VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE3999
Internet and Web Programming
J-Component

WhipUp
Online Portfolio Builder

Submitted to:
Prof. Lydia Jane G
Slot: B2/ L21+L22

Submitted by:
Charizma Gupta
17BCI0157

Online Portfolio Builder

Users will be capable of **signing up, logging in** and **hosting their portfolios** on the website. They will just have to edit a pre-existing template portfolio to reflect their own work by uploading project links and writing appropriate descriptions.
Node.js, Express & MongoDB were used to build the backend. **POSTMAN** was used to help with testing the backend while the frontend hadn't been fully developed.
React & Redux were used to build the backend.

What I learned while developing WhipUp:

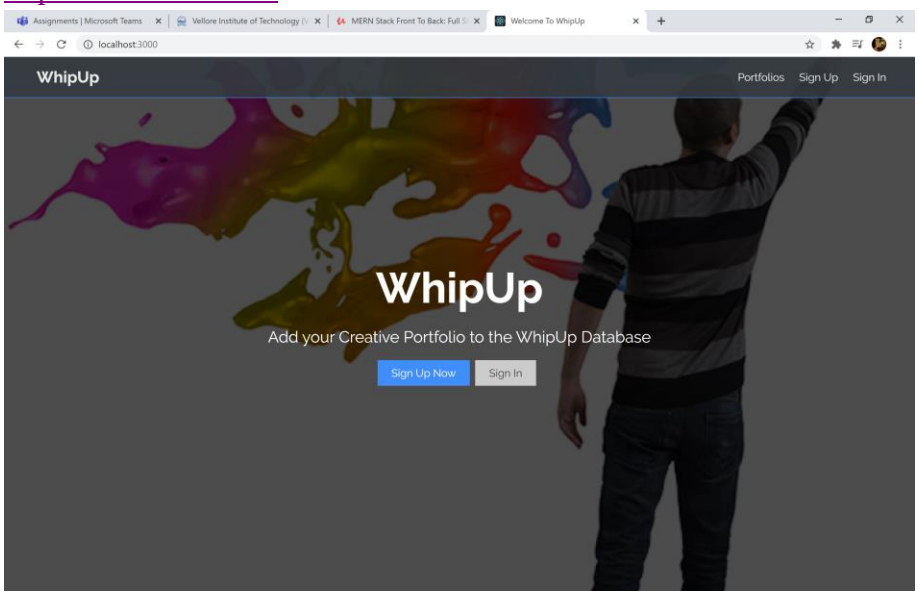
- Building an extensive backend API with Node.js & Express
- Protecting routes/endpoints with JWT (JSON Web Tokens)
- Extensive API testing with Postman
- Integrating React with our backend in an elegant way, creating a great workflow
- Building a frontend to work with the API
- Using Redux for app state management
- Creating reducers and actions for our resources
- Creating many container components that integrate with Redux
- Testing with the Redux Chrome extension

Functionalities

1. Sign In/Sign Up
2. View all Portfolios
3. Create/Edit your own Portfolio
4. Add/Edit Projects
5. Add/Edit Title, Tags and Description for each individual project
6. Has a Page Not Found result for an unknown url

Home Page

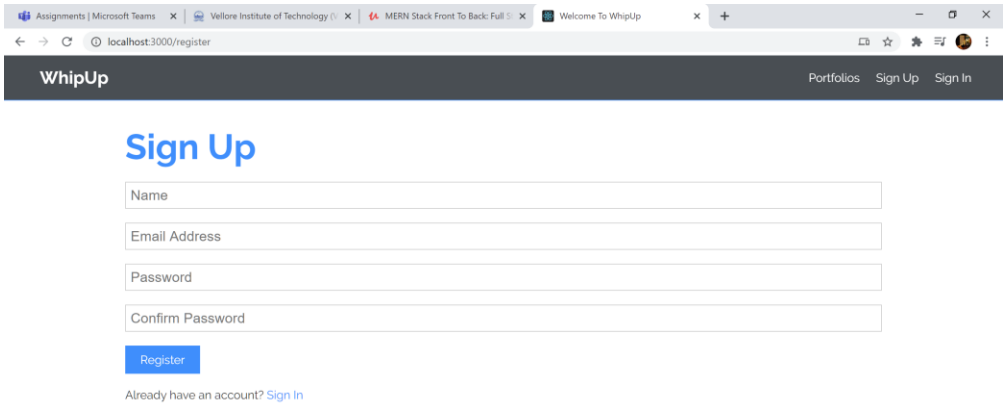
http://localhost:3000/



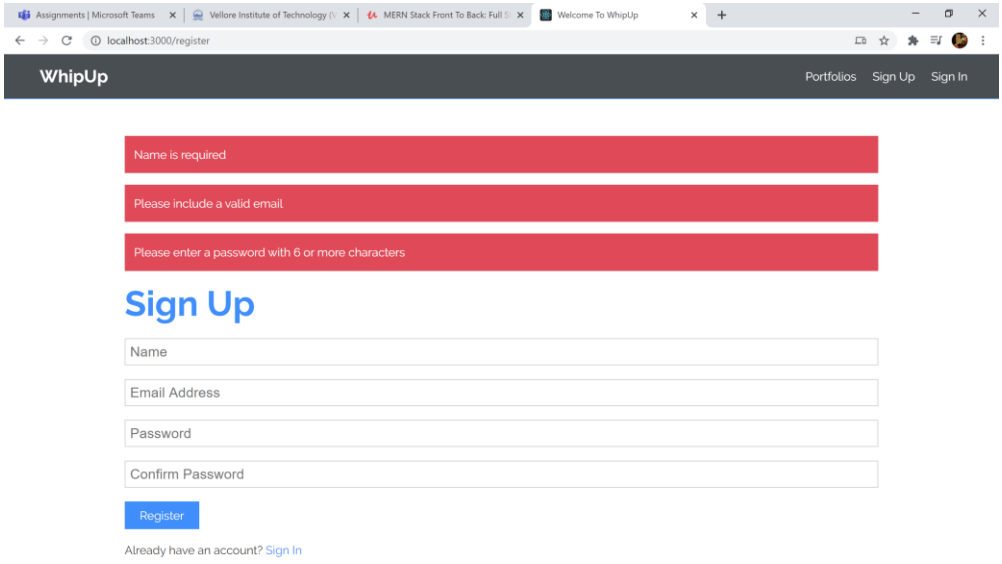
Register

http://localhost:3000/register

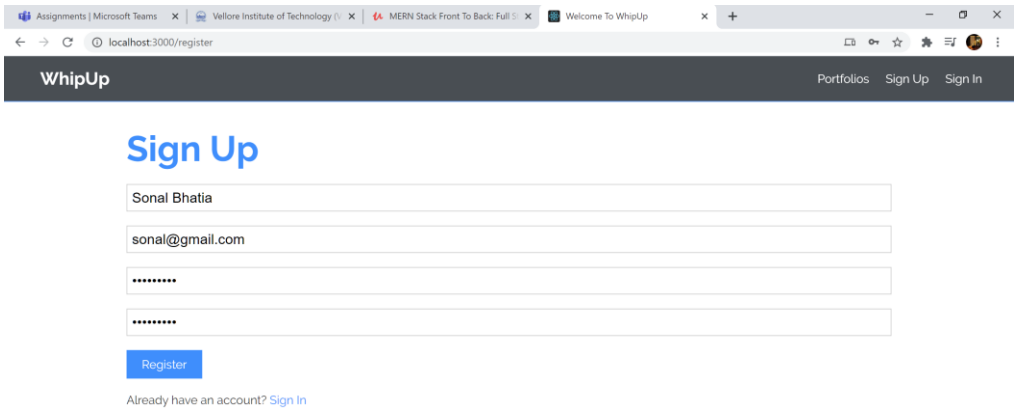
<Route exact path='/register' component={Register} />



