase Login”

Oleh:

Daffa Cahyo Alghifari 1941720104 (05)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

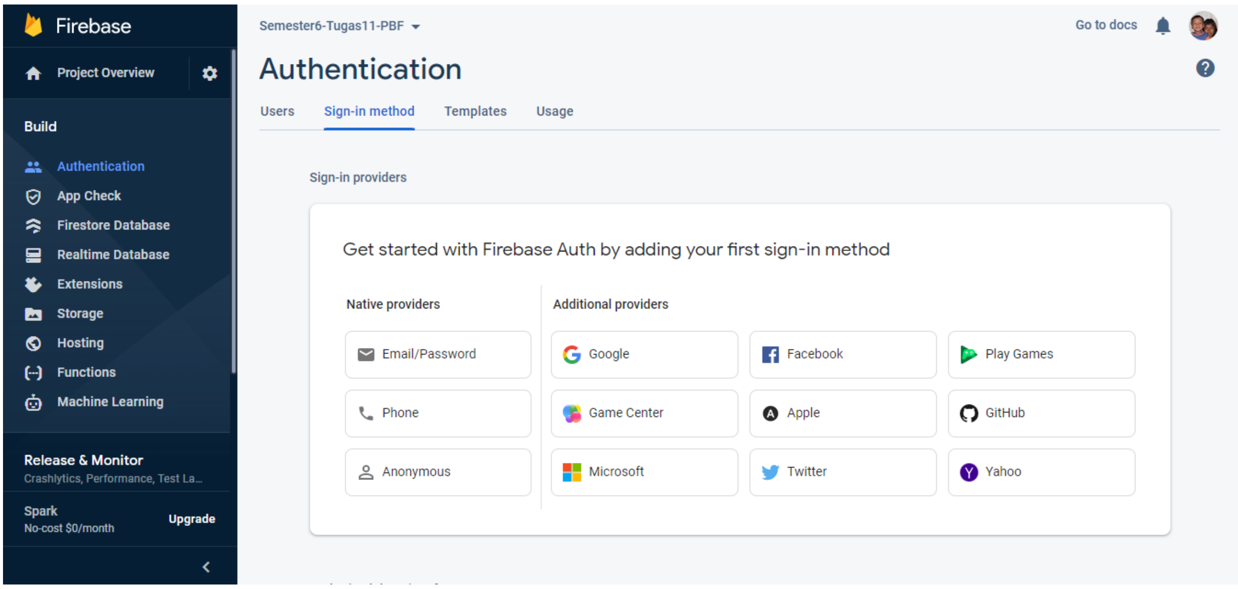
POLITEKNIK NEGERI MALANG

Praktikum FIREBASE LOGIN

Firebase adalah serangkaian alat (bantu) untuk developer aplikasi, namun bukan hanya untuk developer aplikasi Android. Ini untuk developer aplikasi iOS dan juga developer aplikasi web. Akan tetapi, karena kursus ini adalah tentang development Android, pelajaran ini hanya membahas tentang cara menggunakan Firebase dengan aplikasi Android.

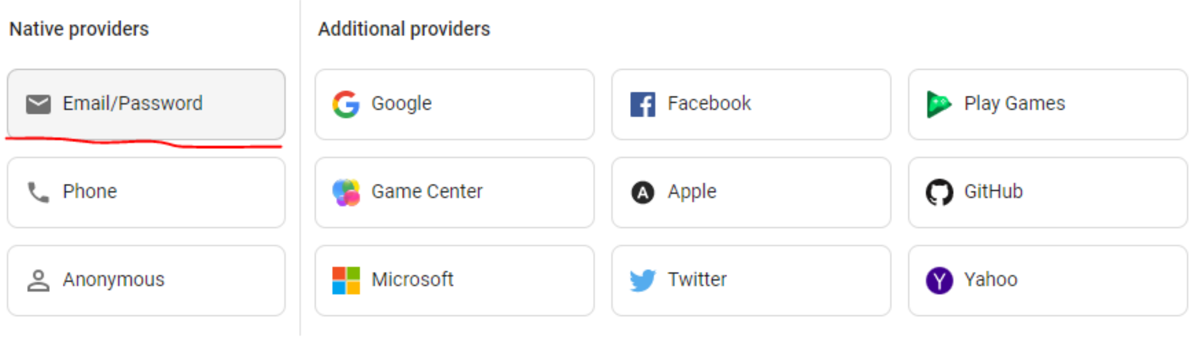
Sebagai developer Android, Anda menggunakan Android Studio untuk membangun aplikasi, namun Anda bisa menggunakan Firebase untuk menambahkan fitur ke aplikasi, mendapatkan pengguna aplikasi yang lebih luas, menguji aplikasi, menghasilkan pendapatan dari aplikasi, dan mendapatkan analisis tentang penggunaan aplikasi tersebut.

1. Setting Firebase Console Login ke firebase console dan add a new project. Pilih authentification dan pilih tab menu Set up sign in method

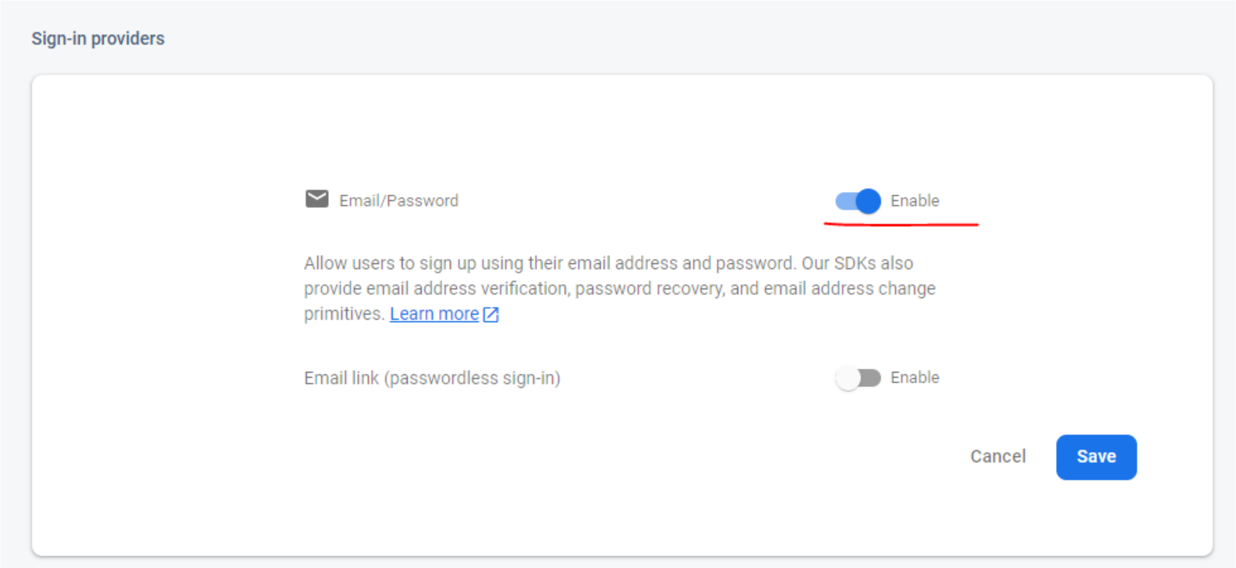


Get started with Firebase Auth by adding your first sign-in method

Pilih ini.



Enable dan save.



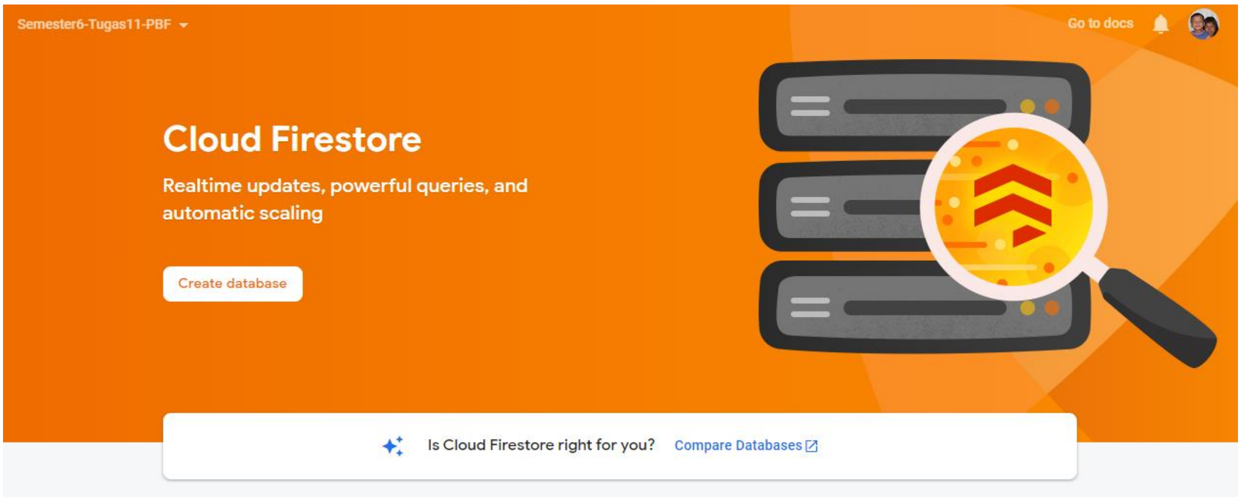
Sign-in providers

Email/Password ● Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. Learn more

Email link (passwordless sign-in) ○ Enable

2. Setelah enabling athentification pilih menu Database dan pilih create database.

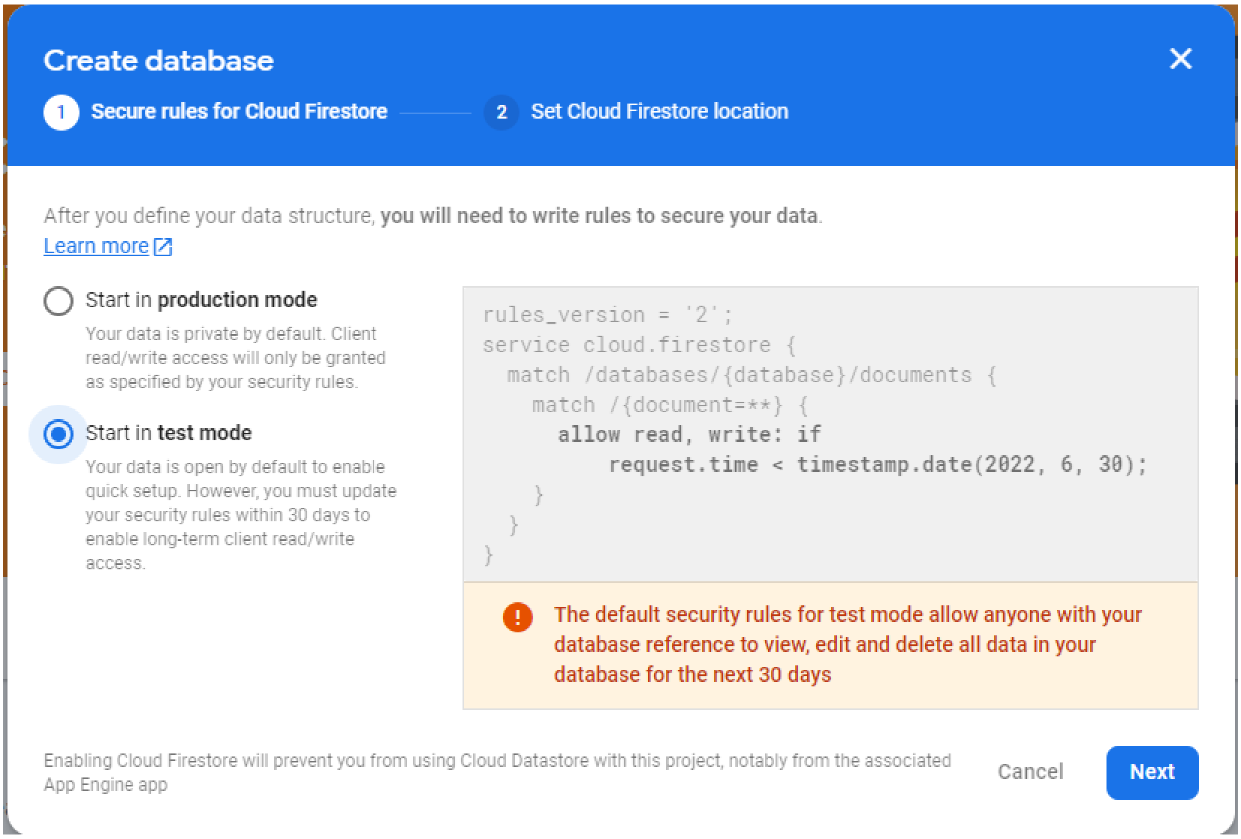


Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

Is Cloud Firestore right for you? Compare Databases

3. Select Start in test mode.



Create database

✕
1 Secure rules for Cloud Firestore
2 Set Cloud Firestore location

After you define your data structure, **you will need to write rules to secure your data**.
Learn more

○ Start in **production mode**
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

