

qwertyuiopasdfghjklzxcvbnmqwertyui
opasdfghjklzxcvbnmqwertyuiopasdfgh
jklzxcvbnmqwertyuiopasdfghjklzxcvb

WEB PROGRAMMING PROJECT
DevZone

nmqwertyuiopasdfghjklzxcvbnmqwer
tyuiopasdfghjklzxcvbnmqwertyuiopas
dfghjklzxcvbnmqwertyuiopasdfghjklzx
cvbnmqwertyuiopasdfghjklzxcvbnmq
wertyuiopasdfghjklzxcvbnmqwertyuio
pasdfghjklzxcvbnmqwertyuiopasdfghj
klzxcvbnmqwertyuiopasdfghjklzxcvbn
mqwertyuiopasdfghjklzxcvbnmqwerty

uiopasdfghjklzxcvbnmqwertyuiopasdf
ghjklzxcvbnmqwertyuiopasdfghjklzxc
vbnmqwertyuiopasdfghjklzxcvbnmrty
uiopasdfghjklzxcvbnmqwertyuiopasdf
ghjklzxcvbnmqwertyuiopasdfghjklzxc

WEB PROGRAMMING PROJECT

DevBook

A Social Media for Developers who want to connect and collaborate on projects.

Technologies Used:

- MongoDB
- ExpressJS
- ReactJS
- NodeJS
- Heroku

Routes:

GET api/auth/

POST api/auth/

POST api/users/

GET api/profile/me

POST api/profile/

GET api/profile/

GET api/profile/user/:user_id

DELETE api/profile/

PUT api/profile/experience

DELETE api/profile/experience/:exp_id

PUT api/profile/education

DELETE api/profile/education/:edu_id

GET api/profile/github/:username

POST api/posts

GET api/posts

GET api/posts/:id

DELETE api/posts/:id

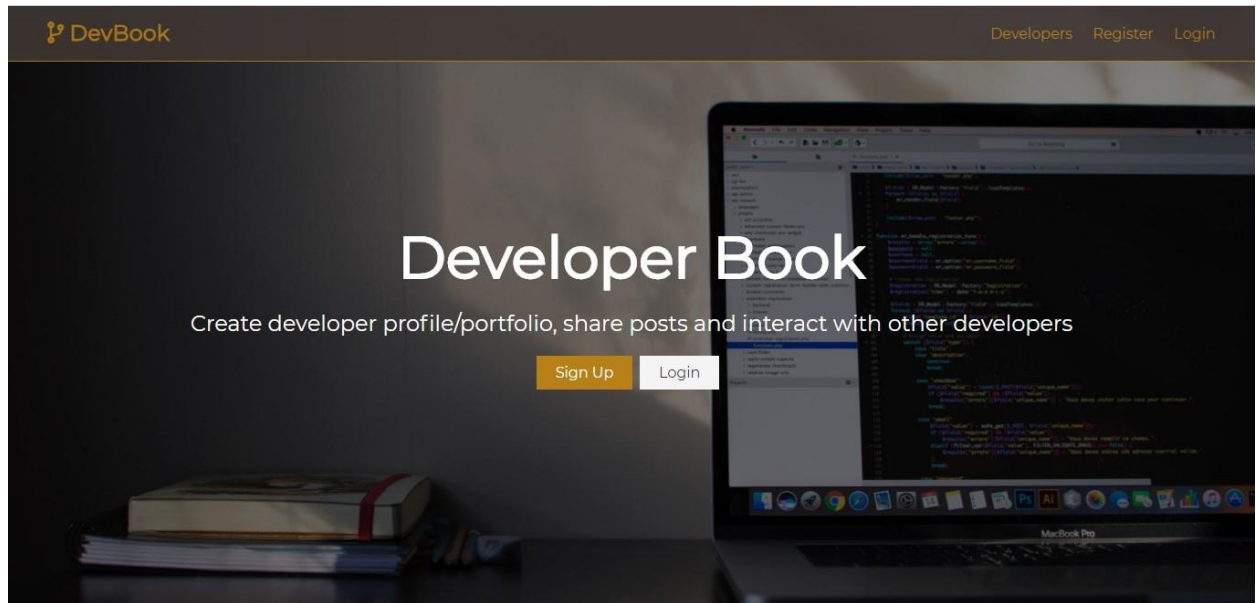
PUT api/posts/like/:id

PUT api/posts/unlike/:id

POST api/posts/comment/:id

DELETE api/posts/comment/:id/:comment_id Demo Screenshots

Home Page



Registration

DevBook

Developers Register Login

Sign Up


Create Your Account

Enter your Github username for a DP eg. Mugilan-Codes

Register


Already Have an Account? [Sign In](#)

