

Project Summary

To build a [React.js](#) Axios Book Search or Finder Web App Using Google Books API in Browser.

Tools Used

- React JS
- Bootstrap
- Google Book Api
- Axios
- Nodejs
- Express Js
- MongoDB

Create ReactJS project

Step 1

Now create a new React.js project by using the following command.

➤ `npm init react-app front`

Step 2

Open the newly created project in Visual Studio code and install react bootstrap in this project by using the following command.

➤ `npm init react-bootstrap --save`

Step 3

Now, open the index.js file and add import Bootstrap.

➤ `import 'bootstrap/dist/css/bootstrap.min.css';`

Step 4

Now Install the Axios library by using the following command.

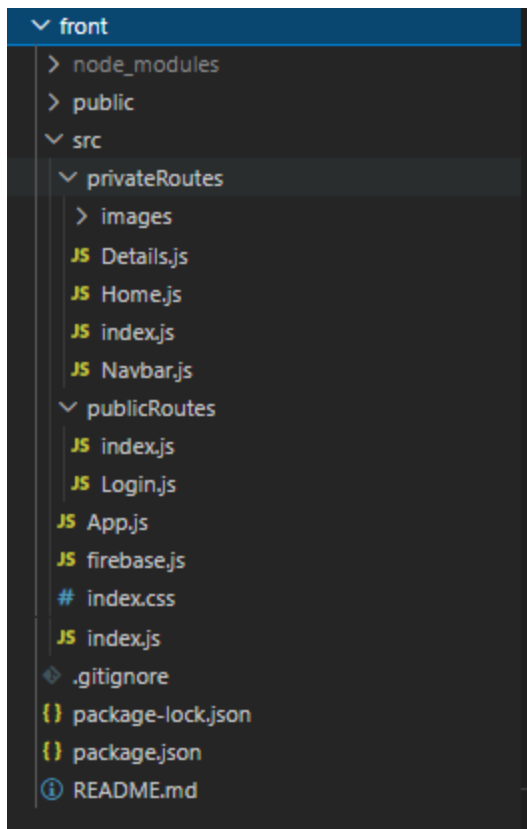
Axios is a lightweight HTTP client based on the \$http service.

```
➤ npm install --save axios
```

Step 4

Create Folders And Files given below

- Src
 - PrivateRoutes
 - Index.js
 - Home.js
 - Detail.js
 - Navbar.js
 - PublicRoutes
 - Index.js
 - Login.js



Step 5

Setup google Login

➤ npm I firebase

First you need to create a [new project in firebase console](#), Copy the Firebase SDK snippet and enable the Google Sign In Auth method under Authentication section.

- Create firebase.js file under src folder and write code as shown below

```
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 import { getAnalytics } from "firebase/analytics";
4 import { GoogleAuthProvider } from 'firebase/auth'
5 // TODO: Add SDKs for Firebase products that you want to use
6 // https://firebase.google.com/docs/web/setup#available-libraries
7
8 // Your web app's Firebase configuration
9 // For Firebase JS SDK v7.20.0 and later, measurementId is optional
10 const firebaseConfig = {
11   apiKey: "AIzaSyC15Kc0-KFK6-qgjsEK19B5BFnuTg6Z8uI",
12   authDomain: "book-cf1ca.firebaseio.com",
13   projectId: "book-cf1ca",
14   storageBucket: "book-cf1ca.appspot.com",
15   messagingSenderId: "457965605926",
16   appId: "1:457965605926:web:570a3c583375f2701fc2e7",
17   measurementId: "G-9JJ886T0MW"
18 };
19
20 // Initialize Firebase
21 export const app = initializeApp(firebaseConfig);
22 export const provider = new GoogleAuthProvider()
23
24 const analytics = getAnalytics(app);
```

Copy the Firebase SDK snippet from firebase app and paste it .

Step 6

Now Install the React Router DOM library by using the following command.

React Router DOM is an npm package that enables you to implement dynamic routing in a web app.

