**NAAN MUDHALVAN TASK-3**

**“ JOB SEARCH WEBSITE**

**USING REACT JS”**

**Initial Page Display Changes:**

Style.scss:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| $body-color: #949ca5;  $default-padding: 2rem;  \* {  margin: 0;  padding: 0;  box-sizing: border- box;  }  html {  font-size: 62.5%;  }  body {  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen,  Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;  font-size: 1.6rem;  background: #242582;  color: $body-color;  min-height: 100vh;  letter-spacing: 1px;  }  ul,  ol {  margin-left: 1rem;  }  .header {  padding: 2rem;  display: flex;  justify-content: center;  align-items: center;  margin-bottom: 2rem;  .title {  font-weight: 600;  font-size: 3rem;  color: #fff;  }  }  .search-section {  width: 70%;  margin: 20px auto;  input {  padding: $default-padding;  font-size: 1.5rem;  }  .btn-search {  font-size: 2.2rem;  letter-spacing: 1px;  width: 13rem;  line-height: 1.5;  color: #fff;  border: none;  }  }  .filters {  display: flex;  width: 70%;  .filter-btn {  margin: 2rem;  margin-left: 0;  padding: 1rem;  display: block;  border: 1px solid #888;  border-radius: 25px;  cursor: pointer;  }  .active {  background: #0069d9;  color: #fff;  }  .full-time-checkbox {  input {  margin-top: 0.5rem;  }  label {  margin-left: 1rem;  }  }  }  .search-results {  width: 90%;  margin: 0 auto;  .job-item {  display: flex;  padding: 1rem;  border: 1px solid #000;  border-radius: 0.5rem;  margin: 2rem;  cursor: pointer;  & > div {  margin-left: 2rem;  }  &:hover {  box-shadow: 0 0 10px rgba(255, 255, 255, 0.4);  }  .job-info {  .job-title {  margin-bottom: 2rem;  }  }  .company-logo {  img {  max-width: 10rem;  height: 10rem;  border-radius: 50%;  }  }  .job-info {  flex: 1;  }  .post-info {  display: flex;  justify-content: center;  align-items: center;  }  }  }  .load-more {  display: flex;  justify-content: center;  margin-bottom: 5rem;  button {  border: none;  background: #0e7bff;  color: #fff;  border-radius: 20px;  padding: 10px;  letter-spacing: 1px;  outline: none;  }  }  .job-details {  width: 90%;  margin: 50px auto;  .back-link {  a:link,  a:visited {  display: block;  margin-bottom: 1rem;  }  a:hover,  a:active {  color: #fff;  text-decoration: none;  }  }  .title {  font-weight: bold;  }  .main-section {  display: flex;  .left-section {  flex: 2;  margin: 0;  margin-right: 2rem;  }  .right-section {  flex: 1;  margin: 5rem 0 0 0;  & > div {  margin-bottom: 5rem;  }  h3 {  margin-bottom: 1rem;  }  .company-details {  .company-logo {  max-width: 10rem;  height: 10rem;  border-radius: 50%;  }  .company-name {  margin-top: 1rem;  }  .company-url {  word-break: break-all;  }  }  .how-to-apply {  a {  display: inline-block;  word-break: break-all;  }  }  }  }  }  .loading {  font-size: 2rem;  text-align: center;  font-size: 2rem;  color: #fff;  }  #loader {  width: 100%;  height: 100%;  position: fixed;  left: 0;  bottom: 0;  z-index: 99999;  background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5));  .message {  position: relative;  font-size: 20px;  top: 50%;  left: 50%;  transform: translateX(-10%);  color: #fff;  }  }  .loader-open {  overflow: hidden;  }  .disabled {  cursor: not-allowed;  opacity: 0.4;  }  .errorMsg {  max-width: 70%;  margin: 0 auto;  color: #f21e08;  padding: 1rem;  letter-spacing: 2px;  }  .hide {  display: none;  }  @media only screen and (max-width: 992px) {  .search-section {  width: 80%;  }  }  @media only screen and (max-width: 768px) {  .search-form {  .row {  flex-direction: column;  }  .col {  margin-bottom: 2rem;  }  }  .search-section .btn-search {  width: 100%;  }  .job-item {  flex-direction: column;  }  .search-results .job-item .post-info {  display: inline;  margin: 2rem 0 0 2rem;  }  .main-section {  flex-direction: column;  }  .job-info .job-title {  margin-top: 1rem;  }  }  jobs.js:  const jobsReducer = (state = [], action) => {  switch (action.type) {  case 'SET\_JOBS':  return action.jobs;  case 'LOAD\_MORE\_JOBS':  return [...state, ...action.jobs];  default:  return state;  }  };  export default jobsReducer;  errors.js:  const errorsReducer = (state = {}, action) => {  switch (action.type) {  case 'SET\_ERRORS':  return {  error: action.error  };  case 'RESET\_ERRORS':  return {};  default:  return state;  }  };  export default errorsReducer;  store.js:  import { createStore, combineReducers, applyMiddleware, compose } from 'redux';  import thunk from 'redux-thunk';  import jobsReducer from '../reducers/jobs';  import errorsReducer from '../reducers/errors';  const composeEnhancers = window.