Login

DevBook

DevelopersRegisterLogin

Sign In


 Sign Into Your Account



Login

Don't Have an Account? [Sign Up](#)


<https://hidden-spire-69243.herokuapp.com/login>

Dashboard

DevBook

DevelopersPosts Dashboard  Logout

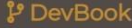


Dashboard

 Welcome Aravinth

You have not setup a Profile. Please add some info

Create Profile

Create Profile

 Developers Posts  Dashboard  Logout

Create Your Profile

♥ Let's get some information about you

* = Required Fields

Student or Learning

Give us an idea of where you are at in your career

Company

Could be your own company or one you work for

Website




Could be your own or a company website

Location

City & state suggested (eg. Boston, MA)

Python, Javascript

Please use comma separated values (eg. HTML,CSS,JavaScript,PHP)

 DevBook's content is open to all and a comment will include your username Developers Posts  Dashboard  Logout

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98


99


100


Tell us a little about yourself


Add Social Network Links


Optional


 Twitter URL

 Facebook URL

 YouTube URL

 LinkedIn URL

 Reddit URL

 Instagram URL

Submit

Go Back

Dashboard

Welcome Aravinth

[Edit Profile](#)[Add Experience](#)[Add Education](#)

Experience Creadentials

Company	Title	Years	

Education Credentials

School	Degree	Field Of Study	Years	

[Delete My Account](#)

Add Experience and Add Education

Add An Experience

Add any developer/programming positions that you have had in the past

* = required field

From Date



☒ Current Job


To Date


Profiles

DevBook

DevelopersRegisterLogin

Developers

 Browse and Connect with Developers



Mugilan E.S.

Student or Learning

Chennai, Tamilnadu, IN


View Profile

✓ HTML

✓ CSS

✓ JavaScript

✓ Python



Will Adams

Other at Bushblade Knives

UK

View Profile


✓ React

✓ Vue

✓ Linux

DevBook

DevelopersPostsDashboardLogout




klnkl

Developer at 456

456456

View Profile

✓ 456



Aravinth

Student or Learning

View Profile

✓ Python


✓ Javascript

Profile

DevBook

DevelopersRegisterLogin






Back to Profiles



Mugilan E.S.

Student or Learning

Chennai, Tamilnadu, IN



DevBook

DevelopersRegisterLogin

About Mugilan E.S.

This is the first bio in this app

Skill Set

✓ HTML

✓ CSS

✓ JavaScript

✓ Python

✓ React

✓ Node

Experience

No Experience Credentials

Education

Anna University
13/Sep/2019 - Now
Degree: Master's Degree
Field of Study: Compute Applications
Description: Doing Postgraduate degree in Masters of Computer Application.
Aiming to complete the degree by 2021

DevBook

Aiming to complete the degree by 2021DevelopersRegisterLogin

DG Vaishnav

04/Jun/2016 - 04/May/2019
Degree: Bachelor's Degree
Field of Study: Computer Science
Description: Did Undergraduate degree on B.Sc.(CSC) in DGVC, Chennai.

Github Repos

dotfiles

Under Work

Stars: 0

Watchers: 0

Forks: 0

tribute-page

A FreeCodeCamp Project

Stars: 0

Watchers: 0

Forks: 0

Post

DevBook

DevelopersPostsDashboardLogout


Posts

♥ Welcome to the Developer Community

Say Something...

Hello, World!

Submit



This is my first Post...

Posted on 18/Apr/2020


4

Discussion 3

DevBook

DevelopersPostsDashboardLogout

Submit




Hello, World!

Posted on 18/Aug/2020

1

Discussion

Aravinth



This is my first Post...

Posted on 18/Apr/2020

4

Discussion 3

Mugilan E.S.


<https://hidden-spire-69243.herokuapp.com/posts/5f3b9b2bb2ac330017d5bc54>

Comment

DevBook

Hello, World!

DevelopersPostsDashboardLogout




Aravinth

Posted on 18/Aug/2020

Leave a Comment

Comment on this Post


Submit



Aravinth

This is my comment!

Posted on 18/Aug/2020



BACKEND API – Node.Js, Express.JS, MongoDB

server.js – Entry Point

```

const express = require('express');
const connectDB = require('./config/db');
const path = require('path');

const app = express();

// Connect to Database
connectDB();

// Initialize Middleware
app.use(express.json({ extended: false }));

// Define Routes
app.use('/api/users', require('./routes/api/users'));
app.use('/api/auth', require('./routes/api/auth'));
app.use('/api/profile', require('./routes/api/profile'));
app.use('/api/posts', require('./routes/api/posts'));

// Serve Static assets in production
if (process.env.NODE_ENV === 'production') {
  // Set Static Folder
  app.use(express.static('client/build'));

  app.get('*', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'client', 'build', 'index.html'));
  });
}

// SERVER
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server started on PORT ${PORT}`));

