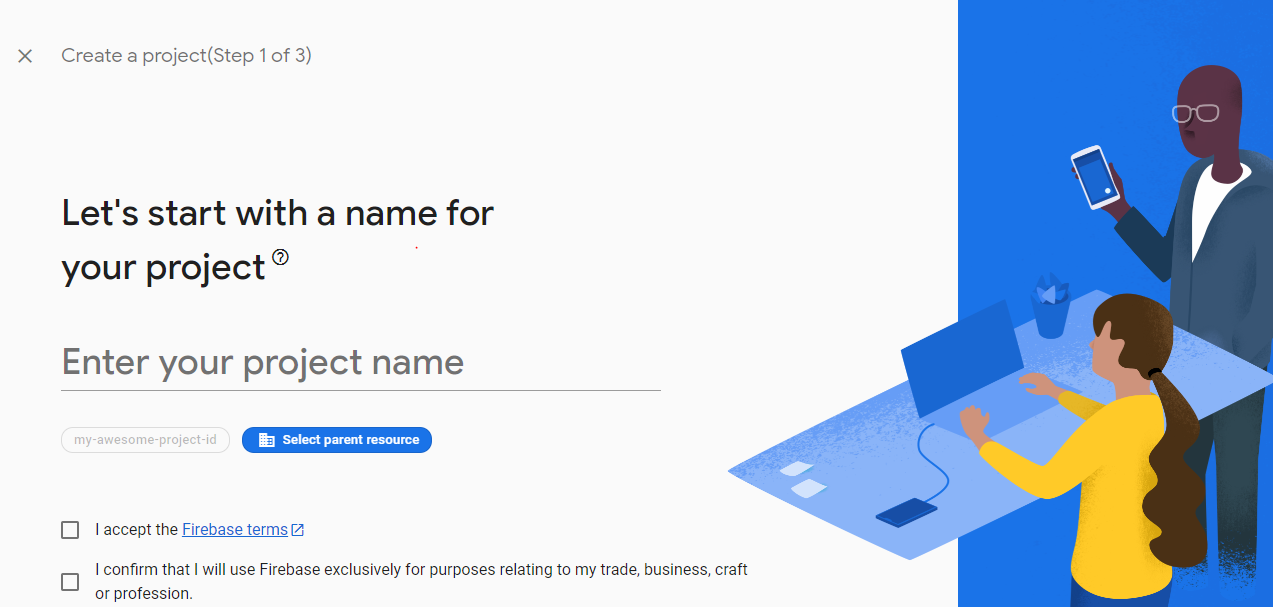
### Sign In with Google using Firebase

#### Objectives and Outcomes

In this exercise, you will configure and build a login function by Google account using Firebase. At the end of this exercise, you will be able to:

* Setup a project in Firebase
* Create a login function with Google using Firebase. Store all of the logic in Context for managing the authenticated state of the user.