➤ `npm i react-router-dom`

Step 7

Setup React Router using React Router DOM

➤ In `index.js`

```
JS index.js X
front > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import {BrowserRouter} from 'react-router-dom' ←
6  import 'bootstrap/dist/css/bootstrap.min.css';
7
8  const root = ReactDOM.createRoot(document.getElementById('root'));
9  root.render(
10 <BrowserRouter>
11 <App />
12 </BrowserRouter>
13
14
15 );
16
17
18
```



➤ In src/privateRoutes/index.js

```
EXPLORER
OPEN EDITORS
X JS index.js front\src\privateRoutes
BOOK
> back
v front
  > node_modules
  > public
  v src
    v privateRoutes
      > images
      JS Details.js
      JS Home.js
      JS index.js
      JS Navbar.js
    v publicRoutes
      JS index.js
      JS Login.js
      JS App.js
      JS firebase.js
      # index.css
      JS index.js
      .gitignore
      {} package-lock.json
      {} package.json
      README.md

front > src > privateRoutes > JS index.js > index
1  import React from 'react'
2  import {Routes,Route} from 'react-router'
3  import Home from './Home'
4  import Details from './Details'
5
6  function index() {
7    return (
8      <Routes>
9        <Route path="/" element={<Home/>} />
10       <Route path="/detail/:id" element={<Details/>} />
11      </Routes>
12    )
13  }
14
15  export default index
```



Step 7

Google Login setup

➤ In Login.js

```
JS Login.js
front > src > publicRoutes > JS Login.js > Login

1 import React from 'react'
2 import { app, provider } from '../firebase'
3 import { signInWithPopup, getAuth } from 'firebase/auth'
4 import { Container, Row } from 'react-bootstrap'
5 import { AiOutlineGoogle } from 'react-icons/ai'
6 import axios from 'axios'
7 export default function Login() {
8   const auth = getAuth()
9
10  const signin = () => {
11    signInWithPopup(auth, provider).then(async (user) => {
12      console.log(user)
13      const { data } = await axios.post('http://localhost:8080/user/regis', { email: user.user.email })
14    }).catch((err) => {
15      console.log(err)
16    })
17  }
18
19  return (
20    <div>Login</div>
21    <Container>
22      <Row style={{ height: '100vh', display: 'flex', justifyContent: 'center', alignItems: 'center' }}>
23        <button className="shadow"
24          style={{
25            backgroundColor: '#407EE9', display: 'flex', alignItems: 'center', justifyContent: 'center', minWidth: '200px', color: 'white',
26            border: '1px solid white', padding: '7px 10px', fontSize: '15px', fontFamily: 'serif', fontWeight: 'bold', textAlign: 'center',
27            borderRadius: '7px', cursor: 'pointer'
28          }}
29          onClick={signin}> <AiOutlineGoogle size={30} style={{ marginLeft: '10px' }} /> Sign in With Google</button>
30      </Row>
31    </Container>
32  )
33 }
34
```

-
- Setting up google login using signInWithPopup, getAuth from firebase/auth by above method in the image.

➤ In App.js

```
front > src > JS App.js > default

1 import { getAuth, onAuthStateChanged } from 'firebase/auth';
2 import React, { useEffect, useState } from 'react';
3 import PrivateRoutes from './privateRoutes/index.js'
4 import PublicRoutes from './publicRoutes'
5
6
7 function App() {
8   const auth = getAuth()
9   const user = auth.currentUser
10  console.log('user', user)
11  const [ckuser, setckuser] = useState()
12  useEffect(() => {
13    onAuthStateChanged(auth, (user) => {
14      console.log('lll', user)
15      if (user?.emailVerified) {
16        setckuser(true)
17        // dispatch(loadNormalUser(user?.uid))
18      } else {
19        setckuser(false)
20        console.log('nouser')
21      }
22    })
23  })
24
25  }, [])
26  const isUserLoggedIn = ckuser;
27  return isUserLoggedIn ? <PrivateRoutes /> : <PublicRoutes />
28 }
29
30 export default App;
31
```



Step 7

Google Book Search Page Setup

- Using Google Book Api
 - [https://www.googleapis.com/books/v1/volumes?q=\\${value}&maxResults=20](https://www.googleapis.com/books/v1/volumes?q=${value}&maxResults=20)