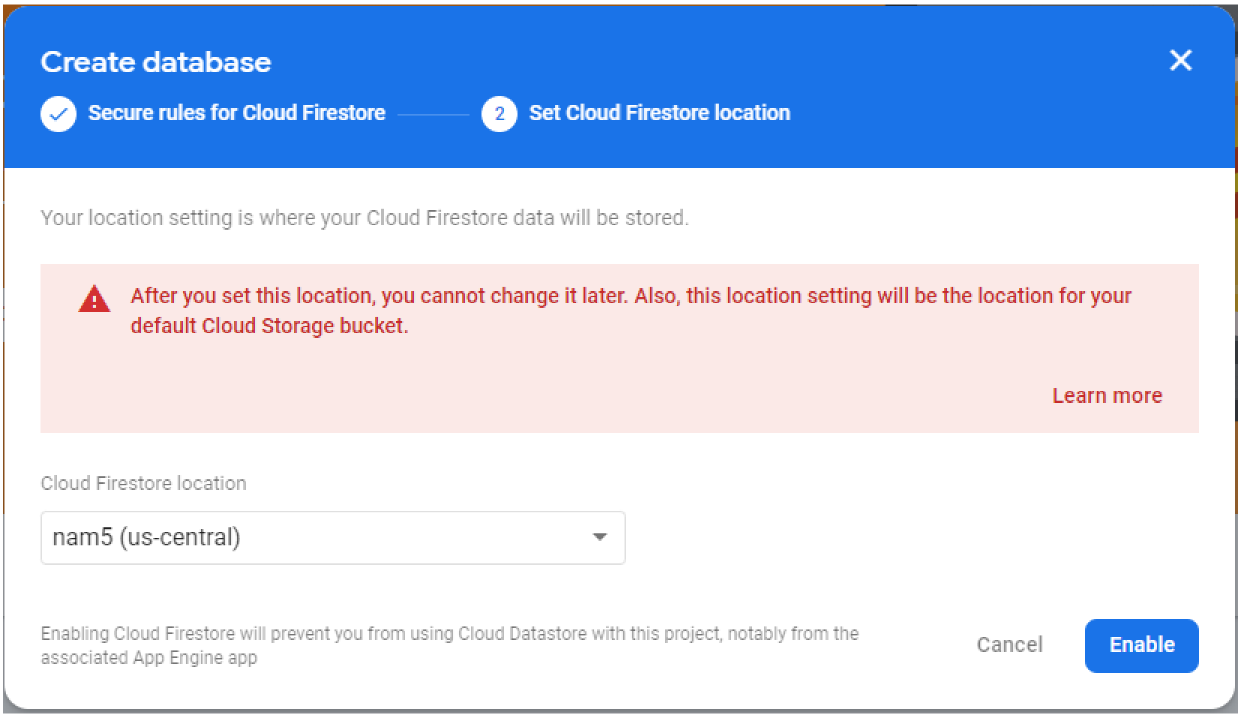
● Start in **test mode**
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
          request.time < timestamp.date(2022, 6, 30);
    }
  }
}
```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

4. PIlih Done.



Create database

✕
✓ Secure rules for Cloud Firestore
2 Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

Learn more

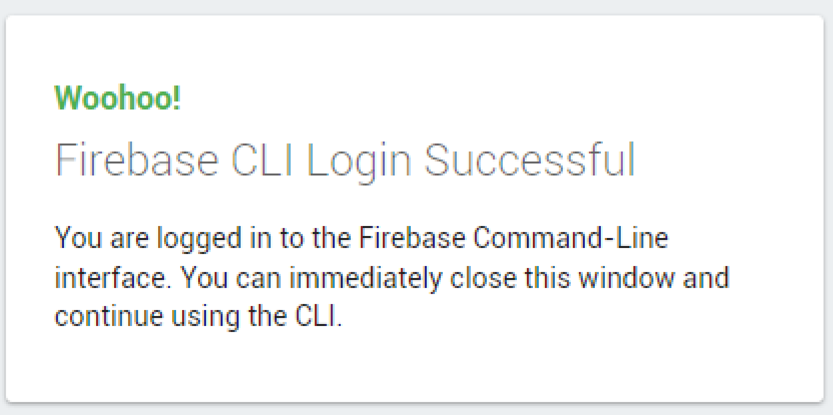
Cloud Firestore location

nam5 (us-central)

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Setting react app dan firebase.

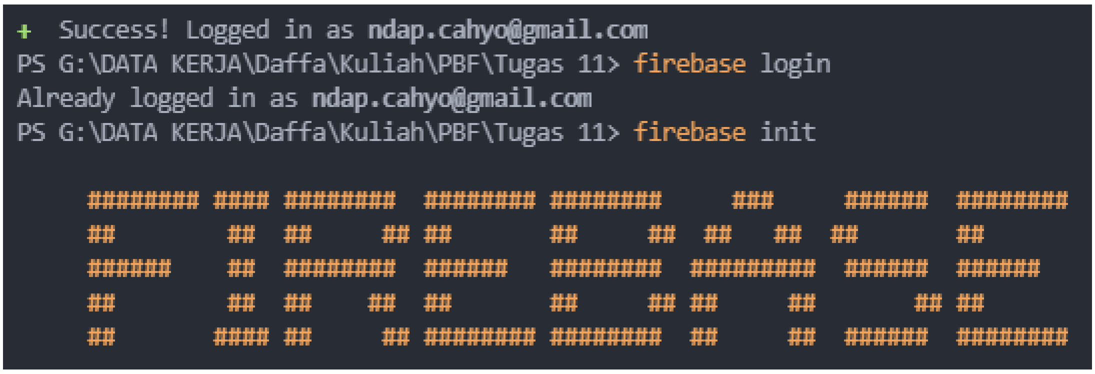
1. Install Firebase dan login.



Woohoo!

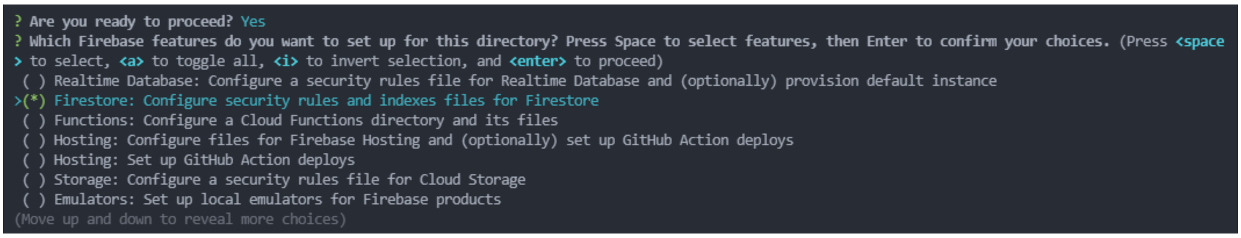
Firebase CLI Login Successful

You are logged in to the Firebase Command-Line interface. You can immediately close this window and continue using the CLI.



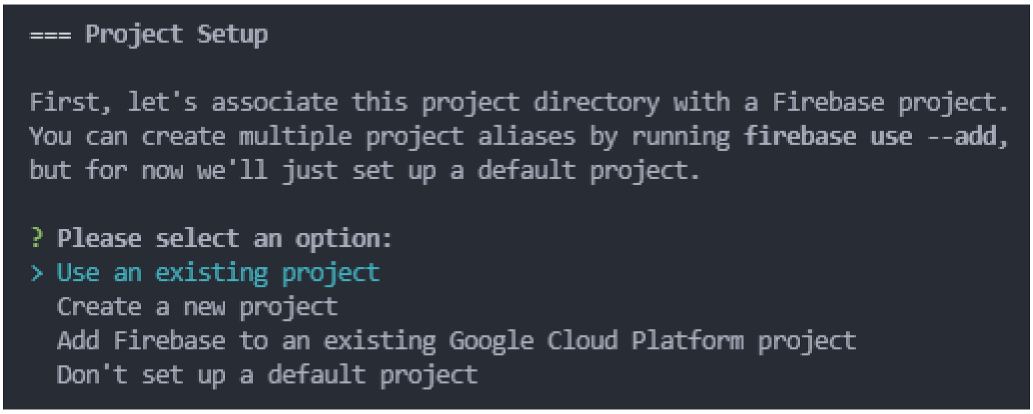
```
+  Success! Logged in as ndap.cahyo@gmail.com
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> firebase login
Already logged in as ndap.cahyo@gmail.com
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> firebase init

     ######## #### ########  ######## ########     ###     ######  ########
     ##        ##  ##     ## ##       ##     ##  ##   ##  ##       ##
     ######    ##  ########  ######   ########  #########  ######  ######
     ##        ##  ##    ##  ##       ##     ## ##     ##       ## ##
     ##       #### ##     ## ######## ########  ##     ##  ######  ########
```

2. Pilih setting untuk firebase nya firebase dan Hosting.

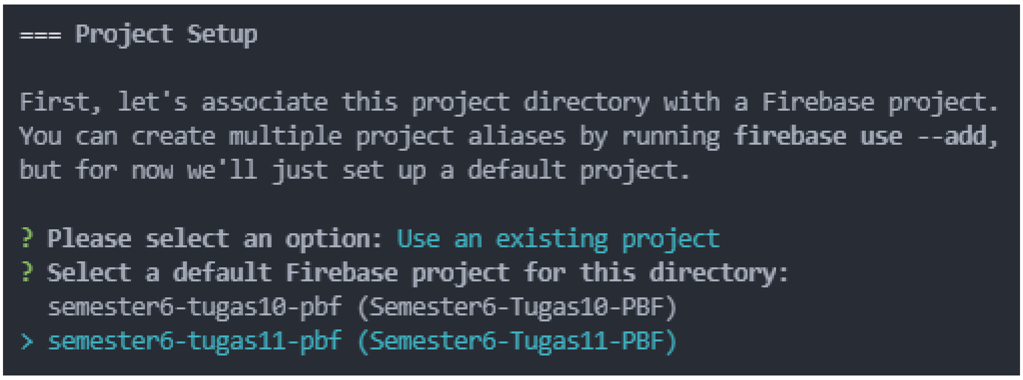


```
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. (Press <space
> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
 ( ) Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance
>(*) Firestore: Configure security rules and indexes files for Firestore
 ( ) Functions: Configure a Cloud Functions directory and its files
 ( ) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
 ( ) Hosting: Set up GitHub Action deploys
 ( ) Storage: Configure a security rules file for Cloud Storage
 ( ) Emulators: Set up local emulators for Firebase products
(Move up and down to reveal more choices)
```

3. Pilih build jangan pilih Do not choose public.

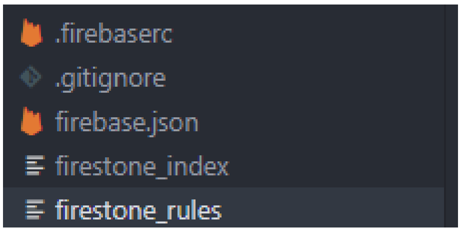


```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option:
> Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project
```



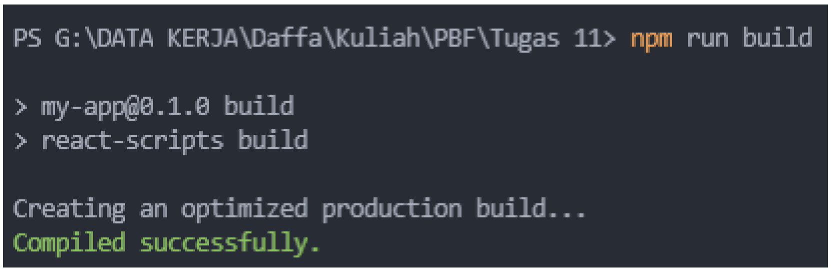
```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory:
  semester6-tugas10-pbf (Semester6-Tugas10-PBF)
> semester6-tugas11-pbf (Semester6-Tugas11-PBF)
```

Buat file tersendiri untuk firestone_index dan firestone_rules.

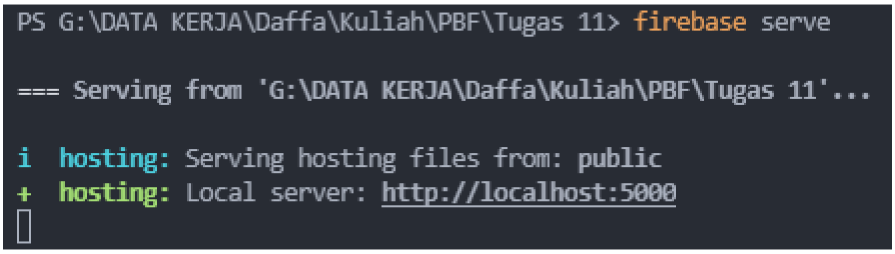


```
.firebaserc
.gitignore
firebase.json
firestone_index
firestone_rules
```

4. Untuk membangun firebase.



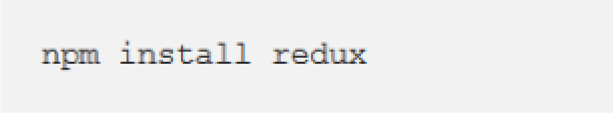
```
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> npm run build

> my-app@0.1.0 build
> react-scripts build

Creating an optimized production build...
Compiled successfully.
```



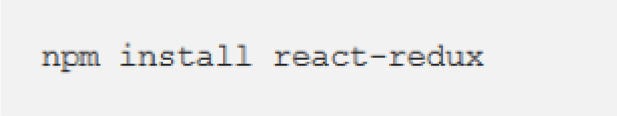
```
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> firebase serve

=== Serving from 'G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11'...

i  hosting: Serving hosting files from: public
+  hosting: Local server: http://localhost:5000
```

Instalasi Redux.

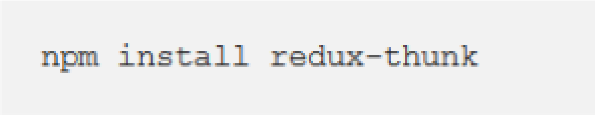
1. Install Redux dan react redux, redux thunk.



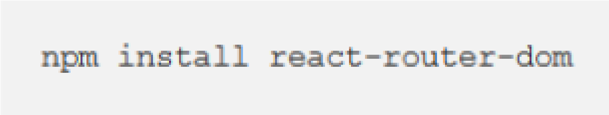
```
npm install redux
```



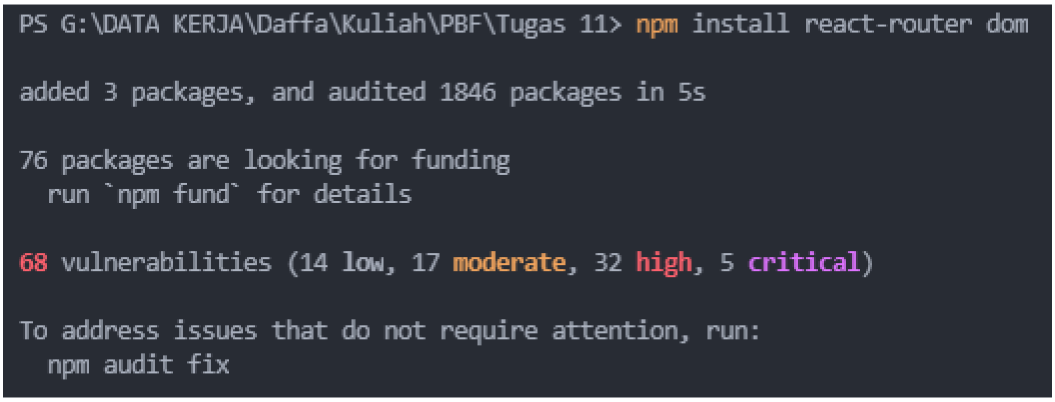
```
npm install react-redux
```