\_\_REDUX\_DEVTOOLS\_EXTENSION\_COMPOSE\_\_ || compose;  const store = createStore(  combineReducers({  jobs: jobsReducer,  errors: errorsReducer  }),  composeEnhancers(applyMiddleware(thunk))  );  console.log(store.getState());  export default store;  index.js:  import React from 'react';  import ReactDOM from 'react-dom';  import { Provider } from 'react-redux';  import store from './store/store';  import HomePage from './components/HomePage';  import 'bootstrap/dist/css/bootstrap.min.css';  import './css/styles.scss';  ReactDOM.render(  <Provider store={store}>  <HomePage />  </Provider>,  document.getElementById('root')  );  Homepage.js:  import React from 'react';  const HomePage = () => {  return <div className="container">Home Page</div>;  };  export default HomePage; Adding Basic Search UI: Header.js:  import React from 'react';  const Header = () => (  <header className="header">  <div className="title">Github Job Search</div>  </header>  );  export default Header;  Search.js:  import React, { useState } from 'react';  import { Form, Button, Row, Col } from 'react-bootstrap';  const Search = (props) => {  const [state, setState] = useState({  description: '',  location: '',  full\_time: false  });  const handleInputChange = (event) => {  const { name, value } = event.target;  if (name === 'full\_time') {  setState((prevState) => ({ ...state, [name]: !prevState.full\_time }));  } else {  setState({ ...state, [name]: value });  }  };  const handleSearch = (event) => {  event.preventDefault();  console.log(state);  };  return (  <div className="search-section">  <Form className="search-form" onSubmit={handleSearch}>  <Row>  <Col>  <Form.Group controlId="description">  <Form.Control  type="text"  name="description"  value={state.description || ''}  placeholder="Enter search term"  onChange={handleInputChange}  />  </Form.Group>  </Col>  <Col>  <Form.Group controlId="location">  <Form.Control  type="text"  name="location"  value={state.location || ''}  placeholder="Enter location"  onChange={handleInputChange}  />  </Form.Group>  </Col>  <Col>  <Button variant="primary" type="submit" className="btn-search">  Search  </Button>  </Col>  </Row>  <div className="filters">  <Form.Group controlId="full\_time">  <Form.Check  type="checkbox"  name="full\_time"  className="full-time-checkbox"  label="Full time only"  checked={state.full\_time}  onChange={handleInputChange}  />  </Form.Group>  </div>  </Form>  </div>  );  };  export default Search;  open HomePage.js and replace it with the following  content  import React from 'react';  import Header from './Header';  import Search from './Search';  const HomePage = () => {  return (  <div>  <Header />  <Search />  </div>  );  };  export default HomePage; Displaying List of Jobs on UI:errors.js:export const setErrors = (error) => ({type: 'SET\_ERRORS',error});export const resetErrors = () => ({type: 'RESET\_ERRORS'})constants.js:export const BASE\_API\_URL = 'http://localhost:5000';jobs.js:import axios from 'axios';import moment from 'moment';import { BASE\_API\_URL } from '../utils/constants';import { setErrors } from './errors';export const initiateGetJobs = (data) => {return async (dispatch) => {try {let { description, full\_time, location, page } = data;description = description ? encodeURIComponent(description) : '';location = location ? encodeURIComponent(location) : '';full\_time = full\_time ? '&full\_time=true' : '';if (page) {page = parseInt(page);page = isNaN(page) ? '' : `&page=${page}`;}const jobs = await axios.get(`${BASE\_API\_URL}/jobs?description=${description}&location=${location}${full\_time}${page}`);const