Error handling



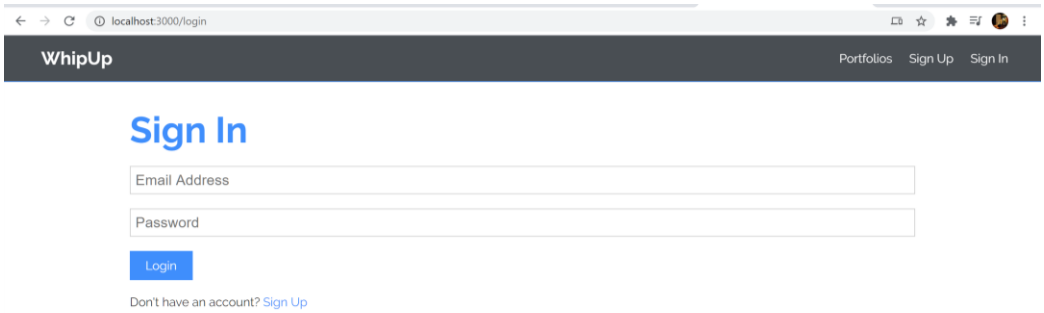
Successful Registration



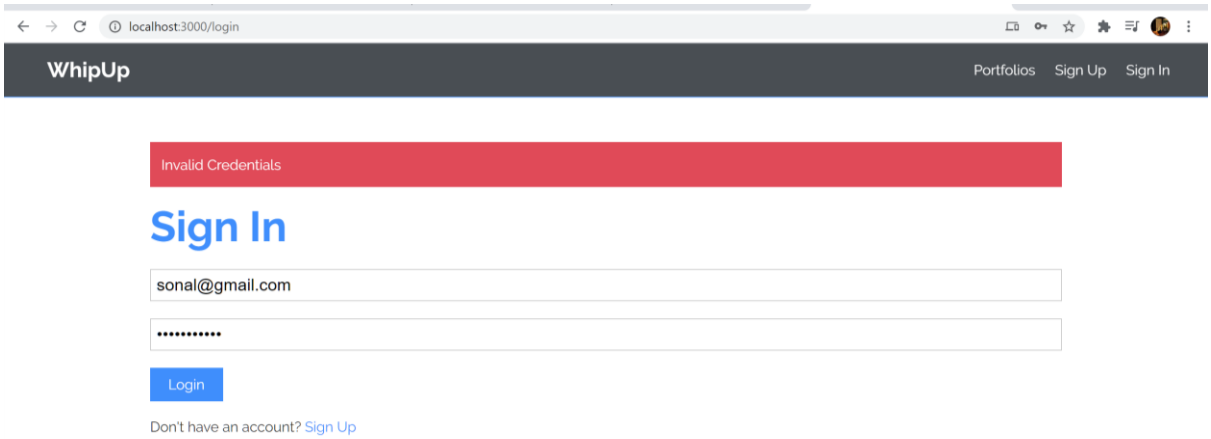
Login

http://localhost:3000/login

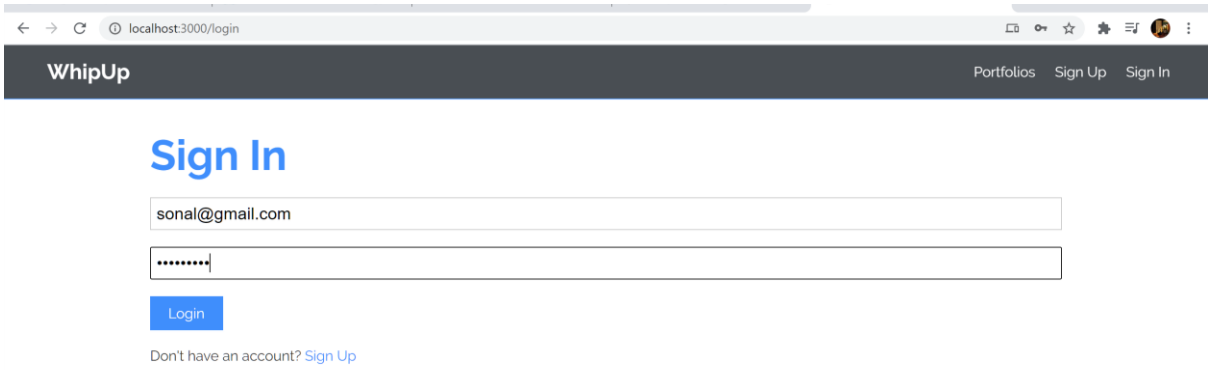
<Route exact path='/login' component={Login} />



Error handling



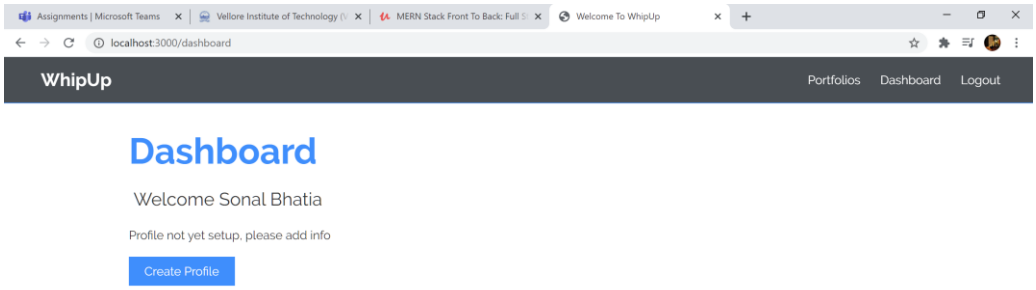
Successful Login



Dashboard

http://localhost:3000/dashboard

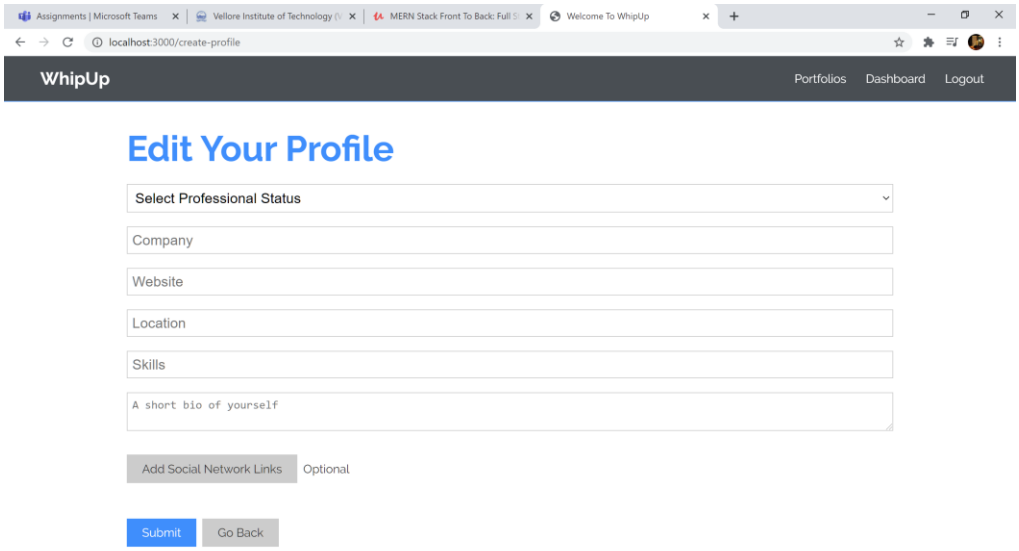
<PrivateRoute exact path='/dashboard' component={Dashboard} />