```
npm install redux-thunk
```

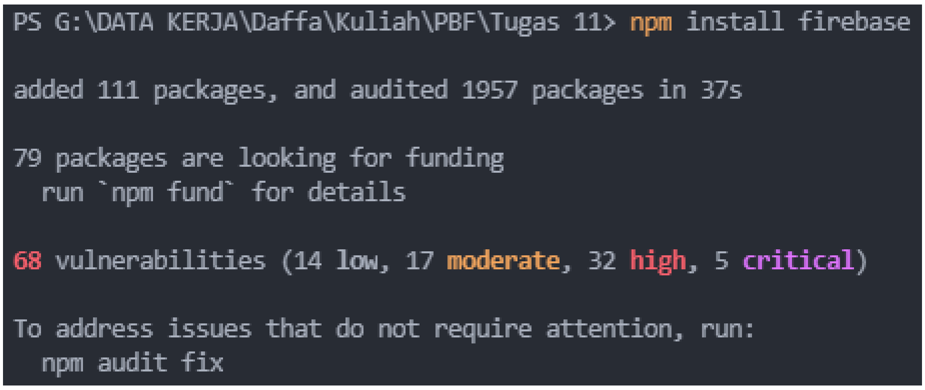


```
npm install react-router-dom
```



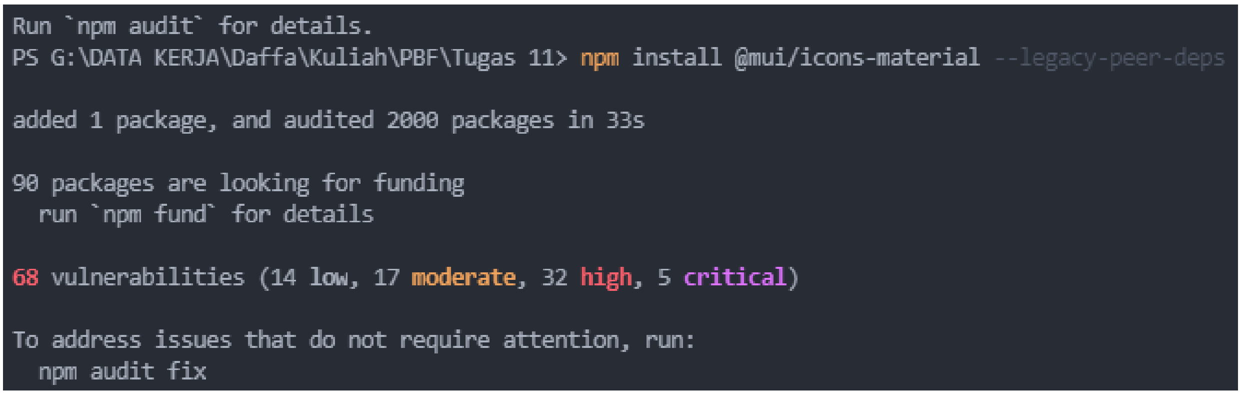
```
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> npm install react-router dom

added 3 packages, and audited 1846 packages in 5s

76 packages are looking for funding
  run `npm fund` for details

68 vulnerabilities (14 low, 17 moderate, 32 high, 5 critical)

To address issues that do not require attention, run:
  npm audit fix
```

2. Install firebase.



```
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> npm install firebase

added 111 packages, and audited 1957 packages in 37s

79 packages are looking for funding
  run `npm fund` for details

68 vulnerabilities (14 low, 17 moderate, 32 high, 5 critical)

To address issues that do not require attention, run:
  npm audit fix
```

3. Pilih UI library.

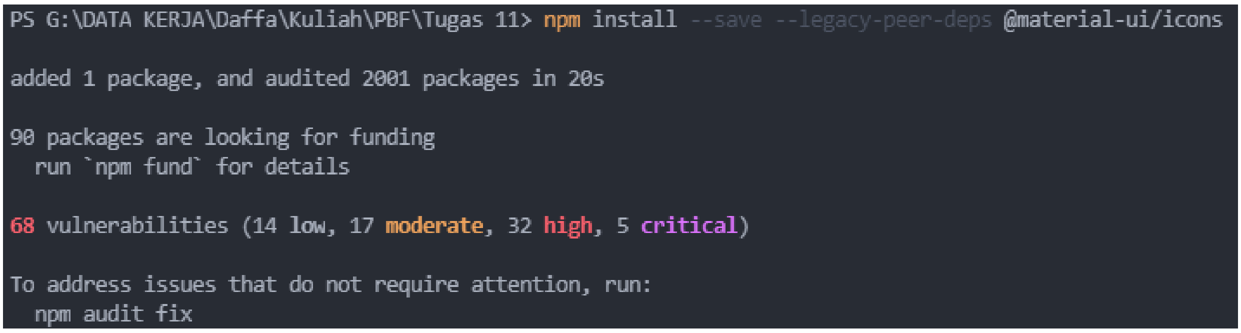


```
Run `npm audit` for details.
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> npm install @mui/icons-material --legacy-peer-deps

added 1 package, and audited 2000 packages in 33s

90 packages are looking for funding
  run `npm fund` for details

68 vulnerabilities (14 low, 17 moderate, 32 high, 5 critical)

To address issues that do not require attention, run:
  npm audit fix
```

4. Paste kan file public/index.html.

```
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swa
p" />
```

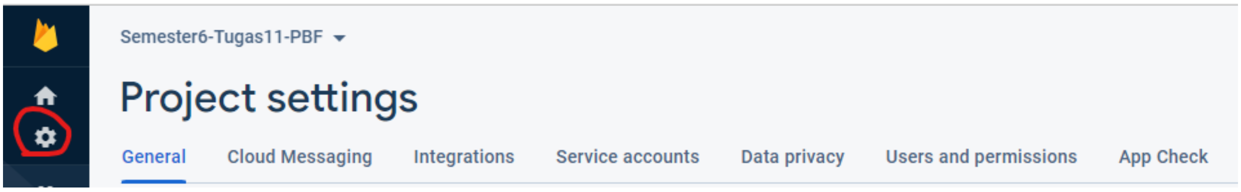
5. Install material icon.



```
PS G:\DATA KERJA\Daffa\Kuliah\PBF\Tugas 11> npm install --save --legacy-peer-deps @material-ui/icons

added 1 package, and audited 2001 packages in 20s

90 packages are looking for funding
  run `npm fund` for details

68 vulnerabilities (14 low, 17 moderate, 32 high, 5 critical)

To address issues that do not require attention, run:
  npm audit fix
```

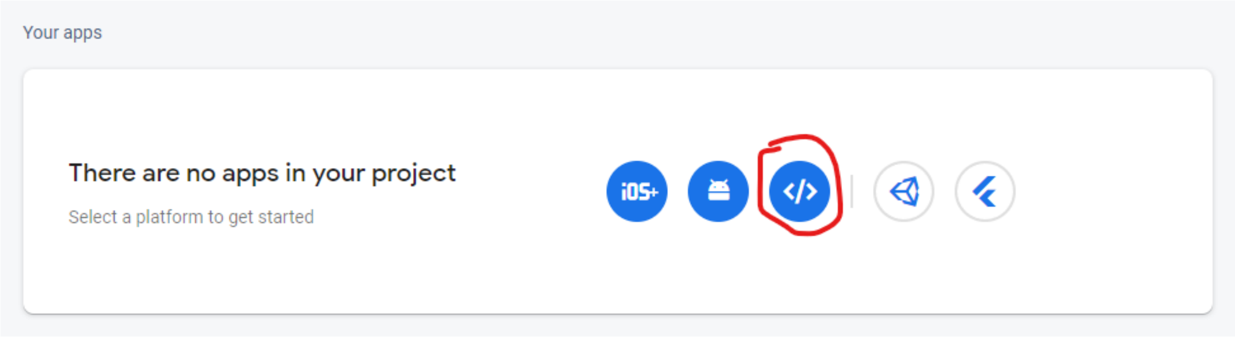
Setting firebase config.

1. Pilih project setting.



Semester6-Tugas11-PBF

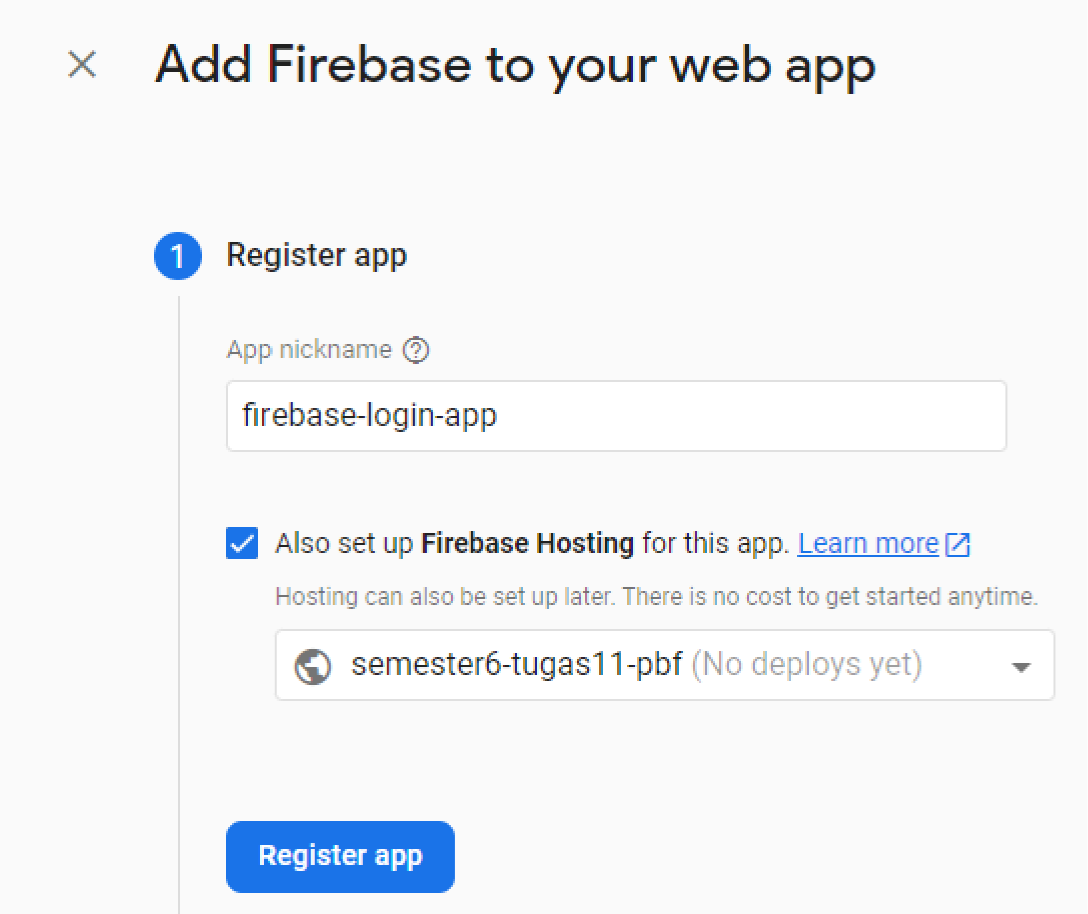
Project settings



Your apps

There are no apps in your project

Select a platform to get started



Add Firebase to your web app

1 Register app

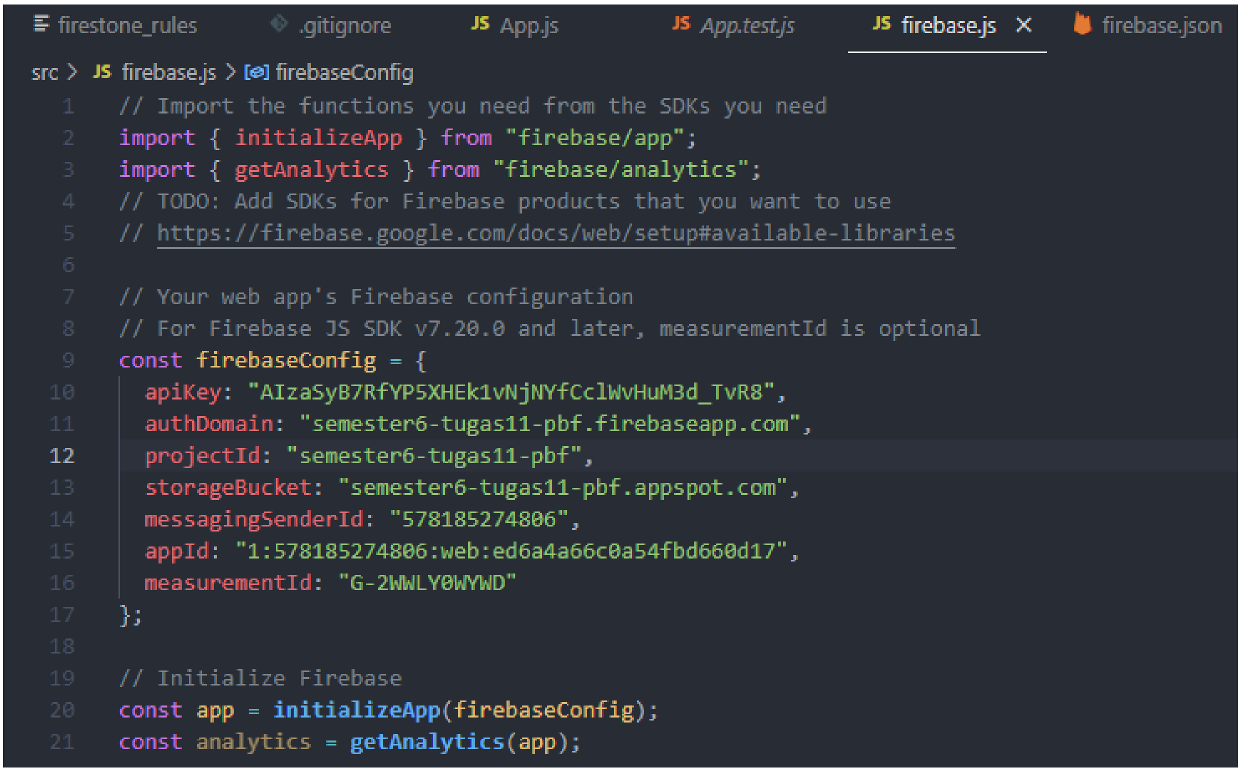
App nickname: firebase-login-app

☑ Also set up **Firebase Hosting** for this app. Learn more

Hosting can also be set up later. There is no cost to get started anytime.

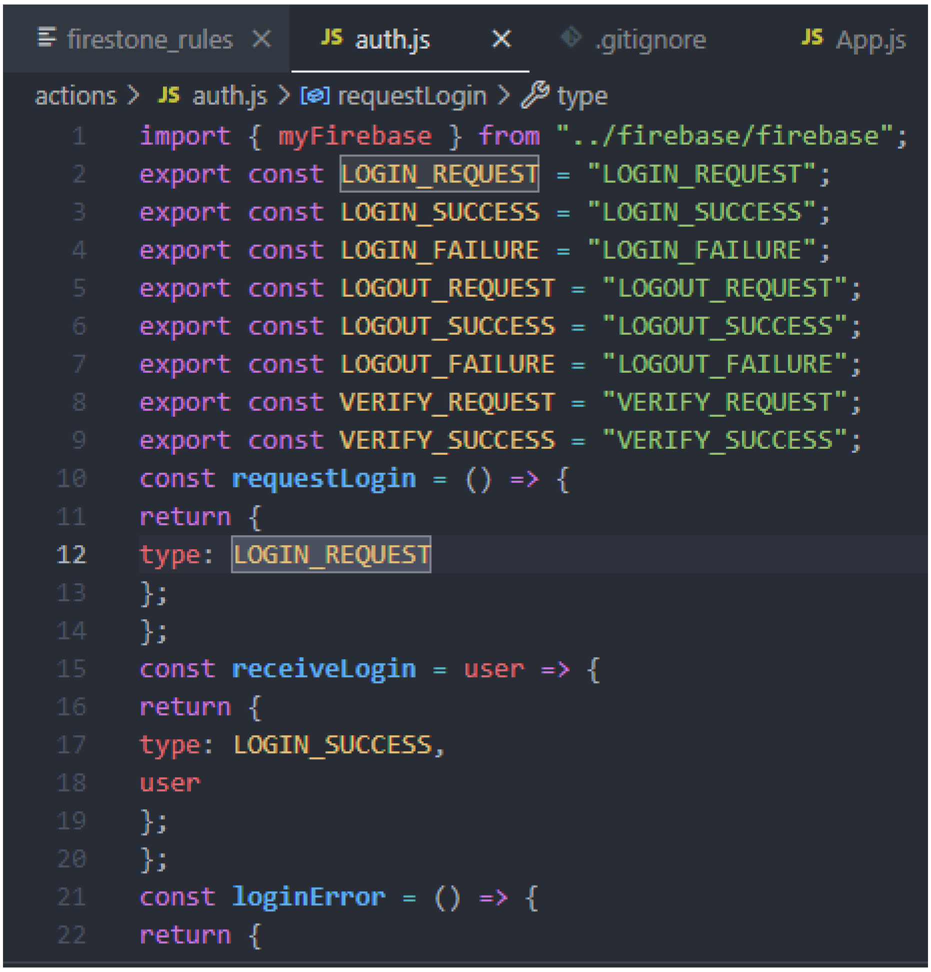
semester6-tugas11-pbf (No deploys yet)

2. Buat file firebase baru dan masukkan kodingan berikut:

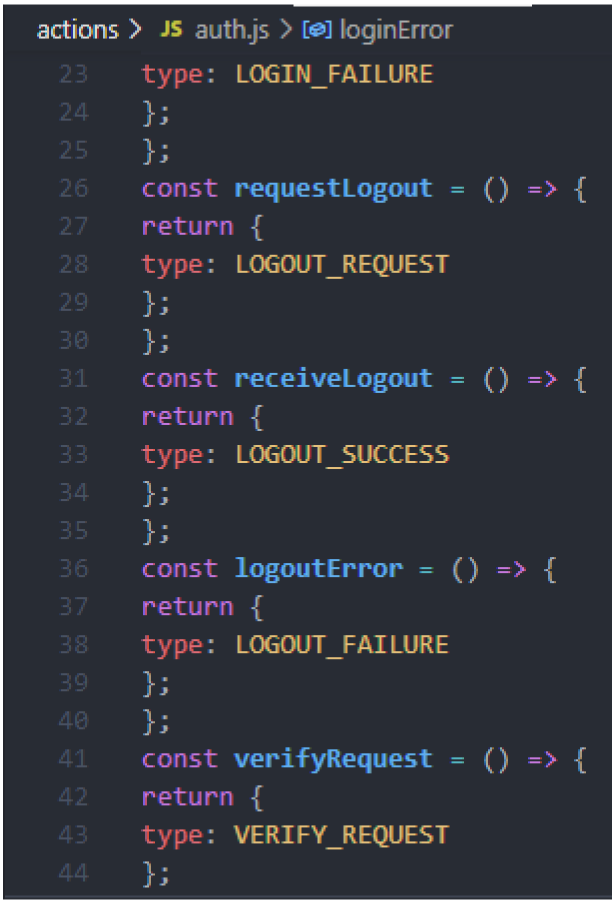