- In `src/privateRoutes/Home.js`

```
src > privateRoutes / Home.js / Home

import React, { useState } from 'react'
import Navbar from './Navbar'
import { Row, Col, Container, Card, Button } from 'react-bootstrap'
import { ImSearch } from 'react-icons/im'
import axios from 'axios'
import book from './images/book.jpg'
import { NavLink } from 'react-router-dom'
import { getAuth, onAuthStateChanged } from 'firebase/auth';

function Home() {
  const [value, setValue] = useState()
  const [booklist, setbooklist] = useState()
  const auth = getAuth()
  const user = auth.currentUser
  console.log('user', user)
  const search = async () => {

    const { data } = await axios.get(`https://www.googleapis.com/books/v1/volumes?q=${value}&maxResults=20`)

    console.log(data)
    setbooklist(data.items)
    const dataupdate = await axios.patch(`http://localhost:8080/user/update`, { email: user.email, key: value })

  }

  return (
    <div className="home">
      <Navbar />
      <div className="home-main">
        <div className="input-div">
          <input type="text" value={value} onChange={e => setValue(e.target.value)} />
          <button onClick={search}>ImSearch />
        </div>
        <div>
          <Container>
            <Row>
              {
                booklist.map((v, i) => {
                  return (
                    <Card key={i}>
                      <Card.Img alt={v.volumeInfo.imageLinks.thumbnail} src={v.volumeInfo.imageLinks.thumbnail} />
                      <Card.Body>
                        <Card.Title>{v.volumeInfo.title}</Card.Title>
                        <Card.Text>{v.volumeInfo.description}</Card.Text>
                        <button variant="outline-info">More Info</button>
                      </Card.Body>
                    </Card>
                  )
                })
              }
            </Row>
          </Container>
        </div>
      </div>
    </div>
  )
}

export default Home
```

-
- Const `[value, setValue] = useState()` is used to store the search input value.
- Search function use google book api, with axios to get the search books output.

Step 8

Google Book Detail Page Setup

- In `src/privateRoutes/Detail.js`

```

function Details() {
  const [bookDetails, setBookDetails] = useState()
  const [id] = useParams()
  useEffect(() => {
    if (id) {
      detail()
    }
  }, [id])

  const detail = async () => {
    const { data } = await axios.get('https://www.googleapis.com/books/v1/volumes/$id')
    console.log(data)
    // setBookList(data.items)
    setBookDetails(data)
  }

  return (
    <div className="detail">
      <Header />
      <Container className="mt-3">
        <Row>
          <Col sm={12} md={4} xl={3} style={{display: 'flex', justify-content: 'center', align-items: 'center'}}>
            <img src={bookDetails?.volumeInfo?.imageLinks?.thumbnail} bookDetails?.volumeInfo?.imageLinks?.thumbnail:book className="img-fluid" />
          </Col>
          <Col>
            <Col sm={12} md={8} xl={9}>
              <h2 className="title">{bookDetails?.volumeInfo?.title}</h2>
              <h2 className="subtitle">{bookDetails?.volumeInfo?.subtitle}</h2>
              <p style={{fontWeight: 'bold'}}>Authors: {bookDetails?.volumeInfo?.authors?.map((v) => {
                return <span style={{fontWeight: 'lighter'}}>{v}, </span>
              })}</p>
              <p>{bookDetails?.volumeInfo?.publishedDate}</p>
              <p style={{fontWeight: 'bold'}}>Publisher: {bookDetails?.volumeInfo?.publisher}</p>
              <p style={{fontWeight: 'bold'}}>Page Count: {bookDetails?.volumeInfo?.pageCount}</p>
              <div style={{display: 'flex', justify-content: 'center', align-items: 'center'}}>
                <button variant="outline-info" href={`https://books.google.co.in/books?id=${id}&printsec=frontcover&source=gbp_summary_r&ad=IN&onpage&f=false`}>Preview</button>
                <button variant="outline-info" href={bookDetails?.volumeInfo?.infoLink} className="ml-3">More Info</button>
              </div>
            </Col>
          </Row>
          <Row>
            <Col sm={12}>
              <h2 className="descrip">Description</h2>
              <p>{bookDetails?.volumeInfo?.description}</p>
            </Col>
          </Row>
        </Container>
      </div>
    )
  }
}

```

-
- detail function use google book api, with axios to get the search book details output with respect to the book id.