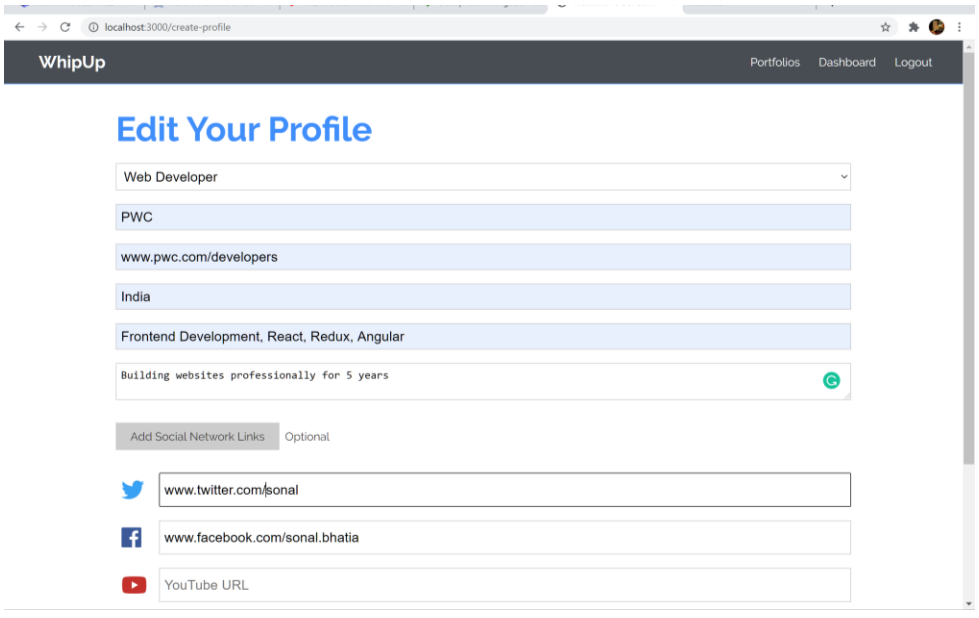
Create Profiles

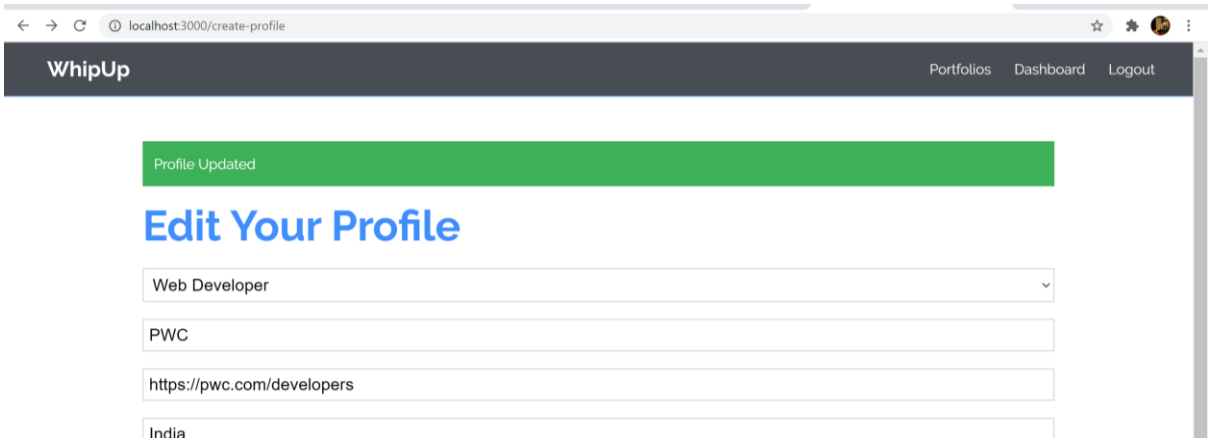
http://localhost:3000/create-profile

<PrivateRoute exact path='/create-profile' component={ProfileForm} />