```javascript
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyB7RfYP5XHEk1vNjNYfCclWvHuM3d_TvR8",
  authDomain: "semester6-tugas11-pbf.firebaseapp.com",
  projectId: "semester6-tugas11-pbf",
  storageBucket: "semester6-tugas11-pbf.appspot.com",
  messagingSenderId: "578185274806",
  appId: "1:578185274806:web:ed6a4a66c0a54fbd660d17",
  measurementId: "G-2WWLY0WYWD"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

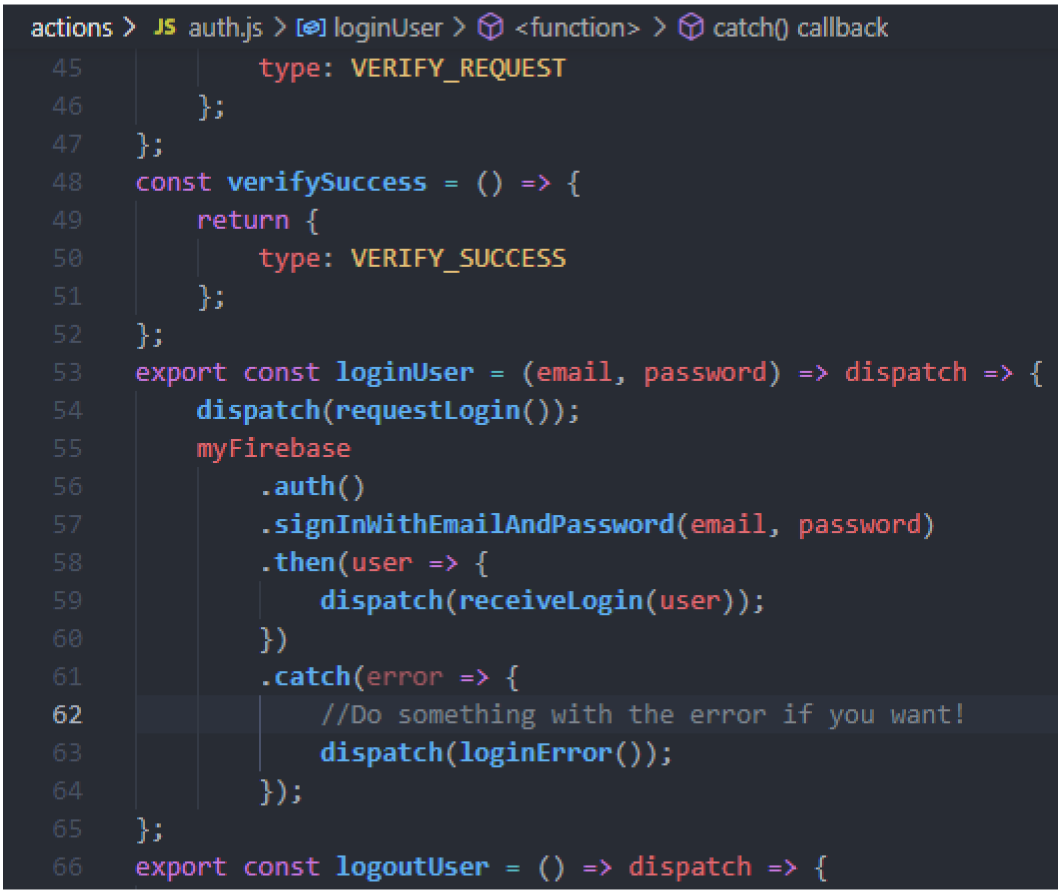
3. Edit actions/auth.js.



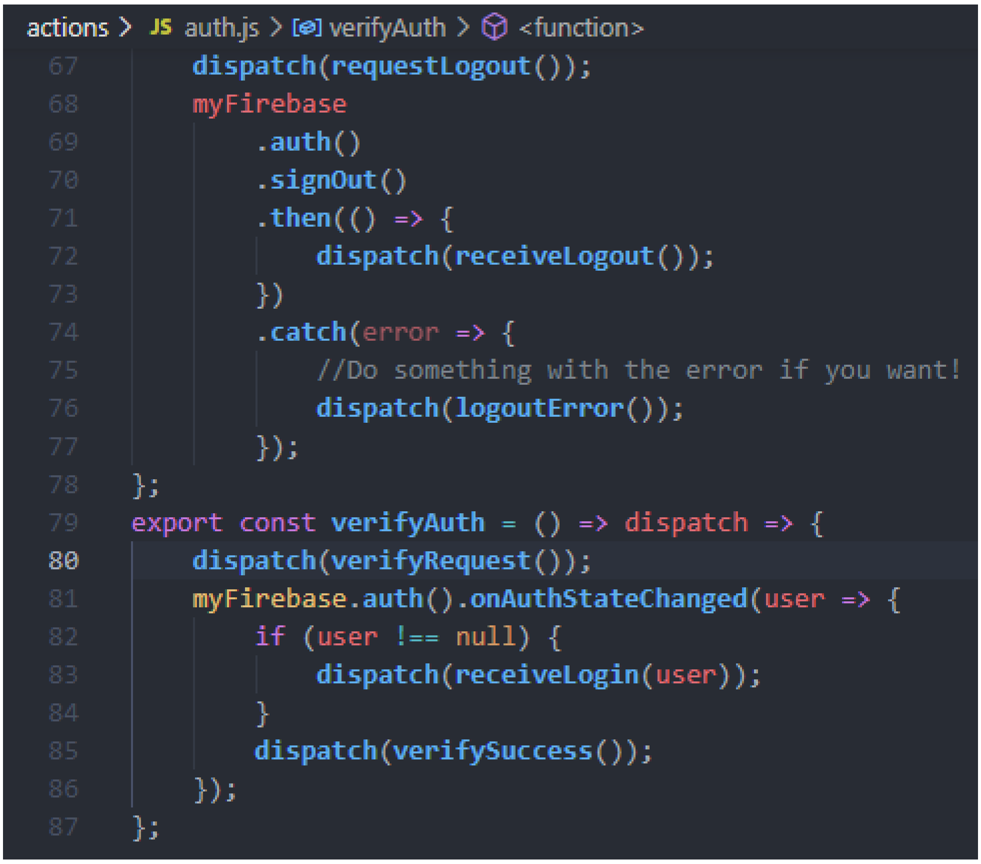
```javascript
import { myFirebase } from "../firebase/firebase";
export const LOGIN_REQUEST = "LOGIN_REQUEST";
export const LOGIN_SUCCESS = "LOGIN_SUCCESS";
export const LOGIN_FAILURE = "LOGIN_FAILURE";
export const LOGOUT_REQUEST = "LOGOUT_REQUEST";
export const LOGOUT_SUCCESS = "LOGOUT_SUCCESS";
export const LOGOUT_FAILURE = "LOGOUT_FAILURE";
export const VERIFY_REQUEST = "VERIFY_REQUEST";
export const VERIFY_SUCCESS = "VERIFY_SUCCESS";
const requestLogin = () => {
return {
type: LOGIN_REQUEST
};
};
const receiveLogin = user => {
return {
type: LOGIN_SUCCESS,
user
};
};
const loginError = () => {
return {
```



```
type: LOGIN_FAILURE
};
};
const requestLogout = () => {
return {
type: LOGOUT_REQUEST
};
};
const receiveLogout = () => {
return {
type: LOGOUT_SUCCESS
};
};
const logoutError = () => {
return {
type: LOGOUT_FAILURE
};
};
const verifyRequest = () => {
return {
type: VERIFY_REQUEST
};
```


actions > JS auth.js > loginUser > <function> > catch() callback

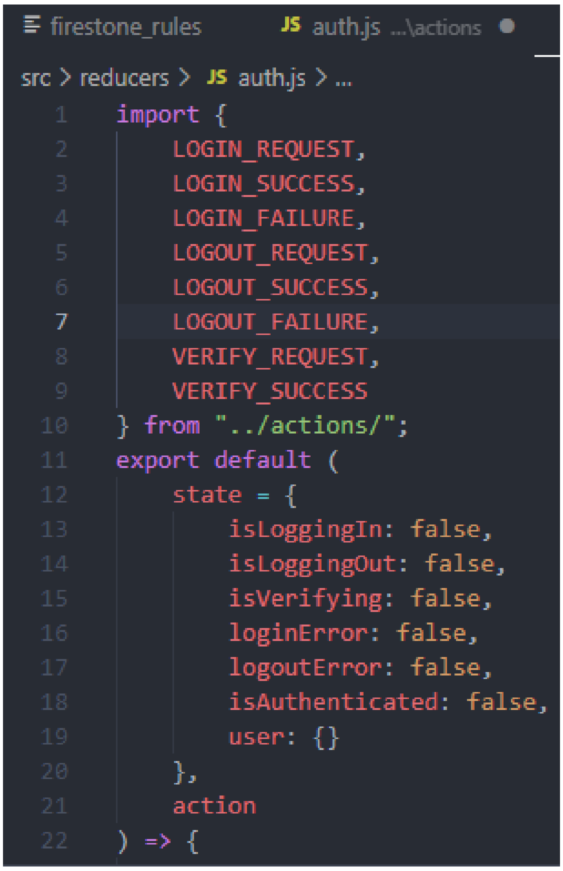
```
        type: VERIFY_REQUEST
    };
};
const verifySuccess = () => {
    return {
        type: VERIFY_SUCCESS
    };
};
export const loginUser = (email, password) => dispatch => {
    dispatch(requestLogin());
    myFirebase
        .auth()
        .signInWithEmailAndPassword(email, password)
        .then(user => {
            dispatch(receiveLogin(user));
        })
        .catch(error => {
            //Do something with the error if you want!
            dispatch(loginError());
        });
};
export const logoutUser = () => dispatch => {
```