#### Login and create a project in Firebase



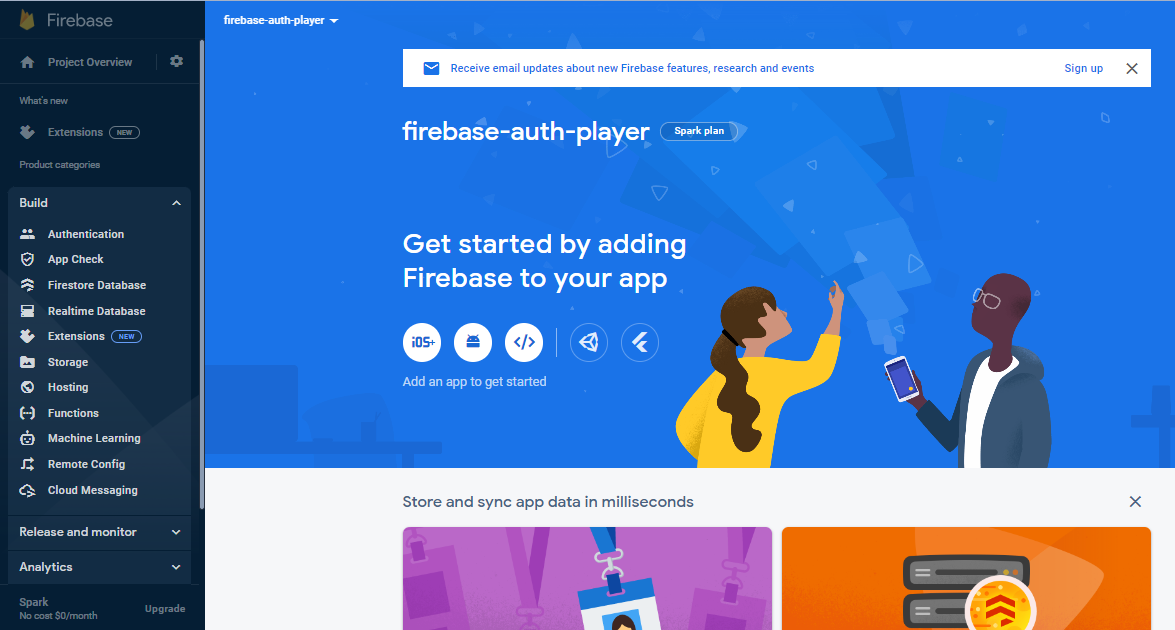
*Enter a project name*

Graphical user interface, text, application

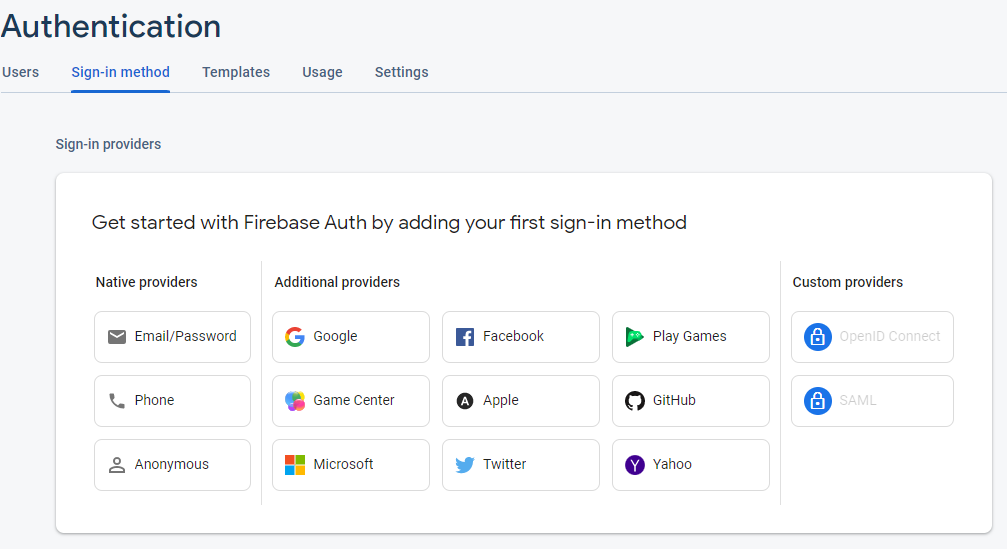
Description automatically generated

*Enable Google Analytics for this project*

#### Graphical user interface, text, application Description automatically generated



* Click on Authentication on the sidebar and click on Get Started to enable the module. Now you will be presented with various authentication options:



* Click on **Google**, enable it, and save it:

#### Install a few npm packages

npm i react-router-dom react-google-button

**SDK setup and configuration**

* Go to <https://console.firebase.google.com/> and select your project above

Graphical user interface, text, application, chat or text message

Description automatically generated

* Run: npm install firebase
* Create a file named firebase.js in src folder, copy the SDKs in Firebase and paste here

Text

Description automatically generated

import { initializeApp } from "firebase/app";

import { getAnalytics } from "firebase/analytics";

import { getAuth } from "firebase/auth";

const firebaseConfig = {

apiKey: "AIzaSyBK2OZR7c\_ xxxxxxxxxxx",

authDomain: "fir-auth-player.firebaseapp.com",

projectId: "fir-auth-player",

storageBucket: "fir-auth-player.appspot.com",

messagingSenderId: "xxxxxxxxxx",

appId: "xxxxxxxxxx",

measurementId: "G-8W8CEFKPF3"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

export const auth = getAuth();

* Managing the authenticated state of the user by Context API

Create a file name AuthConext.js in a context folder

import { useContext, createContext, useEffect, useState } from "react";

const AuthContext = createContext();

export const AuthContextProvider = ({children})=>{

return(

<AuthContext.Provider value={}>

{children}

</AuthContext.Provider>

)

}

export const UserAuth =()=>{

return useContext(AuthContext)