Successful Profile Creation

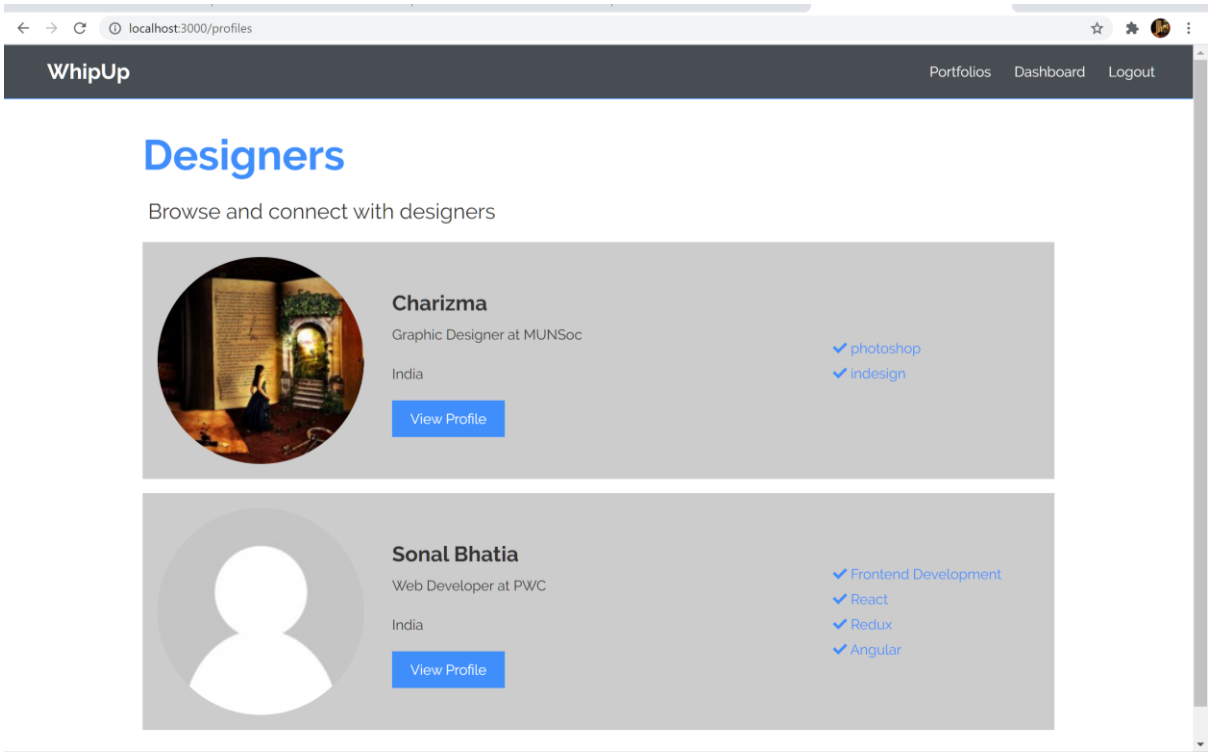




Profiles

http://localhost:3000/profiles

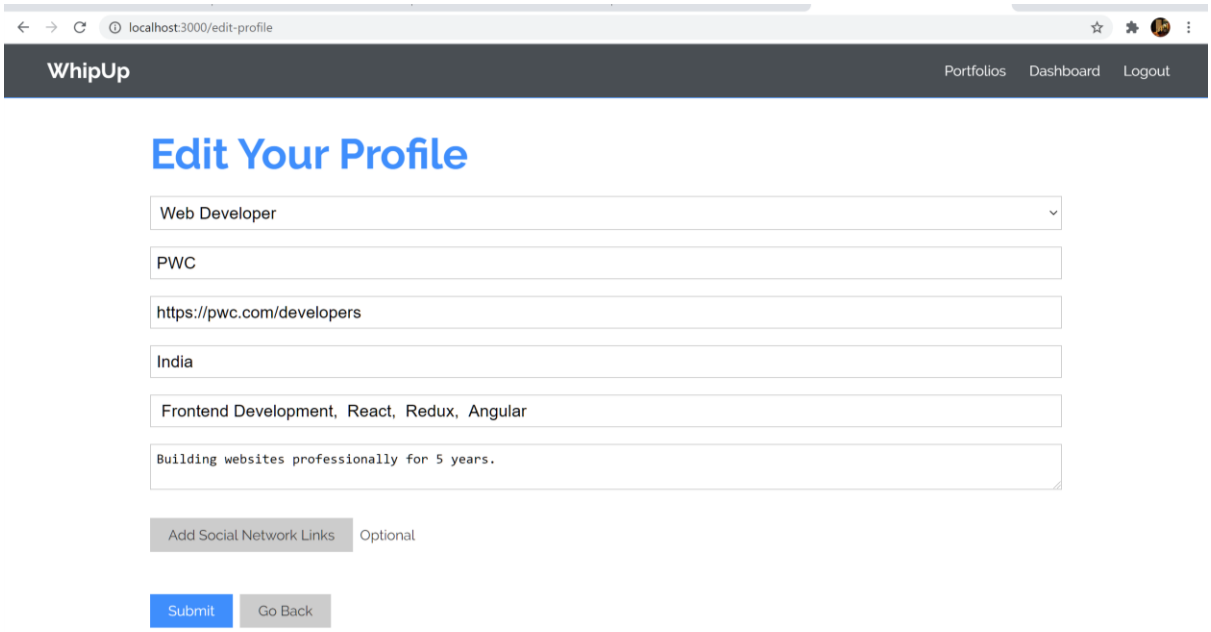
<Route exact path='/profiles' component={Profiles} />