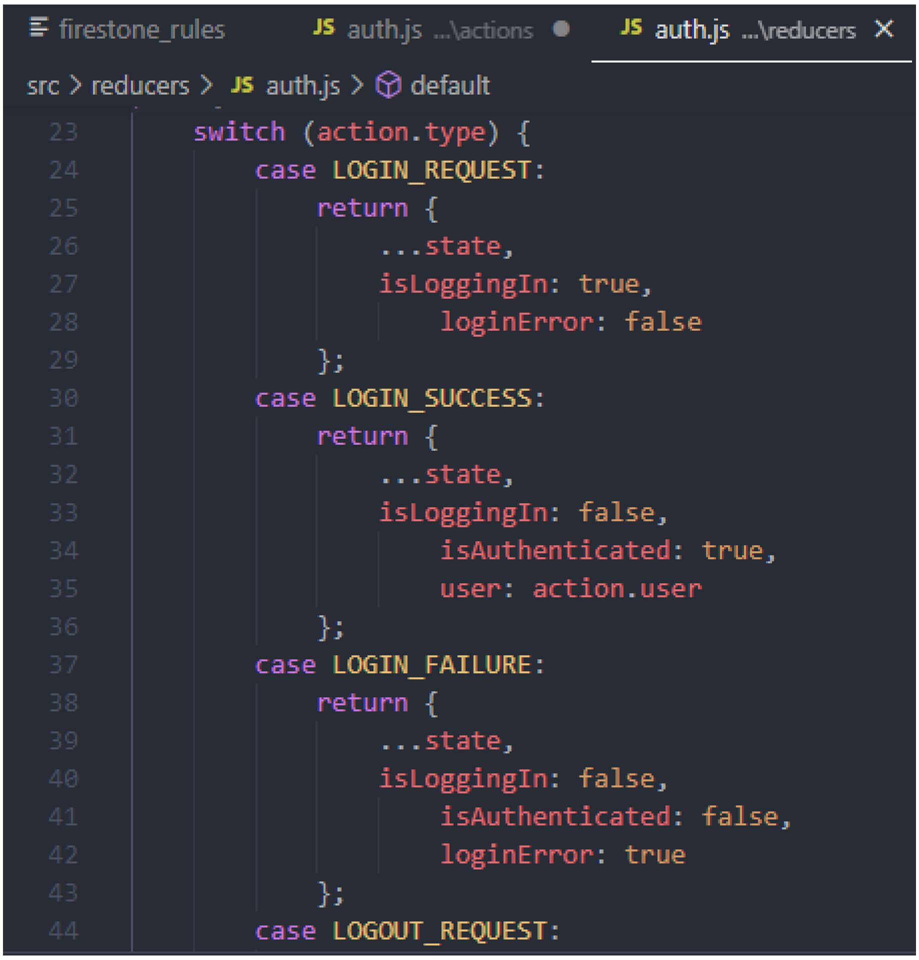
```
    dispatch(requestLogout());
    myFirebase
        .auth()
        .signOut()
        .then(() => {
            dispatch(receiveLogout());
        })
        .catch(error => {
            //Do something with the error if you want!
            dispatch(logoutError());
        });
};
export const verifyAuth = () => dispatch => {
    dispatch(verifyRequest());
    myFirebase.auth().onAuthStateChanged(user => {
        if (user !== null) {
            dispatch(receiveLogin(user));
        }
        dispatch(verifySuccess());
    });
};
```

Setting reducer.

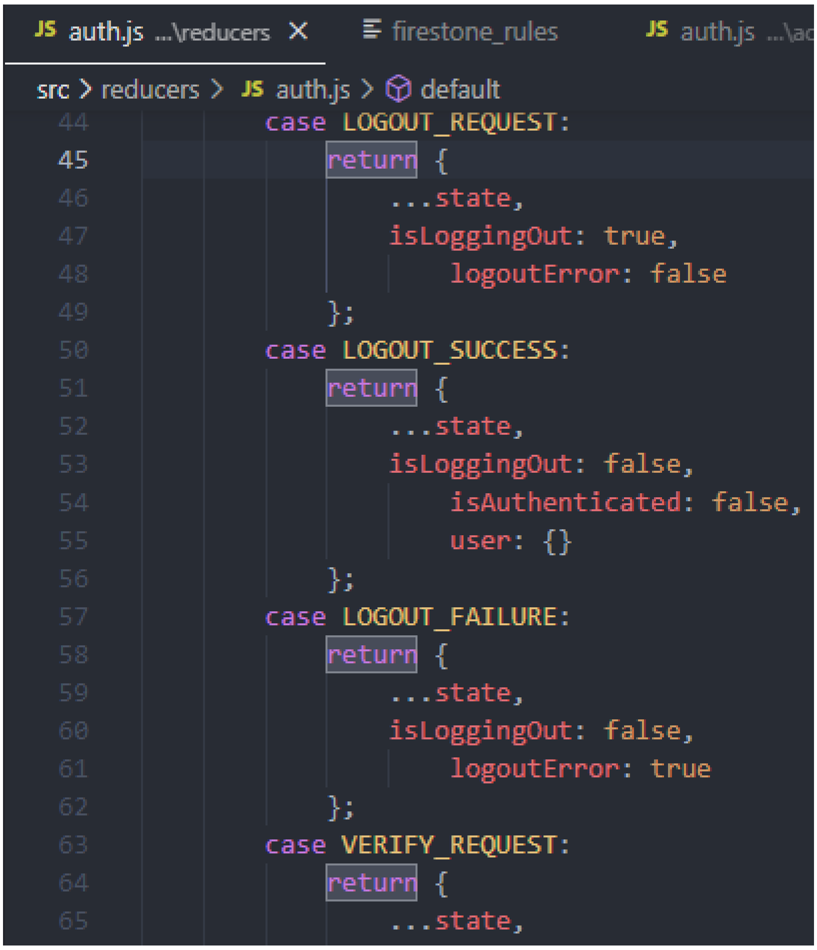
1. Buka reducers/auth.js



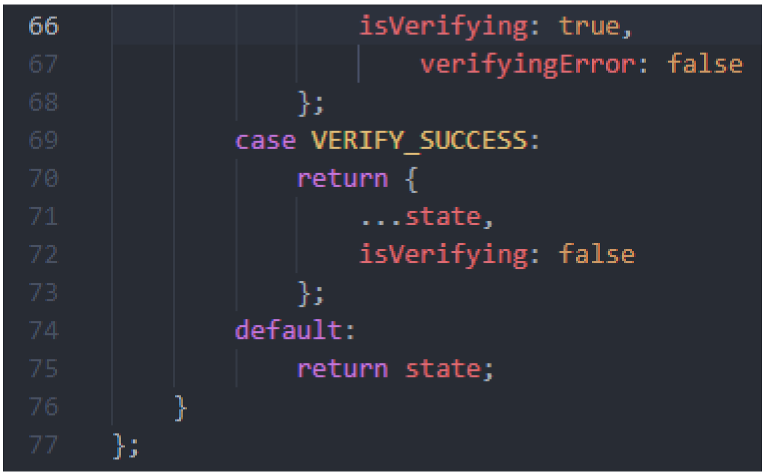
```
import {
    LOGIN_REQUEST,
    LOGIN_SUCCESS,
    LOGIN_FAILURE,
    LOGOUT_REQUEST,
    LOGOUT_SUCCESS,
    LOGOUT_FAILURE,
    VERIFY_REQUEST,
    VERIFY_SUCCESS
} from "../actions/";
export default (
    state = {
        isLoggingIn: false,
        isLoggingOut: false,
        isVerifying: false,
        loginError: false,
        logoutError: false,
        isAuthenticated: false,
        user: {}
    },
    action
) => {
```



```
    switch (action.type) {
        case LOGIN_REQUEST:
            return {
                ...state,
                isLoggingIn: true,
                    loginError: false
            };
        case LOGIN_SUCCESS:
            return {
                ...state,
                isLoggingIn: false,
                    isAuthenticated: true,
                    user: action.user
            };
        case LOGIN_FAILURE:
            return {
                ...state,
                isLoggingIn: false,
                    isAuthenticated: false,
                    loginError: true
            };
        case LOGOUT_REQUEST:
```

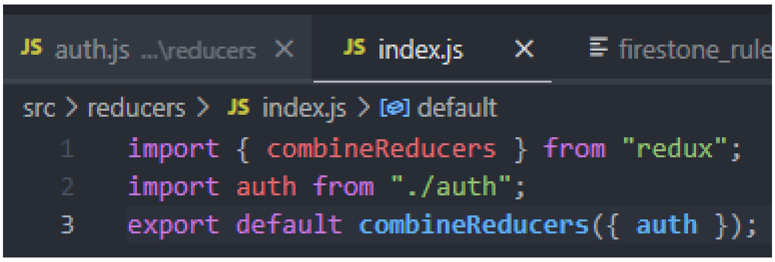


```
    case LOGOUT_REQUEST:
      return {
        ...state,
        isLoggingOut: true,
          logoutError: false
      };
    case LOGOUT_SUCCESS:
      return {
        ...state,
        isLoggingOut: false,
          isAuthenticated: false,
          user: {}
      };
    case LOGOUT_FAILURE:
      return {
        ...state,
        isLoggingOut: false,
          logoutError: true
      };
    case VERIFY_REQUEST:
      return {
        ...state,
```



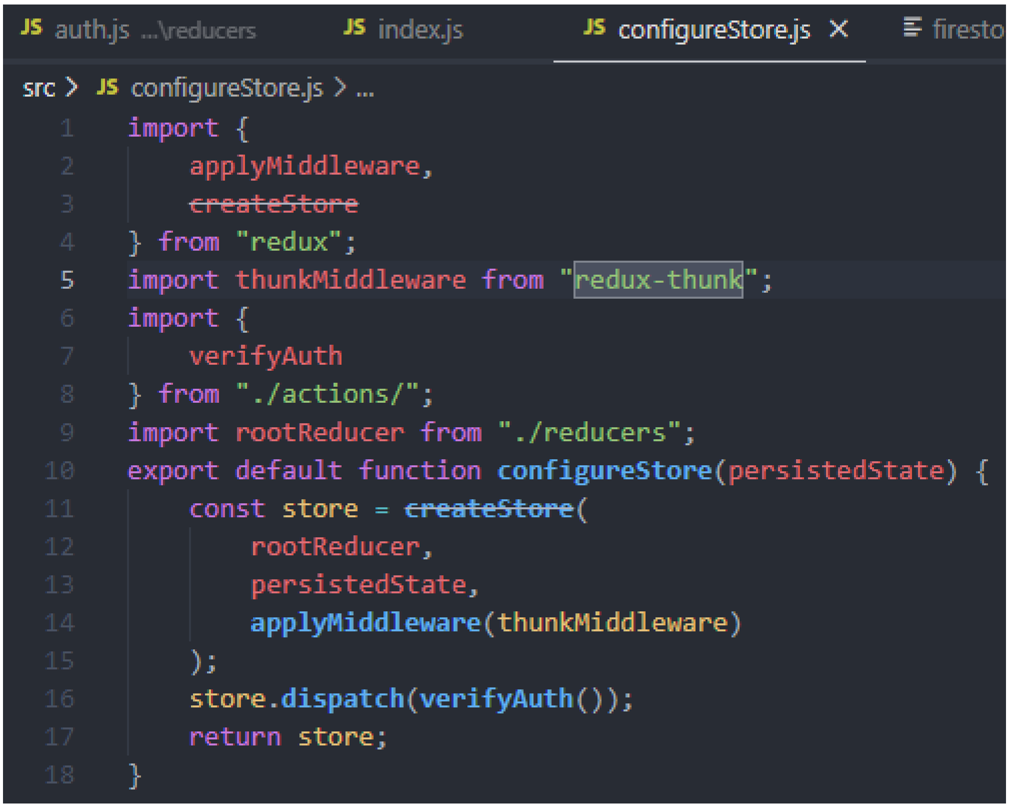
```
        isVerifying: true,
          verifyingError: false
      };
    case VERIFY_SUCCESS:
      return {
        ...state,
        isVerifying: false
      };
    default:
      return state;
  }
};
```

2. Edit reducers/index.js



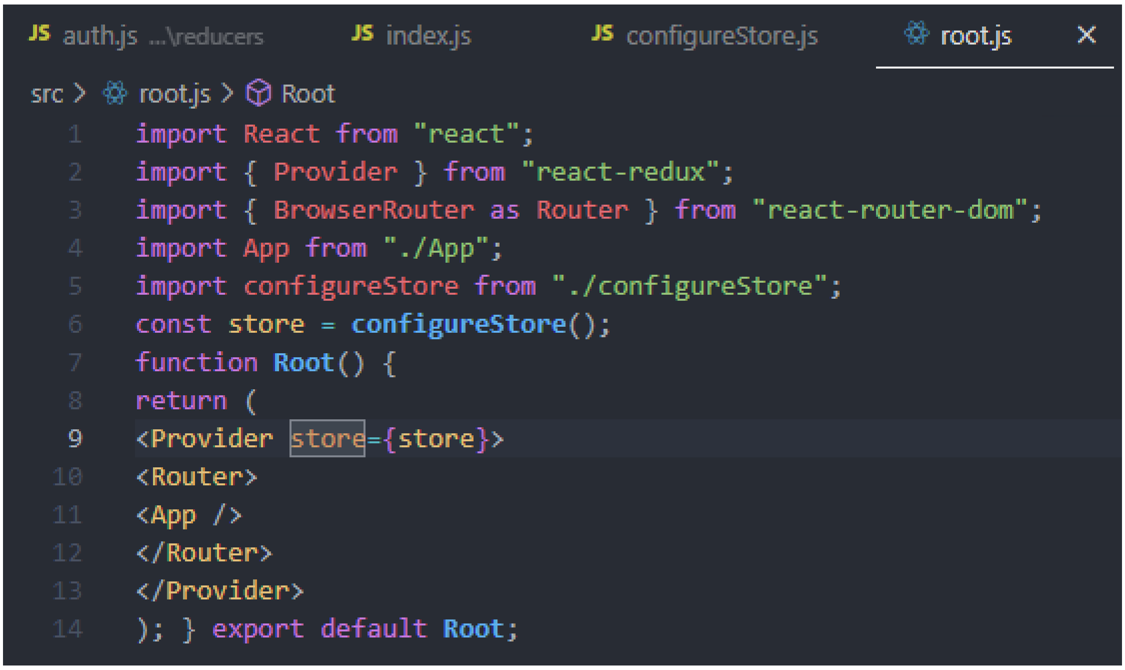
```
import { combineReducers } from "redux";
import auth from "./auth";
export default combineReducers({ auth });
```