Create Backend for keywords store

- ❓ Setup backend using the below command:
 - ❓ npm init back
- ❓ After than cd back
- ❓ Then install the required modules by below command
 - ❓ Npm init mongoose express cors
- After that create :
 - Database page in src/db/db.js

```

back > src > db > JS db.js > ...
1  const mongoose = require('mongoose')
2
3
4  mongoose.connect("mongodb+srv://sanz:sannu05@cluster0.s5xci.mongodb.net/book?retryWrites=true&w=majority", {
5    // useCreateIndex: true,
6    // useFindAndModify: false,
7    useNewUrlParser: true,
8    useUnifiedTopology: true
9  }).then(() => {
10    console.log("db connected")
11  }).catch((err) => {
12    console.log("db failed", err)
13  })
14
15
16
17 |

```

- Model in src/models/main.js


```

back > src > models > JS main.js > <unknown>
1  const mongoose = require('mongoose')
2  const userSchema = mongoose.Schema({
3    email:{type:String,required:true},
4    keywords:{type:Array}
5  })
6
7
8  const Users = mongoose.model('Users',userSchema)
9  module.exports = Users

```

- Routes in src/routes/main.js

```

back > src > routes > JS main.js > route.post('/regis') callback
1  const express = require('express')
2  const route = express.Router()
3  const User = require('../models/main')
4  route.get('/',(req,res)=>{
5    res.json('hello from main routes')
6  })
7
8  route.post('/regis',async(req,res)=>{
9    try {
10     const {email} = req.body
11     const findx = await User.findOne({email})
12     if(findx){
13
14     }else{
15     const data = new User({email})
16     const savedata = await data.save()
17     console.log(savedata)
18     res.json(savedata)
19     }
20   } catch (error) {
21
22   }
23 }
24 })
25 route.patch('/update',async(req,res)=>{
26   try {
27     const {email,key} = req.body
28     const update = await User.findOneAndUpdate({email},{
29       $push:{"keywords":key}
30     })
31     console.log(update)
32   } catch (error) {
33
34   }
35 }
36 })
37
38
39
40 module.exports = route

```

- The server in src/index.js

```

back > src > JS index.js > ...
1  const express = require('express')
2  const app = express()
3  const port = 8080
4  const cors = require('cors')
5  const db = require('./db/db')
6  const mainroute = require('./routes/main')
7
8  app.use(express.json())
9  app.use(cors())
10
11
12  app.use('/user', mainroute)
13  app.get('/get', (req, res) => {
14    |   res.json('hello from server')
15    | })
16
17
18  app.listen(port, () => {
19    |   console.log(`port running at ${port}`)
20    | })

```

- Using this routes in front end using axios
 - In App.js (front) line no :14 , to add register user data

```

JS Loginjs
front > src > publicRoutes > JS Loginjs > Login
1  import React from 'react'
2  import { app, provider } from '../firebase'
3  import { signInWithPopup, getAuth } from 'firebase/auth'
4  import { Container, Row } from 'react-bootstrap'
5  import { AiOutlineGoogle } from 'react-icons/ai'
6  import axios from 'axios'
7  export default function Login() {
8    const auth = getAuth()
9
10   const signin = () => {
11     signInWithPopup(auth, provider).then(async (user) => {
12       console.log(user)
13       const { data } = await axios.post('http://localhost:8080/user/regist', { email: user.user.email })
14     }).catch((err) => {
15       console.log(err)
16     })
17   }
18
19   return (
20     // <div>Login</div>
21     <Container>
22       <Row style={{ height: '100vh', display: 'flex', justifyContent: 'center', alignItems: 'center' }}>
23         <button className='shadow'
24           style={{
25             backgroundColor: '#407EE9', display: 'flex', alignItems: 'center', justifyContent: 'center', minWidth: '200px', color: 'white',
26             border: '1px solid white', padding: '7px 10px', fontSize: '15px', fontFamily: 'serif', fontWeight: 'bold', textAlign: 'center',
27             borderRadius: '7px', cursor: 'pointer'
28           }} onClick={signin}> <AiOutlineGoogle size={30} style={{ marginLeft: '10px' }} /> Sign in With Google</button>
29       </Row>
24     </Container>
25   )
26
27
28
29
30
31
32
33
34