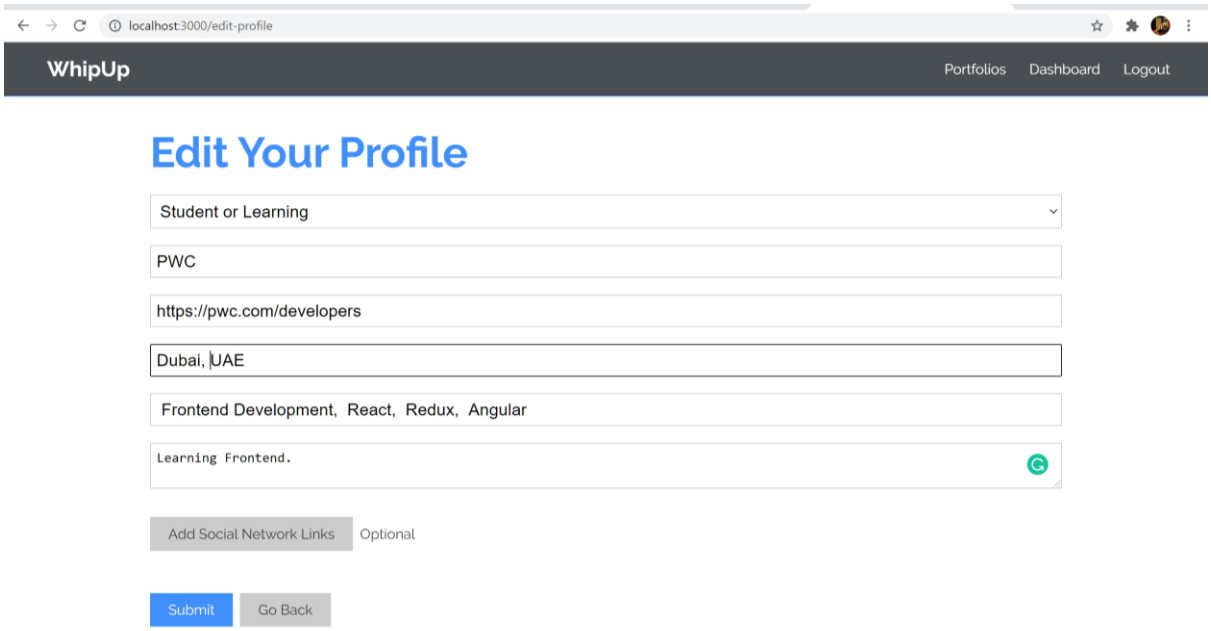


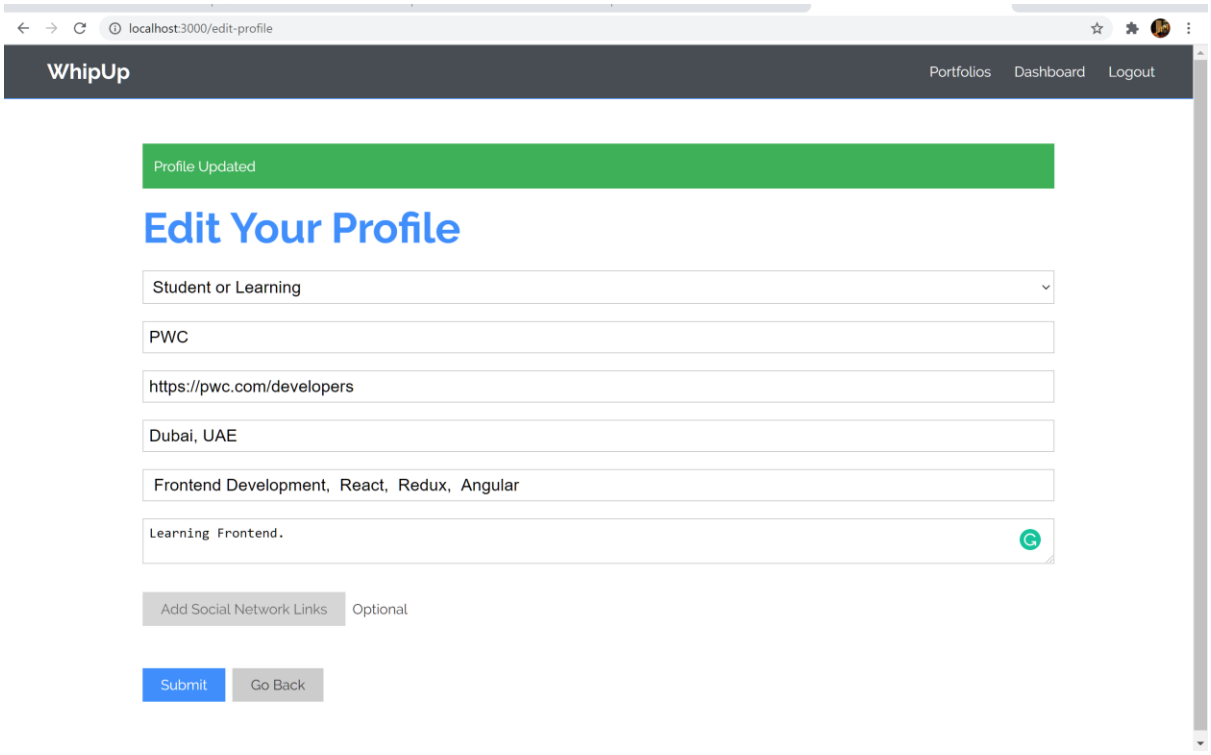
Edit Profiles

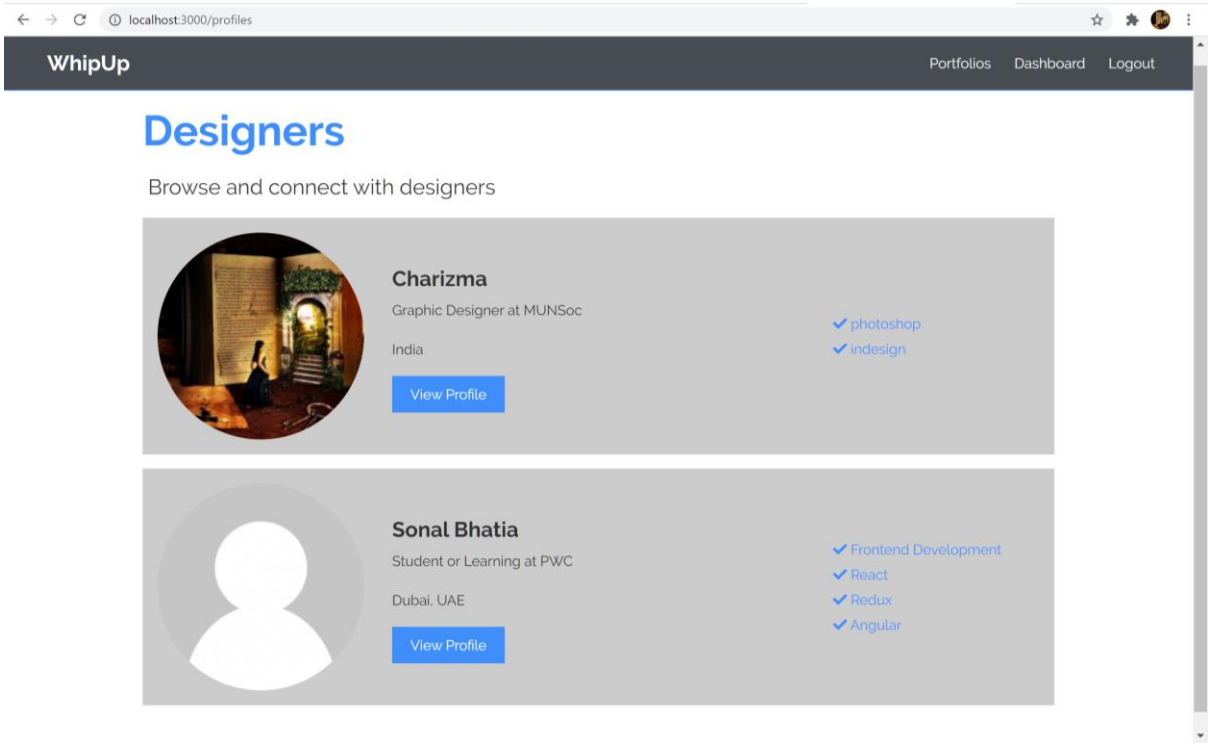
http://localhost:3000/edit-profile

<PrivateRoute exact path='/edit-profile' component={ProfileForm} />


localhost:3000/edit-profile
WhipUp
Portfolios
Dashboard
Logout
Edit Your Profile
Web Developer
PWC
https://pwc.com/developers
India
Frontend Development, React, Redux, Angular
Building websites professionally for 5 years.
Add Social Network Links
Optional
Submit
Go Back


localhost:3000/edit-profile
WhipUp
Portfolios
Dashboard
Logout
Edit Your Profile
Student or Learning
PWC
https://pwc.com/developers
Dubai, UAE
Frontend Development, React, Redux, Angular
Learning Frontend.
Add Social Network Links
Optional
Submit
Go Back