3. Setting Redux plumbing.
4. Buat file baru di src bernama configureStore.js dan edit-lah:



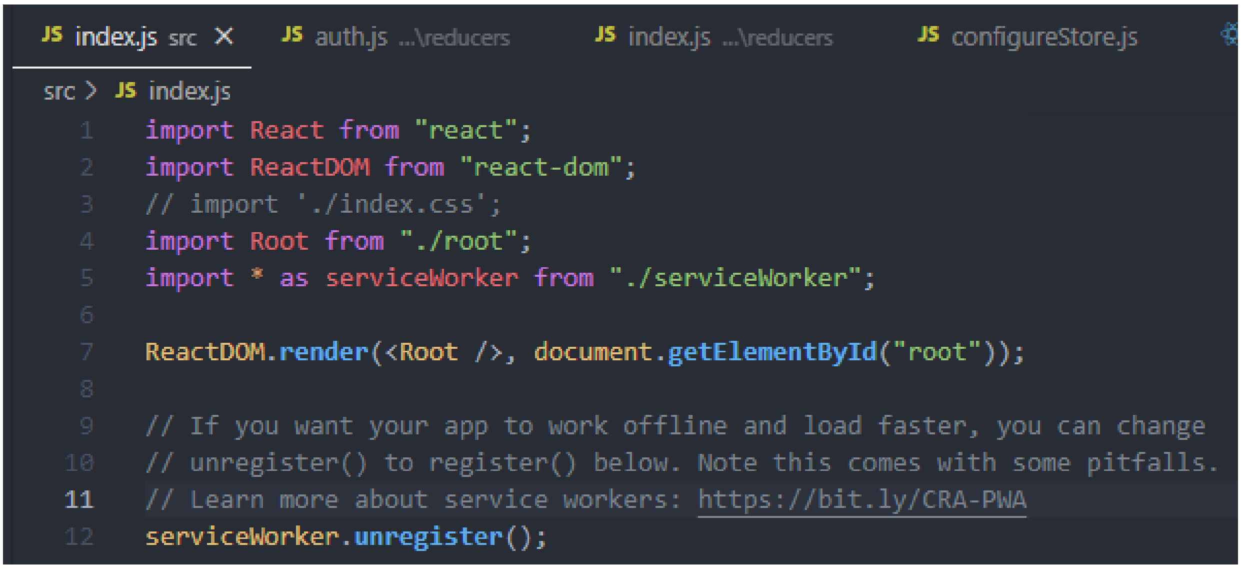
```
import {
    applyMiddleware,
    ~~createStore~~
} from "redux";
import thunkMiddleware from "redux-thunk";
import {
    verifyAuth
} from "./actions/";
import rootReducer from "./reducers";
export default function configureStore(persistedState) {
    const store = ~~createStore~~(
        rootReducer,
        persistedState,
        applyMiddleware(thunkMiddleware)
    );
    store.dispatch(verifyAuth());
    return store;
}
```

5. Buat lagi file baru bernama root.js lalu edit:



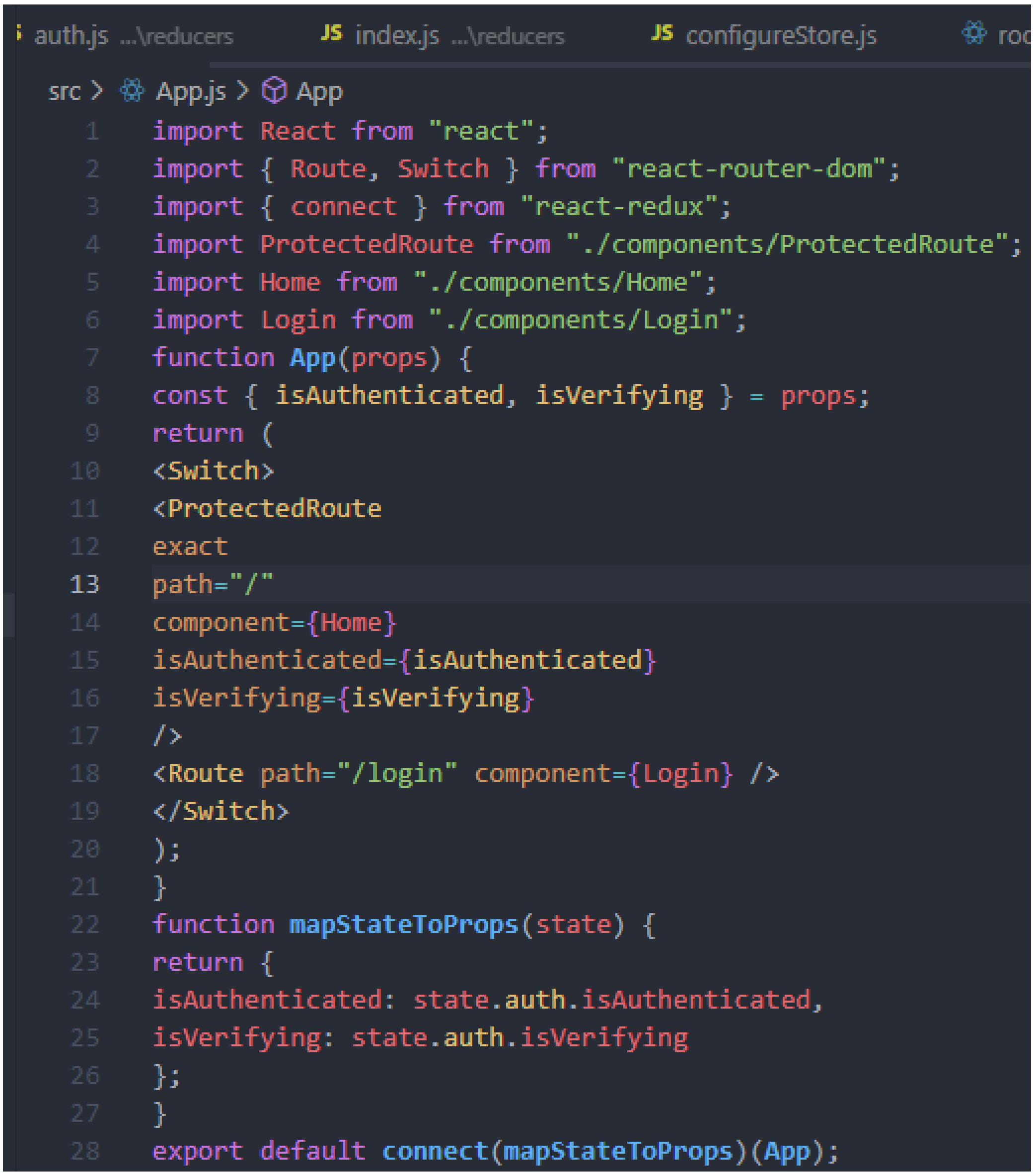
```
import React from "react";
import { Provider } from "react-redux";
import { BrowserRouter as Router } from "react-router-dom";
import App from "./App";
import configureStore from "./configureStore";
const store = configureStore();
function Root() {
return (
<Provider store={store}>
<Router>
<App />
</Router>
</Provider>
); } export default Root;
```

6. Pilih dan edit folder src/index.js.



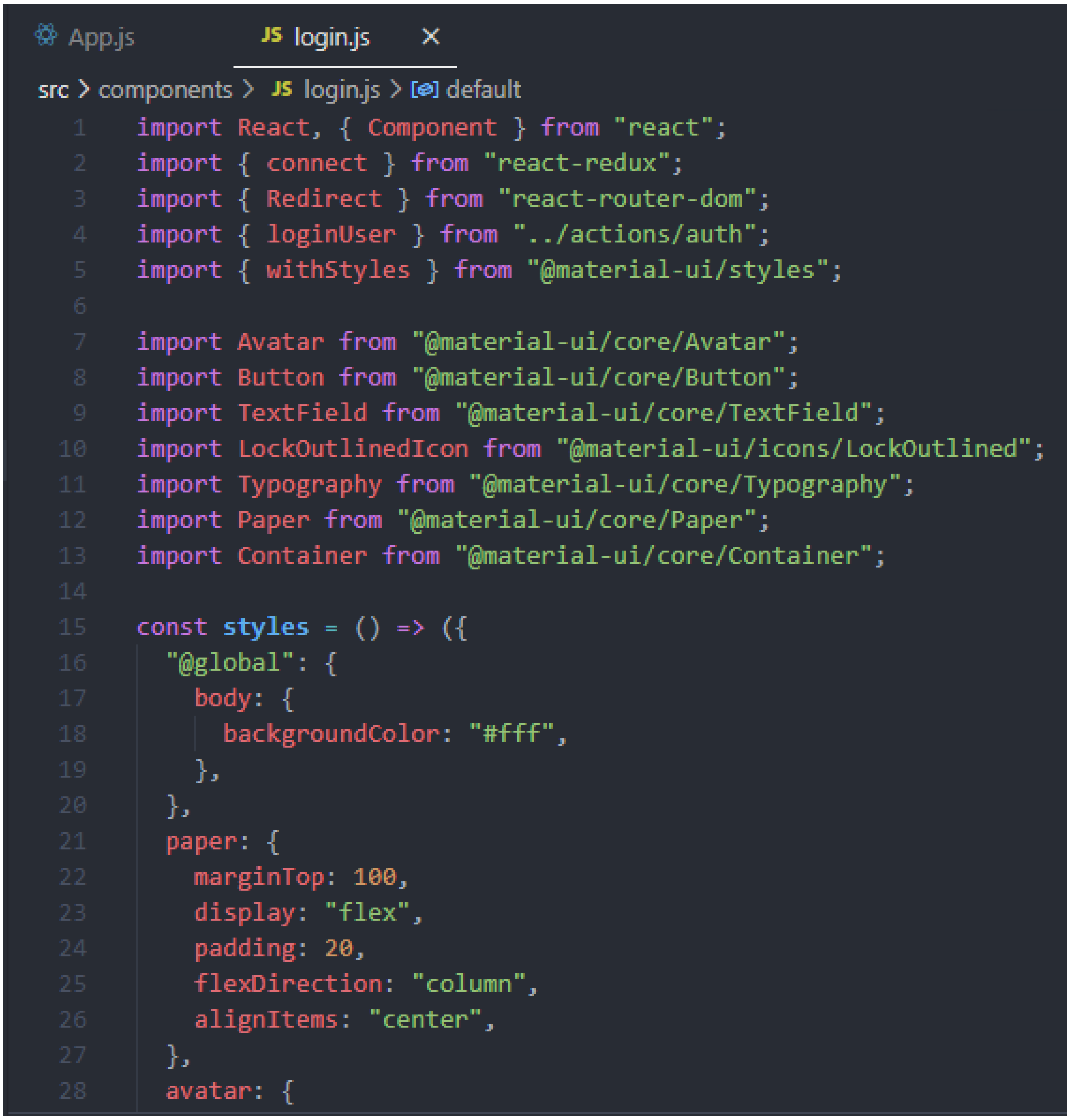
```
import React from "react";
import ReactDOM from "react-dom";
// import './index.css';
import Root from "./root";
import * as serviceWorker from "./serviceWorker";

ReactDOM.render(<Root />, document.getElementById("root"));

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

7. Masukkan coding ini ke App.js.

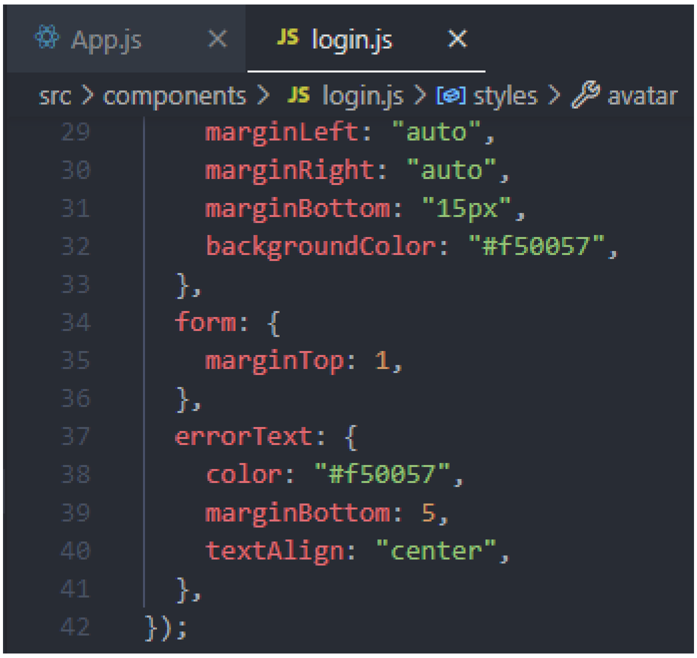