}

Let <AuthContextProvider> wrap your <App/> in Index.js file

……

import { AuthContextProvider } from './context/AuthContext';

……

<AuthContextProvider>

<BrowserRouter>

<App />

</BrowserRouter>

</AuthContextProvider>

* Access the Google auth Provider

Update the AuthContext.js file to access the Google auth Provider

import { useContext, createContext, useEffect, useState } from "react";

import { GoogleAuthProvider, signInWithPopup, signOut, onAuthStateChanged } from "firebase/auth";

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

const AuthContext = createContext();

export const AuthContextProvider = ({children})=>{

const [user, setUser]=useState({});

const googleSignIn = ()=>{

const provider = new GoogleAuthProvider();

signInWithPopup(auth,provider);

};

const logOut =()=>{

signOut(auth)

}

useEffect(()=>{

const unSubscribe = onAuthStateChanged(auth, (currentUser)=>{

setUser(currentUser);

console.log('User', currentUser);

});

return ()=>{

unSubscribe();

}

},[]);

return(

<AuthContext.Provider value={{googleSignIn, logOut, user}}>

{children}

</AuthContext.Provider>

)

}

export const UserAuth =()=>{

return useContext(AuthContext)

}

* Create Login component

import React, { useEffect } from 'react'

import GoogleButton from 'react-google-button';

import { UserAuth } from '../context/AuthContext';

import { useNavigate } from 'react-router-dom';

export default function Login() {

const {googleSignIn, user} = UserAuth();

const navigate = useNavigate()

const handleGoogleSignIn = async()=>{

try {

await googleSignIn();

} catch (error) {

console.log(error)

}

};

useEffect(()=>{

if(user!=null){

navigate('/dashboard')

}

},[user])

return (

<div>

<GoogleButton onClick={handleGoogleSignIn}/>

</div>

)

}

* Update the Navbar to show the login or logout link

import { Link } from 'react-router-dom';

import { UserAuth } from '../context/AuthContext';

……

const {user, logOut} = UserAuth();

const handleSignOut = async ()=>{

try {

await logOut()

} catch (error) {

console.log(error);

}

}

…..

{user?.displayName?(

<div>

<Tooltip title="Open settings">

<IconButton onClick={handleOpenUserMenu} sx={{ p: 0 }}>

<Avatar alt={user.email} src={user.photoURL} />

</IconButton>

</Tooltip>

<Menu

sx={{ mt: '45px' }}

id="menu-appbar"

anchorEl={anchorElUser}

anchorOrigin={{

vertical: 'top',

horizontal: 'right',

}}

keepMounted

transformOrigin={{

vertical: 'top',

horizontal: 'right',

}}

open={Boolean(anchorElUser)}

onClose={handleCloseUserMenu}

>

<MenuItem onClick={handleCloseUserMenu}>

<Typography textAlign="center" ><Link to='/dashboard' style={{textDecoration:"none"}}>Dashboard</Link></Typography>

</MenuItem>

<MenuItem>

<Typography textAlign="center" onClick={handleSignOut}>Logout</Typography>

</MenuItem>

</Menu>

</div>

):(

<Link to='/login' style={{textDecoration:"none"}} >

<Button

sx={{ my: 2, color: 'white', display: 'block' }}

>

Sign in

</Button>

</Link>

)}

* The full code of NavBar component

import \* as React from 'react';

import AppBar from '@mui/material/AppBar';

import Box from '@mui/material/Box';

import Toolbar from '@mui/material/Toolbar';

import IconButton from '@mui/material/IconButton';

import Typography from '@mui/material/Typography';

import Menu from '@mui/material/Menu';

import MenuIcon from '@mui/icons-material/Menu';

import Container from '@mui/material/Container';

import Avatar from '@mui/material/Avatar';

import Button from '@mui/material/Button';

import MenuItem from '@mui/material/MenuItem';

import Tooltip from '@mui/material/Tooltip';

import AdbIcon from '@mui/icons-material/Adb';

import { Link } from 'react-router-dom';

import { UserAuth } from '../context/AuthContext';

const pages = ['Home', 'About', 'News'];

function ResponsiveAppBar() {

const {user, logOut} = UserAuth();

const [anchorElUser, setAnchorElUser] = React.useState(null);

const handleSignOut = async ()=>{

try {

await logOut()

} catch (error) {

console.log(error);

}

}

const [anchorElNav, setAnchorElNav] = React.useState(null);

const handleOpenNavMenu = (event) => {

setAnchorElNav(event.currentTarget);