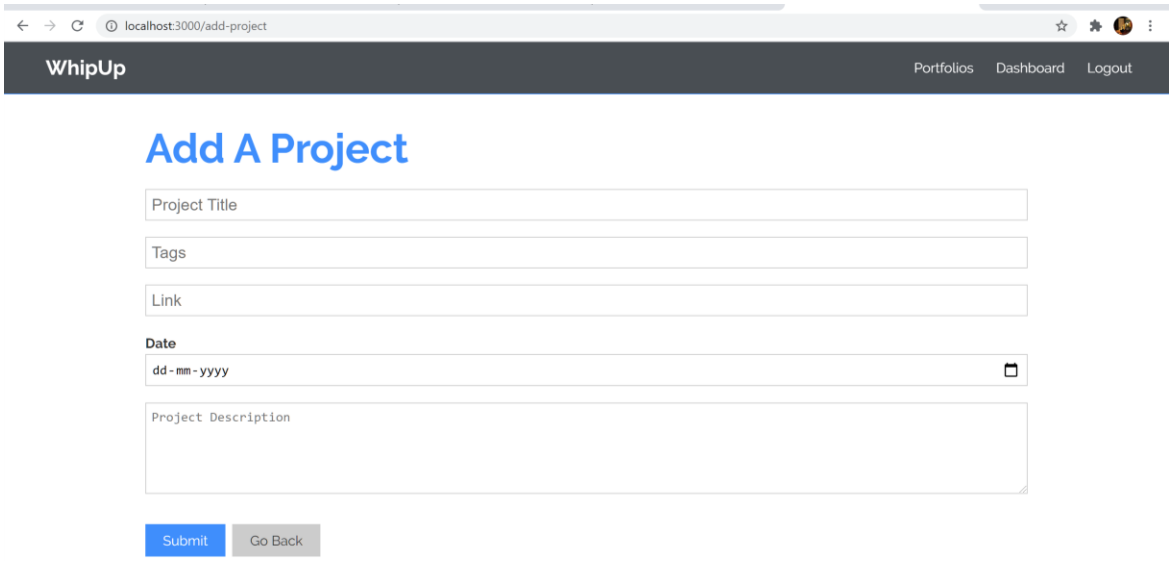


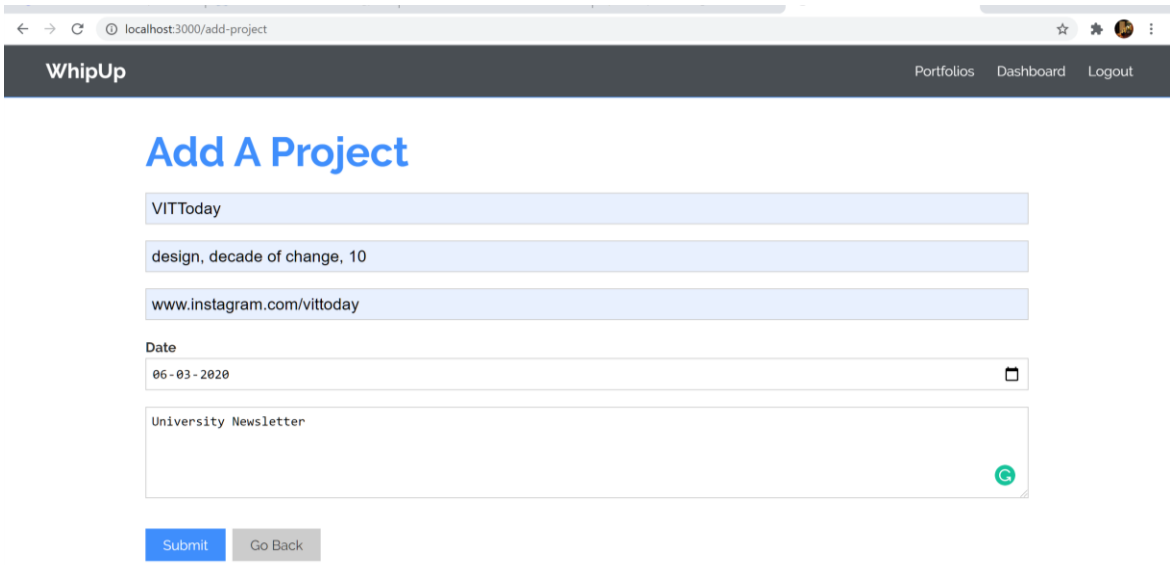


Add Project

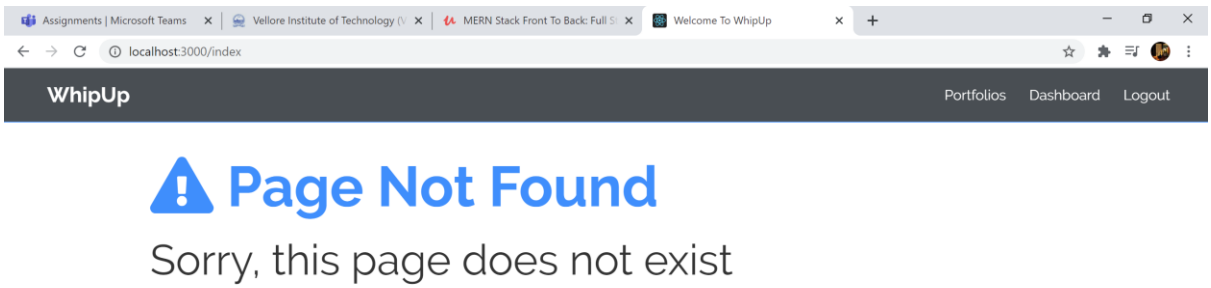
http://localhost:3000/add-project

<PrivateRoute exact path='/add-project' component={AddProject} />


localhost:3000/add-project
WhipUp
Portfolios
Dashboard
Logout
Add A Project
Project Title
Tags
Link
Date
dd-mm-yyyy
Project Description
Submit
Go Back

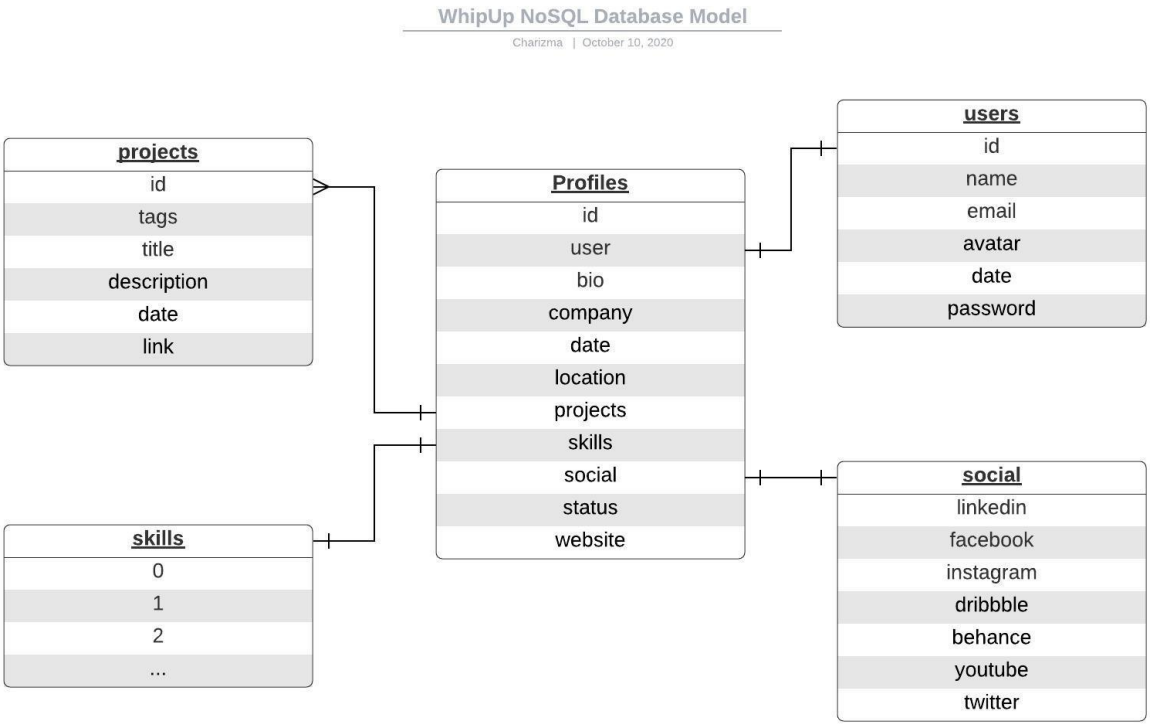

localhost:3000/add-project
WhipUp
Portfolios
Dashboard
Logout
Add A Project
VITToday
design, decade of change, 10
www.instagram.com/vittoday
Date
06-03-2020
University Newsletter
Submit
Go Back