```
import React from "react";
import { Route, Switch } from "react-router-dom";
import { connect } from "react-redux";
import ProtectedRoute from "./components/ProtectedRoute";
import Home from "./components/Home";
import Login from "./components/Login";
function App(props) {
const { isAuthenticated, isVerifying } = props;
return (
<Switch>
<ProtectedRoute
exact
path="/"
component={Home}
isAuthenticated={isAuthenticated}
isVerifying={isVerifying}
/>
<Route path="/login" component={Login} />
</Switch>
);
}
function mapStateToProps(state) {
return {
isAuthenticated: state.auth.isAuthenticated,
isVerifying: state.auth.isVerifying
};
}
export default connect(mapStateToProps)(App);
```

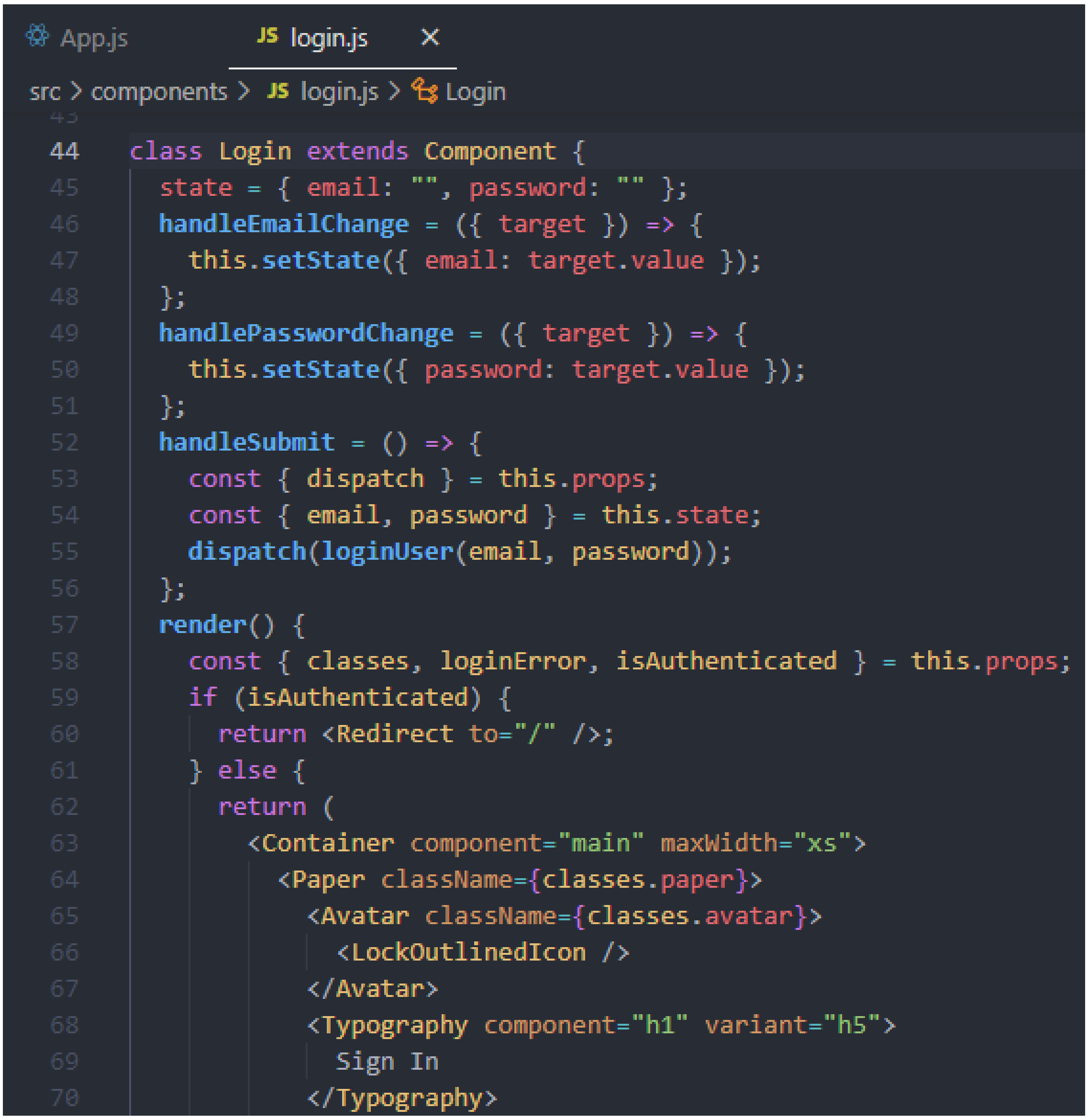
8. Buatlah file dan simpan components/Login.js



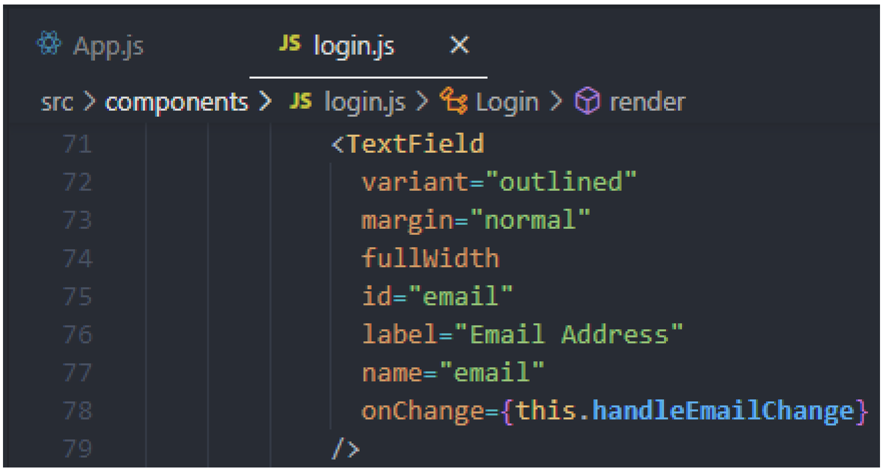
```
import React, { Component } from "react";
import { connect } from "react-redux";
import { Redirect } from "react-router-dom";
import { loginUser } from "../actions/auth";
import { withStyles } from "@material-ui/styles";

import Avatar from "@material-ui/core/Avatar";
import Button from "@material-ui/core/Button";
import TextField from "@material-ui/core/TextField";
import LockOutlinedIcon from "@material-ui/icons/LockOutlined";
import Typography from "@material-ui/core/Typography";
import Paper from "@material-ui/core/Paper";
import Container from "@material-ui/core/Container";

const styles = () => ({
  "@global": {
    body: {
      backgroundColor: "#fff",
    },
  },
  paper: {
    marginTop: 100,
    display: "flex",
    padding: 20,
    flexDirection: "column",
    alignItems: "center",
  },
  avatar: {
```



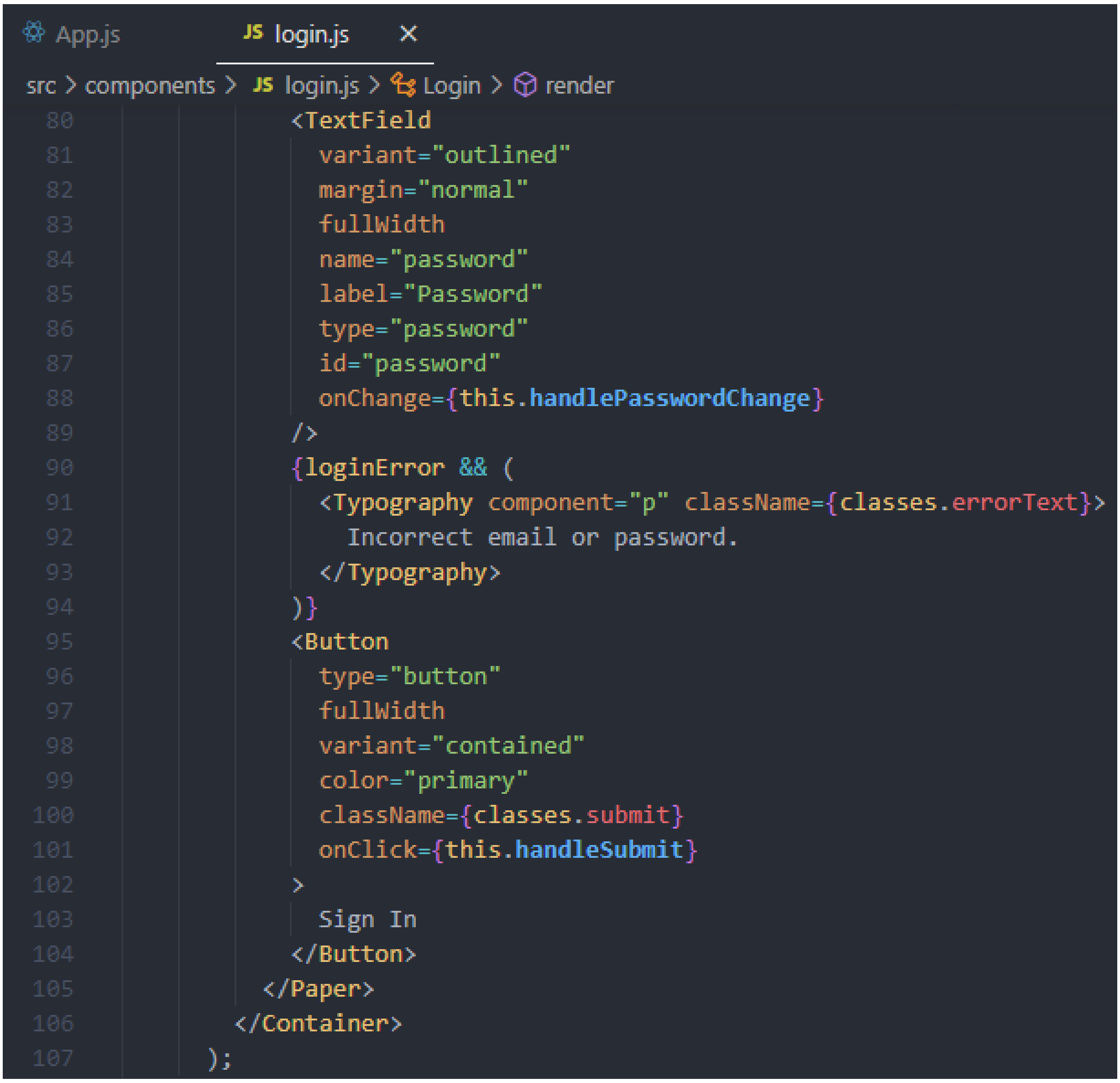
```
    marginLeft: "auto",
    marginRight: "auto",
    marginBottom: "15px",
    backgroundColor: "#f50057",
  },
  form: {
    marginTop: 1,
  },
  errorText: {
    color: "#f50057",
    marginBottom: 5,
    textAlign: "center",
  },
});
```



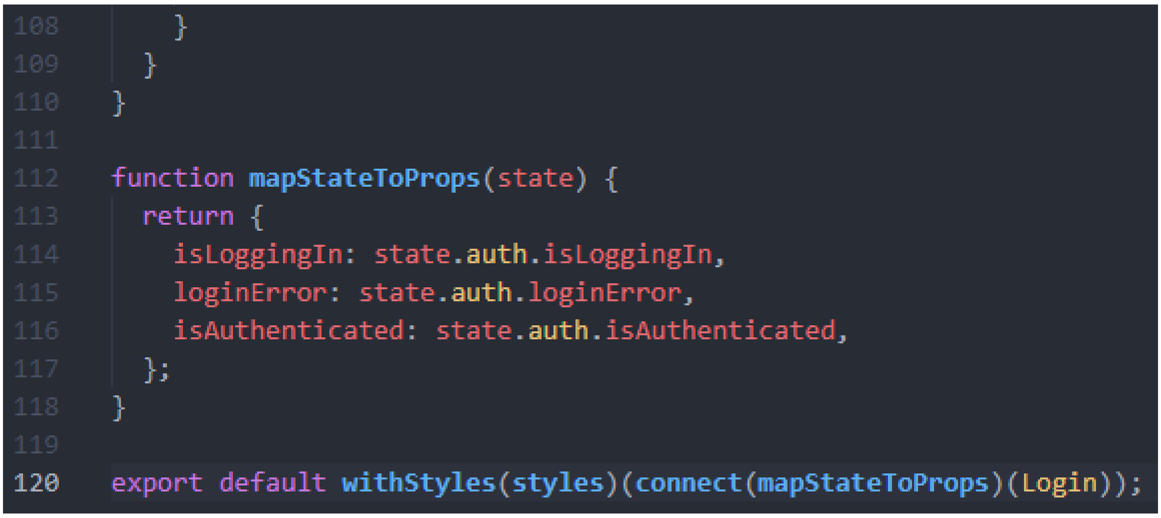
```
class Login extends Component {
  state = { email: "", password: "" };
  handleEmailChange = ({ target }) => {
    this.setState({ email: target.value });
  };
  handlePasswordChange = ({ target }) => {
    this.setState({ password: target.value });
  };
  handleSubmit = () => {
    const { dispatch } = this.props;
    const { email, password } = this.state;
    dispatch(loginUser(email, password));
  };
  render() {
    const { classes, loginError, isAuthenticated } = this.props;
    if (isAuthenticated) {
      return <Redirect to="/" />;
    } else {
      return (
        <Container component="main" maxWidth="xs">
          <Paper className={classes.paper}>
            <Avatar className={classes.avatar}>
              <LockOutlinedIcon />
            </Avatar>
            <Typography component="h1" variant="h5">
              Sign In
            </Typography>
```