```

config/db.js – Database connection

```

const mongoose = require('mongoose');
const config = require('config');
const db = config.get('mongoURI');

// Connection to MongoDB
const connectDB = async () => {
  try {
    await mongoose.connect(db, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
      useCreateIndex: true,
      useFindAndModify: false,
    });
    console.log('MongoDB Connected...');
  } catch (err) {
    console.error(err.message);
    // Exit process with failure
    process.exit(1);
  }
};

module.exports = connectDB;

```

config/default.json

```
{  
  "mongoURI": "<Mongo_DB_Connection_URL>"  
}
```

Define Routes In these files routes/api/auth.js

– Auth Route routes/api/users.js – User

Route routes/api/profile.js – Profile Route

routes/api/posts.js – Posts Route

models/User.js – User Model

```
const mongoose = require('mongoose');  
  
const UserSchema = new mongoose.Schema({  
  name: {  
    type: String,  
    required: true,  
  },  
  email: {  
    type: String,  
    required: true,  
    unique: true,  
  },  
  githubusername: {  
    type: String,  
  },  
  avatar: {  
    type: String,  
  },  
  password: {  
    type: String,  
    required: true,  
  },  
  date: {  
    type: Date,  
    default: Date.now,  
  },  
});  
  
module.exports = User = mongoose.model('user', UserSchema);
```

```
const mongoose = require('mongoose');

const ProfileSchema = new mongoose.Schema({
  user: { type: mongoose.Schema.Types.ObjectId, ref: 'user' },
  company: { type: String },
  website: { type: String },
  location: { type: String },
  status: { type: String, required: true },
  skills: { type: [String], required: true },
  bio: { type: String },
  githubusername: { type: String },
  experience: [
    {
      title: { type: String, required: true },
      company: { type: String, required: true },
      location: { type: String },
      from: { type: Date, required: true },
      to: { type: Date },
      current: { type: Boolean, default: false },
      description: { type: String },
    }
  ],
  education: [
    {
      school: { type: String, required: true },
      degree: { type: String, required: true },
      fieldofstudy: { type: String, required: true },
      from: { type: Date, required: true },
      to: { type: Date },
      current: { type: Boolean, default: false },
      description: { type: String },
    }
  ],
  social: {
    youtube: { type: String },
    twitter: { type: String },
    facebook: { type: String },
    linkedin: { type: String },
    instagram: { type: String },
    reddit: { type: String },
  },
  date: {
    type: Date,
    default: Date.now,
  },
});

module.exports = Profile = mongoose.model('profile', ProfileSchema);
```


models/Post.js – Post Model

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const PostSchema = new Schema({
  user: { type: Schema.Types.ObjectId, ref: 'users' },
  text: { type: String, required: true },
  name: { type: String },
  avatar: { type: String },
  likes: [{ user: { type: Schema.Types.ObjectId, ref: 'users' } }],
  comments: [
    {
      user: { type: Schema.Types.ObjectId, ref: 'users' },
      text: { type: String, required: true },
      name: { type: String },
      avatar: { type: String },
      date: { type: Date, default: Date.now },
    },
  ],
  date: { type: Date, default: Date.now },
});

module.exports = Post = mongoose.model('post', PostSchema);
```

middleware/auth.js – Auth Middleware

```
const jwt = require('jsonwebtoken');
const config = require('config');

module.exports = (req, res, next) => {
  const token = req.header('x-auth-token');

  if (!token) {
    return res.status(401).json({ msg: 'No Token, Authorization Denied' });
  }

  try {
    const decoded = jwt.verify(token, config.get('jwtSecret'));

    req.user = decoded.user;
    next();
  } catch (err) {
    res.status(401).json({ msg: 'Token is invalid' });
  }
};
```

package.json – The Packages and Scripts used

```
{
  "name": "devbook",
  "version": "1.0.0",
  "description": "Social Media for Developers",
  "main": "server.js",
  "scripts": {
    "start": "node server",
    "server": "nodemon server",
    "client": "npm start --prefix client",
    "dev": "concurrently \"npm run server\" \"npm run client\"",
    "heroku-postbuild": "NPM_CONFIG_PRODUCTION=false npm install --prefix client && npm run build --prefix client"
  },
  "author": "Mugilan E.S.",
  "license": "MIT",
  "dependencies": {
    "axios": "^0.19.2",
    "bcryptjs": "^2.4.3",
    "config": "^3.3.1",
    "express": "^4.17.1",
    "express-validator": "^6.4.0",
    "jsonwebtoken": "^8.5.1",
    "mongoose": "^5.9.7",
    "normalize-url": "^5.0.0"
  },
  "devDependencies": {
    "concurrently": "^5.1.0",
    "nodemon": "^2.0.3"
  }
}
```


FRONTEND – ReactJS

client/package.json