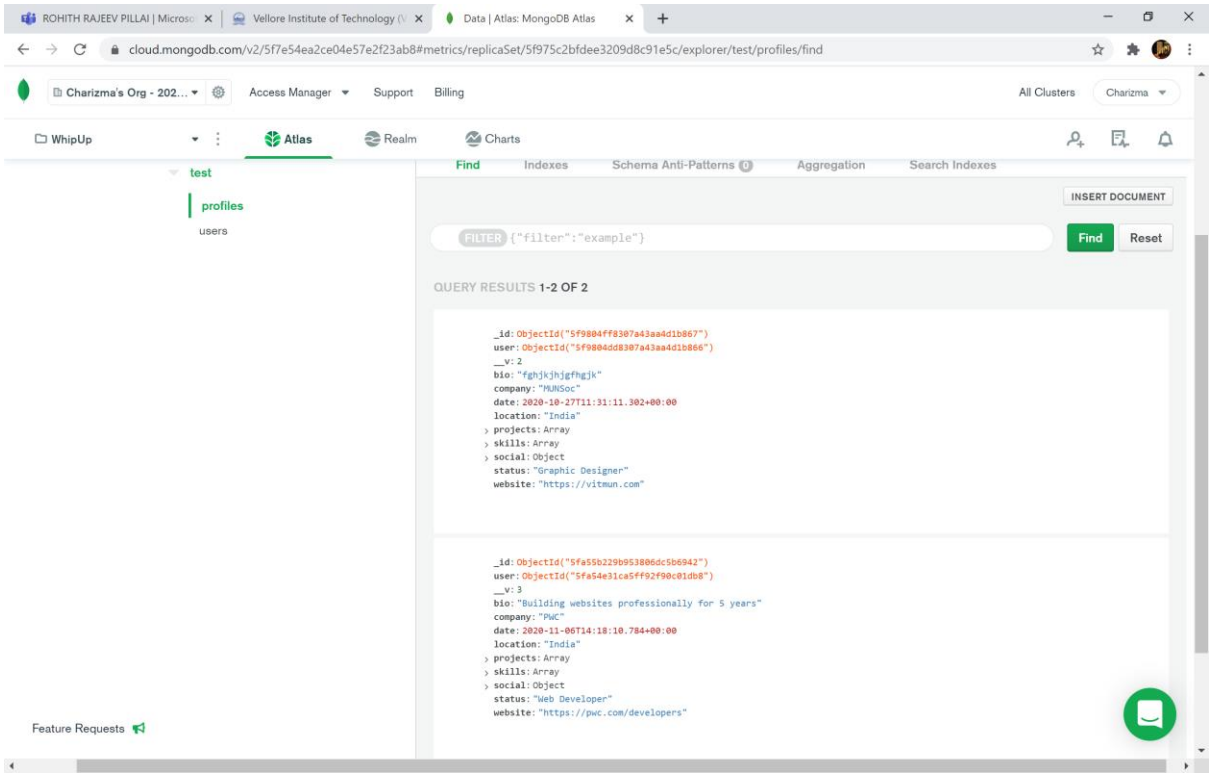
Unknown URL

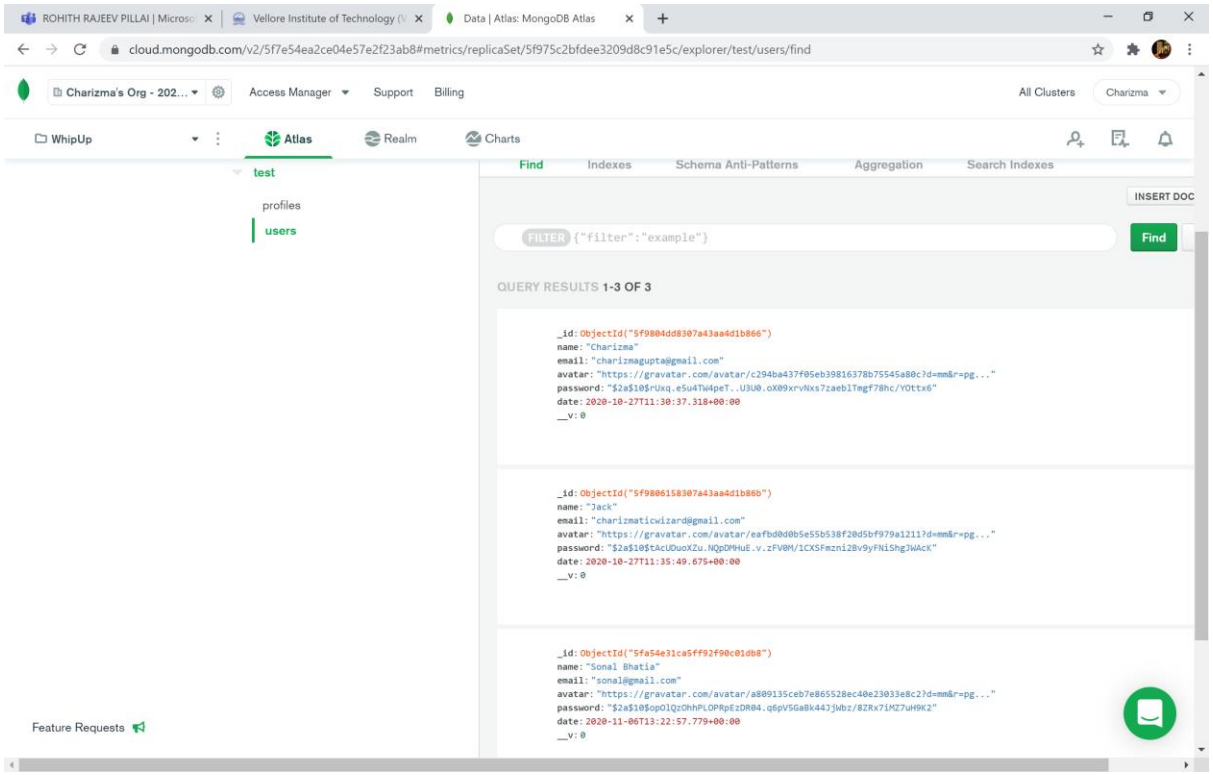

Assignments | Microsoft Teams
Vellore Institute of Technology
MERN Stack Front To Back: Full
Welcome To WhipUp
localhost:3000/index
WhipUp
Portfolios
Dashboard
Logout
Page Not Found
Sorry, this page does not exist

Website's DB

The NoSQL Database of MongoDB has been used.







Code

SERVER SIDE

server.js

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

const app = express();

//Connect Database
connectDB();

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

app.get('/', (req, res) => res.send('API Running'));

//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'));

const PORT = process.env.PORT || 5000;
```

```
app.listen(PORT, () => console.log(`Server started on Port ${PORT}`));
```

Profile.js

```
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,
  },
  projects: [
    {
      img: {
        data: Buffer,
        contentType: String,
      },
      title: {
        type: String,
        required: true,
      },
      tags: {
        type: [String],
        required: true,
      },
```

```
      date: {
        type: Date,
      },
      description: {
        type: String,
        required: true,
      },
      link: {
        type: String,
      },
    },
  ],
  social: {
    linkedin: {
      type: String,
    },
    facebook: {
      type: String,
    },
    instagram: {
      type: String,
    },
    dribbble: {
      type: String,
    },
    behance: {
      type: String,
    },
    youtube: {
      type: String,
    },
    twitter: {
      type: String,
    },
  },
  date: {
    type: Date,
    default: Date.now,
  },
});

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

User.js

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

```
const UserSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
  },
  email: {
    type: String,
    required: true,
    unique: true,
  },
  password: {
    type: String,
    required: true,
  },
  avatar: {
    type: String,
  },
  date: {
    type: Date,
    default: Date.now,
  },
});

module.exports = User = mongoose.model('user', UserSchema);
```

CLIENT SIDE

index.js

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

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

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';
import { LOGOUT } from './actions/types';
```

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

import './App.css';

const App = () => {
  useEffect(() => {
    // check for token in LS
    if (localStorage.token) {
      setAuthToken(localStorage.token);
    }
    store.dispatch(loadUser());

    // log user out from all tabs if they log out in one tab
    window.addEventListener('storage', () => {
      if (!localStorage.token) store.dispatch({ type: LOGOUT });
    });
  }, []);

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

export default App;
```

store.js

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

```
import setAuthToken from './utils/setAuthToken';

const initialState = {};

const middleware = [thunk];

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

let currentState = store.getState();

store.subscribe(() => {
  // keep track of the previous and current state to compare changes
  let previousState = currentState;
  currentState = store.getState();
  // if the token changes set the value in localStorage and axios headers
  if (previousState.auth.token !== currentState.auth.token) {
    const token = currentState.auth.token;
    setAuthToken(token);
  }
});

export default store;
```