```
            <TextField
              variant="outlined"
              margin="normal"
              fullWidth
              id="email"
              label="Email Address"
              name="email"
              onChange={this.handleEmailChange}
            />
```

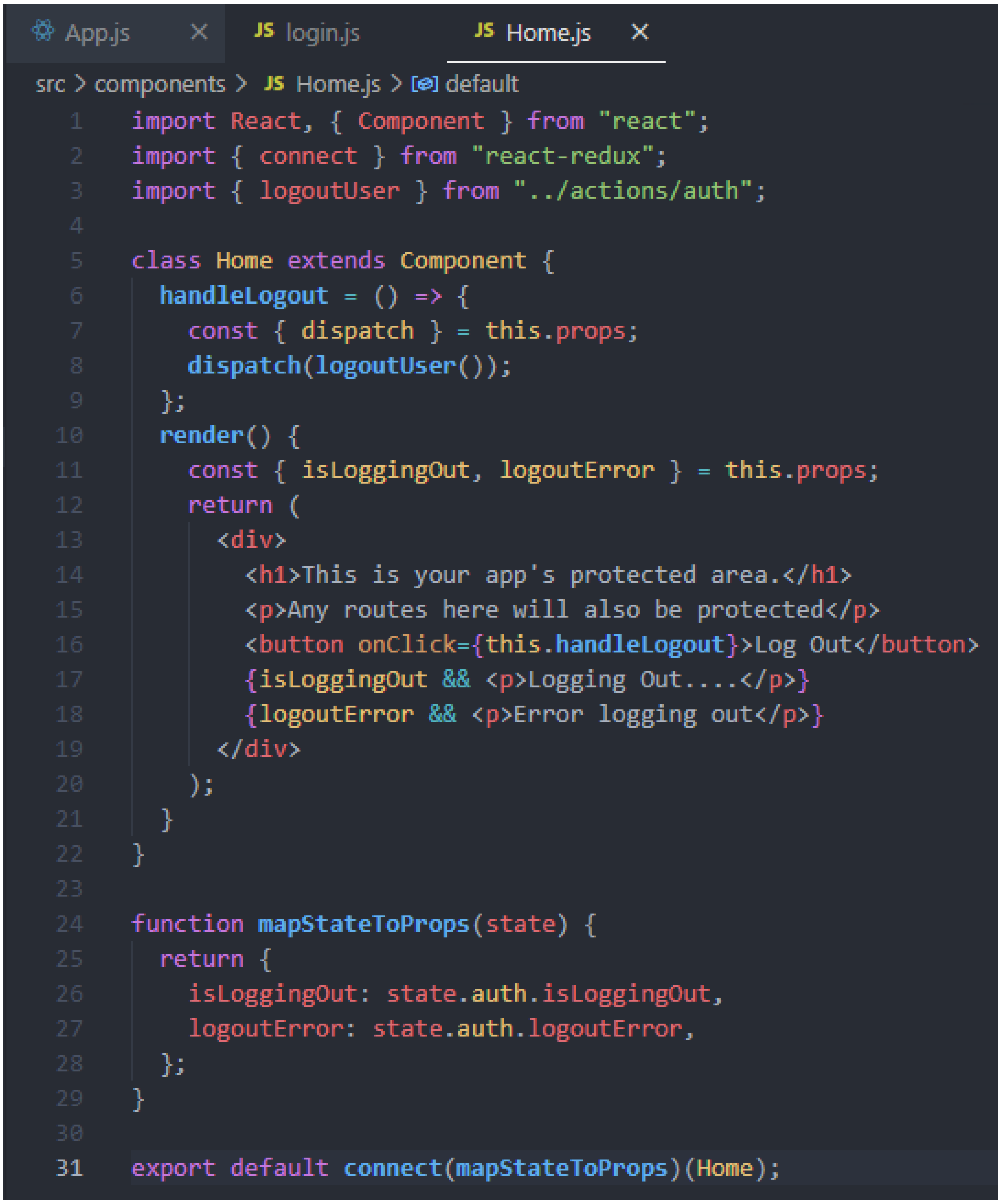


```
            <TextField
              variant="outlined"
              margin="normal"
              fullWidth
              name="password"
              label="Password"
              type="password"
              id="password"
              onChange={this.handlePasswordChange}
            />
            {loginError && (
              <Typography component="p" className={classes.errorText}>
                Incorrect email or password.
              </Typography>
            )}
            <Button
              type="button"
              fullWidth
              variant="contained"
              color="primary"
              className={classes.submit}
              onClick={this.handleSubmit}
            >
              Sign In
            </Button>
          </Paper>
        </Container>
      );
```



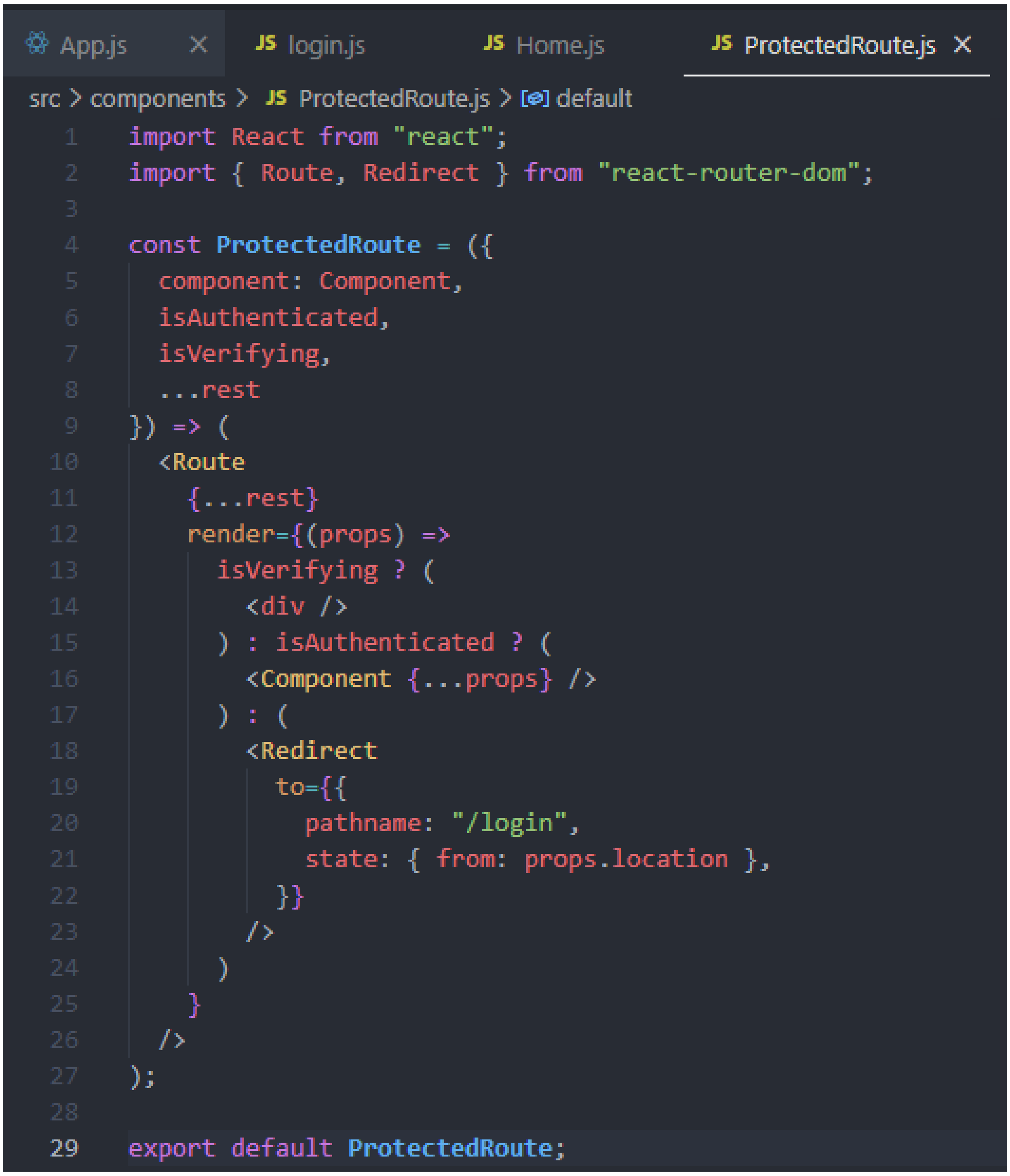
```
    }
  }
}

function mapStateToProps(state) {
  return {
    isLoggingIn: state.auth.isLoggingIn,
    loginError: state.auth.loginError,
    isAuthenticated: state.auth.isAuthenticated,
  };
}

export default withStyles(styles)(connect(mapStateToProps)(Login));
```

9. Sekarang buat file Home.js.



```
import React, { Component } from "react";
import { connect } from "react-redux";
import { logoutUser } from "../actions/auth";

class Home extends Component {
  handleLogout = () => {
    const { dispatch } = this.props;
    dispatch(logoutUser());
  };
  render() {
    const { isLoggingOut, logoutError } = this.props;
    return (
      <div>
        <h1>This is your app's protected area.</h1>
        <p>Any routes here will also be protected</p>
        <button onClick={this.handleLogout}>Log Out</button>
        {isLoggingOut && <p>Logging Out....</p>}
        {logoutError && <p>Error logging out</p>}
      </div>
    );
  }
}

function mapStateToProps(state) {
  return {
    isLoggingOut: state.auth.isLoggingOut,
    logoutError: state.auth.logoutError,
  };
}

export default connect(mapStateToProps)(Home);
```

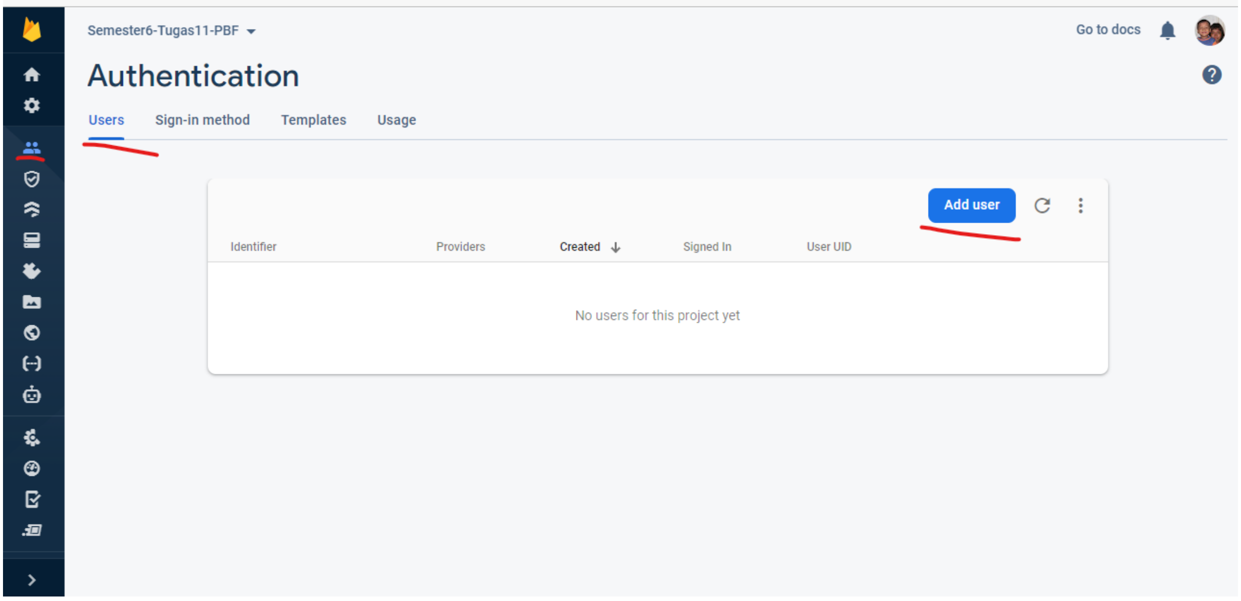
10. Yang terakhir adalah ProtectedRoute.s.



```
import React from "react";
import { Route, Redirect } from "react-router-dom";

const ProtectedRoute = ({
  component: Component,
  isAuthenticated,
  isVerifying,
  ...rest
}) => (
  <Route
    {...rest}
    render={(props) =>
      isVerifying ? (
        <div />
      ) : isAuthenticated ? (
        <Component {...props} />
      ) : (
        <Redirect
          to={{
            pathname: "/login",
            state: { from: props.location },
          }}
        />
      )
    }
  />
);

export default ProtectedRoute;
```

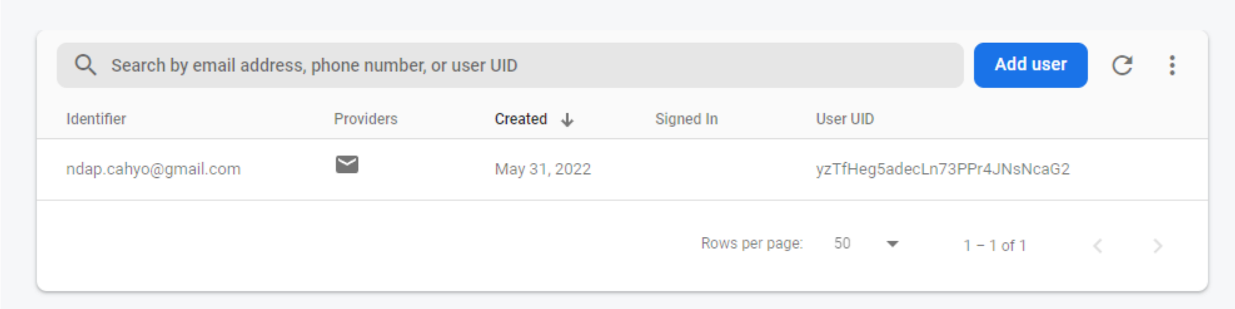
11. Running start.

12. Masuk ke firebase console kemudian add user.



Identifier	Providers	Created ↓	Signed In	User UID

No users for this project yet

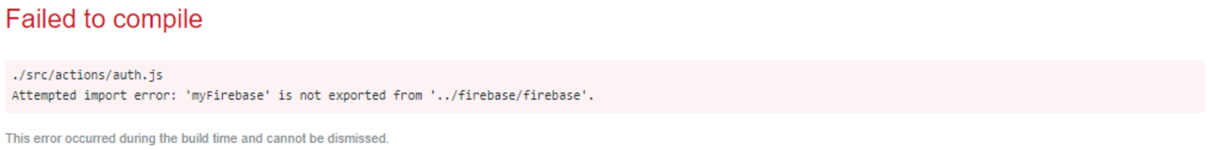


Identifier	Providers	Created ↓	Signed In	User UID
ndap.cahyo@gmail.com	✉	May 31, 2022		yzTfHeg5adecLn73PPr4JNsNcaG2

Rows per page: 50 1 – 1 of 1

Password: daffa123

13. Pilih npm start kemudian masukkan username dan password.



Failed to compile

```
./src/actions/auth.js
Attempted import error: 'myFirebase' is not exported from '../firebase/firebase'.
```

This error occurred during the build time and cannot be dismissed.

14. Kemudian masuk ke dalam halaman beranda

Gagal ☹