sortedJobs = jobs.data.sort((a, b) =>moment(new Date(b.created\_at)) - moment(new Date(a.created\_at)));return dispatch(setJobs(sortedJobs));} catch (error) {error.response && dispatch(setErrors(error.response.data));}};};export const setJobs = (jobs) => ({type: 'SET\_JOBS',jobs});export const setLoadMoreJobs = (jobs) => ({type: 'LOAD\_MORE\_JOBS',jobs});Server.js:const path = require('path');const axios = require('axios');const cors = require('cors');const express = require('express');const app = express();const PORT = process.env.PORT || 5000;const buildPath = path.join(\_\_dirname, '..', 'build');app.use(express.static(buildPath));app.use(cors());app.get('/jobs', async (req, res) => {try {let { description = '', full\_time, location = '', page = 1 } = req.query;description = description ? encodeURIComponent(description) : '';location = location ? encodeURIComponent(location) : '';full\_time = full\_time === 'true' ? '&full\_time=true' : '';if (page) {page = parseInt(page);page = isNaN(page) ? '' : `&page=${page}`;}const query = `https://jobs.github.com/positions.json?description=${description}&location=${location}${full\_time}${page}`;const result = await axios.get(query);res.send(result.data);} catch (error) {res.status(400).send('Error while getting list of jobs.Try again later.');}});app.listen(PORT, () => {console.log(`server started on port ${PORT}`);});Jobitem.js:import React from 'react';import moment from 'moment';const JobItem = (props) => {const {id,type,created\_at,company,location,title,company\_logo,index} = props;return (<div className="job-item" index={index + 1}><div className="company-logo"><img src={company\_logo} alt={company} width="100" height="100" /></div><div className="job-info"><div className="job-title">{title}</div><div className="job-location">{location} | {type}</div><div className="company-name">{company}</div></div><div className="post-info"><div className="post-time">Posted {moment(new Date(created\_at)).fromNow()}</div></div></div>);};export default JobItem;Results.js:import React from 'react';import JobItem from './JobItem';const Results = ({ results }) => {return (<div className="search-results">{results.map((job, index) => (<JobItem key={job.id} {...job} index={index} />))}</div>);};export default Results;package.json:{"name": "server","version": "1.0.0","main": "index.js","license": "MIT","scripts": {"start": "nodemon server.js"},"dependencies": {"axios": "0.19.2","cors": "2.8.5","express": "4.17.1","nodemon": "^2.0.4",}}Displaying Job Details Page:Jobdetails.js:import React from 'react';const JobDetails = ({ details, onResetPage }) => {const {type,title,description,location,company,company\_url,company\_logo,how\_to\_apply} = details;return (<div className="job-details"><div className="back-link"><a href="/#" onClick={onResetPage}>&lt;&lt; Back to results</a></div><div>{type} / {location}</div><div className="main-section"><div className="left-section"><div className="title">{title}</div><hr /><div className="job-description">{description}</div></div><div className="right-section"><div className="company-details"><h3>About company</h3><img src={company\_logo} alt={company} className="company-logo" /><div className="company-name">{company}</div><a className="company-url" href={company\_url}>{company\_url}</a></div><div className="how-to-apply"><h3>How to apply</h3><div>{how\_to\_apply}</div></div></div></div></div>);};export default JobDetails;Image.js:import React from 'react';import { useState } from 'react';import loading from '../images/loading.png';/\* https://via.placeholder.com/100x100?text=Loading \*/const Image = ({ src, alt, ...props }) => {const [isVisible, setIsVisible] = useState(false);const changeVisibility = () => {setIsVisible(true);};return (<React.Fragment><imgsrc={loading}alt={alt}width="100"height="100"style={{ display: isVisible ? 'none' : 'inline' }}{...props}/><imgsrc={src}alt={alt}width="100"height="100"onLoad={changeVisibility}style={{ display: isVisible ? 'inline' : 'none' }}{...props}/></React.Fragment>);};export default Image;output: |