```
{
  "name": "client",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^4.2.4",
    "@testing-library/react": "^9.5.0",
    "@testing-library/user-event": "^7.2.1",
    "axios": "^0.19.2",
    "moment": "^2.24.0",
    "react": "^16.13.1",
    "react-dom": "^16.13.1",
    "react-moment": "^0.9.7",
    "react-redux": "^7.2.0",
    "react-router-dom": "^5.1.2",
    "react-scripts": "3.4.1",
    "redux": "^4.0.5",
    "redux-devtools-extension": "^2.13.8",
    "redux-thunk": "^2.3.0",
    "uuid": "^7.0.3"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "proxy": "http://localhost:5000"
}
```

client/src/index.js

```
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

client/src/store.js

```
import { createStore, applyMiddleware } from 'redux';
import { composeWithDevTools } from 'redux-devtools-extension';
import thunk from 'redux-thunk';
import rootReducer from './reducers';

const initialState = {};
const middleware = [thunk];

export const store = createStore(
  rootReducer,
  initialState,
  composeWithDevTools(applyMiddleware(...middleware))
);
```

client/src/utils/setAuthToken.js

```
import axios from 'axios';

const setAuthToken = (token) => {
  if (token) {
    axios.defaults.headers.common['x-auth-token'] = token;
  } else {
    delete axios.defaults.headers.common['x-auth-token'];
  }
};

export default setAuthToken;
```

client/src/App.js

```
import React, { Fragment, useEffect } from 'react';
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';

import Navbar from './components/layout/Navbar';
import Landing from './components/layout/Landing';
import Routes from './components/routing/Routes';

// Redux
import { Provider } from 'react-redux';
import { store } from './store';
import { loadUser } from './actions/auth';
import setAuthToken from './utils/setAuthToken';

import './App.css';

if (localStorage.token) {
  setAuthToken(localStorage.token);
}

const App = () => {
  useEffect(() => {
    store.dispatch(loadUser());
  }, []);

  return (
    <Provider store={store}>
      <Router>
        <Fragment>
          <Navbar />
          <Switch>
            <Route exact path="/" component={Landing} />
            <Route component={Routes} />
          </Switch>
        </Fragment>
      </Router>
    </Provider>
  );
};

export default App;
```

client/src/components/routing/Routes.js

```
import React from 'react';
import { Route, Switch } from 'react-router-dom';

import Login from '../auth/Login';
import Register from '../auth/Register';
import Alert from '../layout/Alert';
import Dashboard from '../dashboard/Dashboard';
import CreateProfile from '../profile-form/CreateProfile';
import EditProfile from '../profile-form/EditProfile';
import AddExperience from '../profile-form/AddExperience';
import AddEducation from '../profile-form/AddEducation';
import Profiles from '../profiles/Profiles';
import Profile from '../profile/Profile';
import Posts from '../posts/Posts';
import Post from '../post/Post';
import NotFound from '../layout/NotFound';

import PrivateRoute from '../routing/PrivateRoute';

const Routes = () => {
  return (
    <section className='container'>
      <Alert />
      <Switch>
        <Route exact path='/register' component={Register} />
        <Route exact path='/login' component={Login} />
        <Route exact path='/profiles' component={Profiles} />
        <Route exact path='/profile/:id' component={Profile} />
        <PrivateRoute exact path='/dashboard' component={Dashboard} />
        <PrivateRoute exact path='/create-profile' component=
{CreateProfile} />
        <PrivateRoute exact path='/edit-profile' component={EditProfile} />
        <PrivateRoute exact path='/add-experience' component=
{AddExperience} />
        <PrivateRoute exact path='/add-education' component={AddEducation}
/>
        <PrivateRoute exact path='/posts' component={Posts} />
        <PrivateRoute exact path='/posts/:id' component={Post} />
        <Route component={NotFound} />
      </Switch>
    </section>
  );
};

export default Routes;
```

client/src/components/routing/PrivateRoute.js

```
import React from 'react';
import PropTypes from 'prop-types';
import { connect } from 'react-redux';
import { Route, Redirect } from 'react-router-dom';

const PrivateRoute = ({
  component: Component,
  auth: { isAuthenticated, loading },
  ...rest
}) => (
  <Route
    {...rest}
    render={((props) =>
      !isAuthenticated && !loading ? (
        <Redirect to='/login' />
      ) : (
        <Component {...props} />
      )
    )}
  />
);

PrivateRoute.propTypes = { auth: PropTypes.object.isRequired };

const mapStateToProps = (state) => ({ auth: state.auth });

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