```

- In src/privateRoutes/Home.js (front) to store key-words

```

src > privateRoutes > Home.js > Home

import React, { useState } from 'react'
import Navbar from './Navbar'
import { Row, Col, Container, Card, Button } from 'react-bootstrap'
import { ImSearch } from 'react-icons/im'
import axios from 'axios'
import book from './images/book.jpg'
import { NavLink } from 'react-router-dom'
import { getAuth, onAuthStateChanged } from "firebase/auth";

function Home() {
  const [value, setValue] = useState()
  const [booklist, setbooklist] = useState()
  const auth = getAuth()
  const user = auth.currentUser
  console.log('userxxx', user)
  const search = async () => {

    const { data } = await axios.get(`https://www.googleapis.com/books/v1/volumes?q=${value}&maxResults=20`)

    console.log((data))
    setbooklist(data.items)
    const dataupdate = await axios.patch(`http://localhost:8080/user/update`, { email: user.email, key: value })
  }
}

```

Then Run the front-end: `cd front -> npm start`

& Backend : `cd back node src/index.js`

Conclusion

A [React.js](https://reactjs.org/) Axios Book Search or Finder Web App Using Google Books API in Browser has been built and tested successfully.

ScreenShots

 Sign In With Google

accounts.google.com/v/oauth2/auth?identifier=response_type=code&id=...

Sign in with Google

Sign in

to continue to book-cf1.ca.firebaseio.com

Email or phone

Forgot email?

Create account

Next

English (United States) ▾

Help Privacy Terms

Book

Google Books

React

Fullstack React

Learn React, Redux, GraphQL, and more with this fullstack guide. Includes a complete course and a project to build a real-world application.

More info

React: In Book

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

Learning React

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

Learning React

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

The Road to React

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React in Action

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React by Example - R...

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React in Essentials

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

Learning React

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React and React Native

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

Pro React 16

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React: Up & Running

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React Quickly

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

Beginning React (2nd ed...)

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

Learning React Native

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React Projects

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React Explained

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

Getting Started with R...

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info

React 16

Learn React in a book that is easy to read and understand. Includes a complete course and a project to build a real-world application.

More info



Firefly

Heartstopping chase thriller & winner of the Wilbur Smith Award

Authors: Henry Porter,

Published Date :2018-06-19

Publisher :Hachette UK

Page Count :400

[Preview](#)[More Info](#)

Description

<p>The boy could help them bring down a deadly ISIS terror cell. They just have to find him before the other side does.

WINNER OF THE 2019 WILBUR SMITH ADVENTURE WRITING PRIZE

'A welcome return ...
</i>Firefly</i> is both timely and terrific' Mick Herron

'An espionage master' Charles Cumming

<i>'</i></i>British espionage fiction is the best in the world, and Porter is part of the reason why' Lee Child

From the refugee camps of Greece to the mountains of Macedonia, a thirteen year old boy is making his way to Germany and safety. Codenamed 'Firefly', he holds vital intelligence: unparalleled insight into a vicious ISIS terror cell, and details of their plans. But the terrorists are hot on his trail, determined he won't live to pass on the information.

When MI6 become aware of Firefly and what he knows, the race is on to find him. Paul Samson, ex-MI6 agent and now private eye, finds himself recruited to the cause. Fluent in Arabic thanks to his Lebanese heritage, Samson's job is to find Firefly, win his trust and get him to safety.

A devastatingly timely thriller following the refugee trail from Syria to Europe,<i> Firefly</i> is a sophisticated, breathtaking race against time from the acclaimed and award-winning author of <i>Brandenburg</i> and <i>The Dying Light</i>.</p>