App.css

```
/* Global Styles */
:root {
  --primary-color: #3f8efc;
  --dark-color: #343a40;
  --light-color: #ffffff;
  --danger-color: #dc3545;
  --success-color: #28a745;
}

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Raleway', sans-serif;
```

```
  font-size: 1rem;
  line-height: 1.6;
  background-color: #fff;
  color: #333;
}

a {
  color: var(--primary-color);
  text-decoration: none;
}

ul {
  list-style: none;
}

img {
  width: 100%;
}

/* Utilities */
.container {
  max-width: 1100px;
  margin: auto;
  overflow: hidden;
  padding: 0 2rem;
  margin-top: 6rem;
  margin-bottom: 3rem;
}

/* Text Styles*/
.x-large {
  font-size: 4rem;
  line-height: 1.2;
  margin-bottom: 1rem;
}

.large {
  font-size: 3rem;
  line-height: 1.2;
  margin-bottom: 1rem;
}

.lead {
  font-size: 1.5rem;
  margin-bottom: 1rem;
}

.text-center {
```

```
  text-align: center;
}

.text-primary {
  color: var(--primary-color);
}

.text-dark {
  color: var(--dark-color);
}

/* Padding */
.p {
  padding: 0.5rem;
}
.p-1 {
  padding: 1rem;
}
.p-2 {
  padding: 2rem;
}
.p-3 {
  padding: 3rem;
}
.py {
  padding: 0.5rem 0;
}
.py-1 {
  padding: 1rem 0;
}
.py-2 {
  padding: 2rem 0;
}
.py-3 {
  padding: 3rem 0;
}

/* Margin */
.m {
  margin: 0.5rem;
}
.m-1 {
  margin: 1rem;
}
.m-2 {
  margin: 2rem;
}
.m-3 {
```

```
  margin: 3rem;
}
.my {
  margin: 0.5rem 0;
}
.my-1 {
  margin: 1rem 0;
}
.my-2 {
  margin: 2rem 0;
}
.my-3 {
  margin: 3rem 0;
}

.btn {
  display: inline-block;
  background: var(--light-color);
  color: #333;
  padding: 0.4rem 1.3rem;
  font-size: 1rem;
  border: none;
  cursor: pointer;
  margin-right: 0.5rem;
  transition: opacity 0.2s ease-in;
  outline: none;
}

.badge {
  font-size: 0.8rem;
  padding: 0.1rem;
  text-align: center;
  margin: 0.3rem;
  background: var(--light-color);
  color: #333;
}

.alert {
  padding: 0.8rem;
  margin: 1rem 0;
  opacity: 0.9;
  background: var(--light-color);
  color: #333;
}

.btn-primary,
.bg-primary,
.badge-primary,
```

```css
.alert-primary {
  background: var(--primary-color);
  color: #fff;
}

.btn-light,
.bg-light,
.badge-light,
.alert-light {
  background: #ccc;
  color: #333;
}

.btn-dark,
.bg-dark,
.badge-dark,
.alert-dark {
  background: var(--dark-color);
  color: #fff;
}

.btn-danger,
.bg-danger,
.badge-danger,
.alert-danger {
  background: var(--danger-color);
  color: #fff;
}

.btn-success,
.bg-success,
.badge-success,
.alert-success {
  background: var(--success-color);
  color: #fff;
}

.btn-white,
.bg-white,
.badge-white,
.alert-white {
  background: #fff;
  color: #333;
  border: #ccc solid 1px;
}

.btn:hover {
  opacity: 0.8;
```

```
}

.bg-light,
.badge-light {
  border: #ccc solid 1px;
}

.round-img {
  border-radius: 50%;
}

.line {
  height: 1px;
  background: #ccc;
  margin: 1.5rem 0;
}

/* Overlay */
.dark-overlay {
  background-color: rgba(0, 0, 0, 0.7);
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
}

/* Forms */
.form .form-group {
  margin: 1.2rem 0;
}

.form .form-text {
  display: block;
  margin-top: 0.3rem;
  color: #888;
}

.form input[type='text'],
.form input[type='email'],
.form input[type='password'],
.form input[type='date'],
.form select,
.form textarea {
  display: block;
  width: 100%;
  padding: 0.4rem;
  font-size: 1.2rem;
```

```
  border: 1px solid #ccc;
}

.form input[type='submit'],
button {
  font: inherit;
}

.form .social-input {
  display: flex;
}

.form .social-input i {
  padding: 0.5rem;
  width: 4rem;
}

.form .social-input i.fa-twitter {
  color: #38a1f3;
}
.form .social-input i.fa-facebook {
  color: #3b5998;
}
.form .social-input i.fa-instagram {
  color: #3f729b;
}
.form .social-input i.fa-youtube {
  color: #c4302b;
}
.form .social-input i.fa-linkedin {
  color: #0077b5;
}

.table th,
.table td {
  padding: 1rem;
  text-align: left;
}

.table th {
  background: var(--light-color);
}

/* Navbar */
.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
```

```
  padding: 0.7rem 2rem;
  position: fixed;
  z-index: 1;
  width: 100%;
  top: 0;
  border-bottom: solid 1px var(--primary-color);
  opacity: 0.9;
}

.navbar ul {
  display: flex;
}

.navbar a {
  color: #fff;
  padding: 0.45rem;
  margin: 0 0.25rem;
}

.navbar a:hover {
  color: var(--primary-color);
}

.navbar .welcome span {
  margin-right: 0.6rem;
}

/* Landing Page */
.landing {
  position: relative;
  background: url('./img/showcase.png') no-repeat center center/cover;
  height: 100vh;
}

.landing-inner {
  color: #fff;
  height: 100%;
  width: 80%;
  margin: auto;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  text-align: center;
}

/* Profiles Page */
.profile {
```

```
  display: grid;
  grid-template-columns: 2fr 4fr 2fr;
  align-items: center;
  grid-gap: 2rem;
  padding: 1rem;
  line-height: 1.8;
  margin-bottom: 1rem;
}

/* Profile Page */
.profile-grid {
  display: grid;
  grid-template-areas:
    'top top'
    'about about'
    'exp edu'
    'github github';
  grid-gap: 1rem;
}

.profile-top {
  grid-area: top;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  text-align: center;
}

.profile-top img {
  width: 250px;
}

.profile-top .icons a {
  color: #fff;
  margin: 0 0.3rem;
}

.profile-top .icons a:hover {
  color: var(--dark-color);
}

.profile-about {
  grid-area: about;
  text-align: center;
}

.profile-about .skills {
```

```
  display: flex;
  align-items: center;
  justify-content: center;
  text-align: center;
}

.profile-exp {
  grid-area: exp;
}

.profile-edu {
  grid-area: edu;
}

.profile-exp h2,
.profile-edu h2 {
  margin-bottom: 1rem;
}

.profile-exp > div,
.profile-edu > div {
  margin-bottom: 1rem;
  padding-bottom: 1rem;
  border-bottom: #ccc 1px dotted;
}

.profile-exp > div:last-child,
.profile-edu > div:last-child {
  border: 0;
}

.profile-exp p,
.profile-edu p {
  margin: 0.5rem 0;
}

/* Mobile Styles */
@media (max-width: 700px) {
  .container {
    margin-top: 8rem;
  }

  .hide-sm {
    display: none;
  }

  /* Text Styles */
  .x-large {
```

```
  font-size: 3rem;
}

.large {
  font-size: 2rem;
}

.lead {
  font-size: 1rem;
}

/* Navbar */
.navbar {
  display: block;
  text-align: center;
}

.navbar ul {
  text-align: center;
  justify-content: center;
}

.navbar h1 {
  margin-bottom: 1rem;
}

.navbar .welcome {
  display: none;
}

/* Profiles Page */
.profile {
  grid-template-columns: 1fr;
  text-align: center;
}

.profile ul {
  display: none;
}

/* Profile Page */

.profile-top img,
.profile img {
  width: 200px;
  margin: auto;
}
```

```
  .profile-grid {
    grid-template-areas:
      'top'
      'about'
      'exp'
      'edu'
      'github';
  }

  .profile-about .skills {
    flex-direction: column;
  }

  .dash-buttons a {
    display: block;
    width: 100%;
    margin-bottom: 0.2rem;
  }
}
```

x----------x