};

const handleCloseNavMenu = () => {

setAnchorElNav(null);

};

const handleOpenUserMenu = (event) => {

setAnchorElUser(event.currentTarget);

};

const handleCloseUserMenu = () => {

setAnchorElUser(null);

};

return (

<AppBar position="static">

<Container maxWidth="xl">

<Toolbar disableGutters>

<Typography

variant="h6"

noWrap

component="a"

href="/"

sx={{

mr: 2,

display: { xs: 'none', md: 'flex' },

fontFamily: 'monospace',

fontWeight: 700,

letterSpacing: '.3rem',

color: 'inherit',

textDecoration: 'none',

}}

>

LOGO

</Typography>

<Box sx={{ flexGrow: 1, display: { xs: 'flex', md: 'none' } }}>

<IconButton

size="large"

aria-label="account of current user"

aria-controls="menu-appbar"

aria-haspopup="true"

onClick={handleOpenNavMenu}

color="inherit"

>

<MenuIcon />

</IconButton>

<Menu

id="menu-appbar"

anchorEl={anchorElNav}

anchorOrigin={{

vertical: 'bottom',

horizontal: 'left',

}}

keepMounted

transformOrigin={{

vertical: 'top',

horizontal: 'left',

}}

open={Boolean(anchorElNav)}

onClose={handleCloseNavMenu}

sx={{

display: { xs: 'block', md: 'none' },

}}

>

{pages.map((page) => (

<MenuItem key={page} onClick={handleCloseNavMenu}>

<Typography textAlign="center">{page}</Typography>

</MenuItem>

))}

</Menu>

</Box>

<AdbIcon sx={{ display: { xs: 'flex', md: 'none' }, mr: 1 }} />

<Typography

variant="h5"

noWrap

component="a"

href=""

sx={{

mr: 2,

display: { xs: 'flex', md: 'none' },

flexGrow: 1,

fontFamily: 'monospace',

fontWeight: 700,

letterSpacing: '.3rem',

color: 'inherit',

textDecoration: 'none',

}}

>

LOGO

</Typography>

<Box sx={{ flexGrow: 1, display: { xs: 'none', md: 'flex' } }}>

{pages.map((page) => (

<Button

key={page}

onClick={handleCloseNavMenu}

sx={{ my: 2, color: 'white', display: 'block' }}

>

{page}

</Button>

))}

</Box>

<Box sx={{ flexGrow: 0 }}>

{user?.displayName?(

<div>

<Tooltip title="Open settings">

<IconButton onClick={handleOpenUserMenu} sx={{ p: 0 }}>

<Avatar alt={user.email} src={user.photoURL} />

</IconButton>

</Tooltip>

<Menu

sx={{ mt: '45px' }}

id="menu-appbar"

anchorEl={anchorElUser}

anchorOrigin={{

vertical: 'top',

horizontal: 'right',

}}

keepMounted

transformOrigin={{

vertical: 'top',

horizontal: 'right',

}}

open={Boolean(anchorElUser)}

onClose={handleCloseUserMenu}

>

<MenuItem onClick={handleCloseUserMenu}>

<Typography textAlign="center" ><Link to='/dashboard' style={{textDecoration:"none"}}>Dashboard</Link></Typography>

</MenuItem>

<MenuItem>

<Typography textAlign="center" onClick={handleSignOut}>Logout</Typography>

</MenuItem>

</Menu>

</div>

):(

<Link to='/login' style={{textDecoration:"none"}} >

<Button

sx={{ my: 2, color: 'white', display: 'block' }}

>

Sign in

</Button>

</Link>

)}

</Box>

</Toolbar>

</Container>

</AppBar>

);

}

export default ResponsiveAppBar;

* Create Protected component to protect routes

import React from 'react'

import { Navigate } from 'react-router-dom'

import { UserAuth } from '../context/AuthContext'

export default function Protected({children}) {

const {user} = UserAuth();

if(!user){

return <Navigate to='/'/>

}

return children;

}

* Protected route in App.js file

import Protected from './components/Protected';

…………….

<Route path='/dashboard' element={<Protected><Dashboard/></Protected>}></Route>

<Route path='/add' element={<Protected><Add/></Protected>}></Route>

<Route path='/edit' element={<Protected><Edit/></Protected>}></Route>

#### Conclusions

In this exercise, you learnt how to configure and build a login function by Google account using Firebase and protect some routes.