

GANPAT UNIVERSITY
U.V. PATEL COLLEGE OF ENGINEERING

B.Tech 5th Semester CE/IT

2CEIT5PE4: Software Packages

Practical: 8

Working with React.js

Make environment setup for react application.

To create react application

npx create-react-app pr8

To run react application

npm start

Create single page application using react.js which is able to perform following functionality:

Que:1 Add title and logo of website.

App.js

```
import logo from './uvpce_logo.jpg'
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img className="logo" src={logo} alt="logo"/>
        <p className="title">UVPCE</p>
      </header>
    </div>
  );
}

export default App;
```

App.css

```
.App-logo {
  height: 40vmin;
  pointer-events: none;
}

.App-header {
  display: flex;
  flex-direction: row;
  align-items: center;
  font-size: calc(10px + 2vmin);
  color: white;
  justify-content:center;
}

.title{
  color: black;
  margin-left: 10px;
}

.logo{
  height: 50px;
  width: 100px;
}
```

pr8/public/index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!-- manifest.json provides metadata used when your web app is
```

```
    installed on a user's mobile device or desktop. See
    https://developers.google.com/web/fundamentals/web-app-manifest/
-->
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<!--
    Notice the use of %PUBLIC_URL% in the tags above.
    It will be replaced with the URL of the `public` folder
    during the build.
    Only files inside the `public` folder can be referenced from the HTML.
    Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
    work correctly both with client-side routing and a non-root public URL.
    Learn how to configure a non-root public URL by running `npm run
    build`.
-->
<title>React App</title>
</head>
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
  -->
</body>
</html>
```

pr8/src/index.js

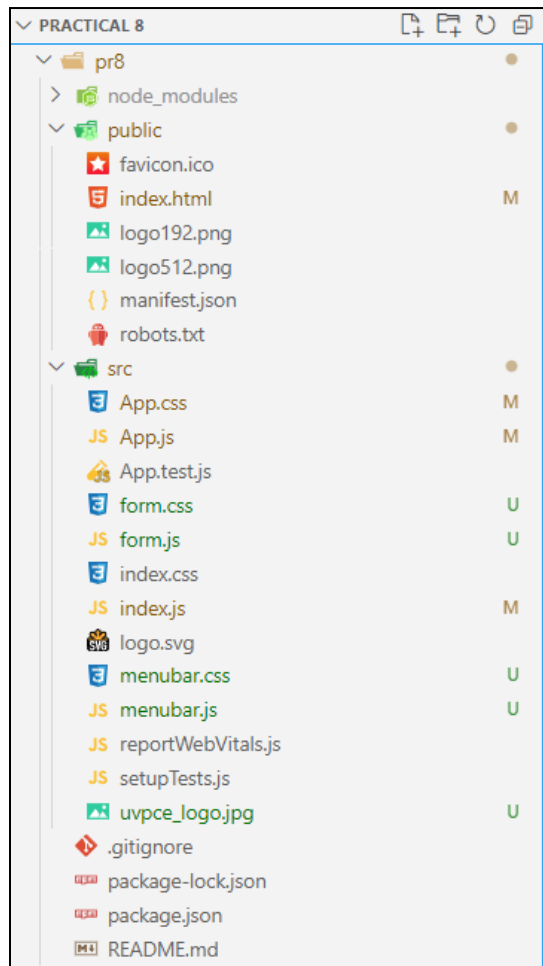
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

pr8/src/index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
  monospace;
}
```

Resources: /pr8/src/uvpce_logo.jpg**Output:**

Compiled successfully!

You can now view **pr8** in the browser.

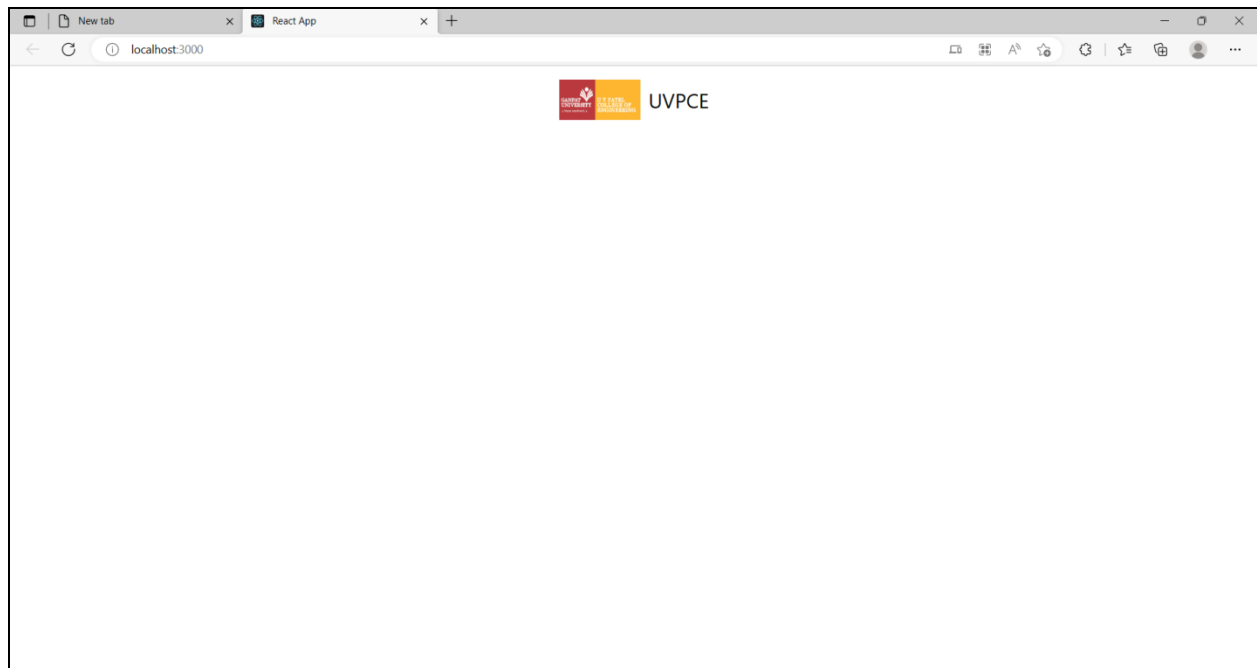
Local: <http://localhost:3000>

On Your Network: <http://192.168.43.58:3000>

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**





Que:2 Create horizontal or vertical menu using CSS.

pr8/src/menubar.js

```
import './menubar.css';

function menubar() {
  return (
    <div>
      <ul>
        <li className='active'><a>Home</a></li>
        <li><a>Contact</a></li>
        <li><a>About</a></li>
      </ul>
    </div>
  );
}

export default menubar;
```

pr8/src/menubar.css

```
ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
  overflow: hidden;  
  background-color: #333;  
}  
  
li {  
  float: left;  
  border-right: 1px solid #bbb;  
}  
  
li:last-child {  
  border-right: none;  
}  
  
li a {  
  display: block;  
  color: white;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}  
  
li a:hover:not(.active) {  
  background-color: #111;  
}  
  
.active {  
  background-color: #04AA6D;  
}
```

pr8/public/index.html

```
<div id="navbar"></div>
```

Add above line of code in **pr8/public/index.html** after **<div id="root"></div>** this tag.

pr8/src/index.js

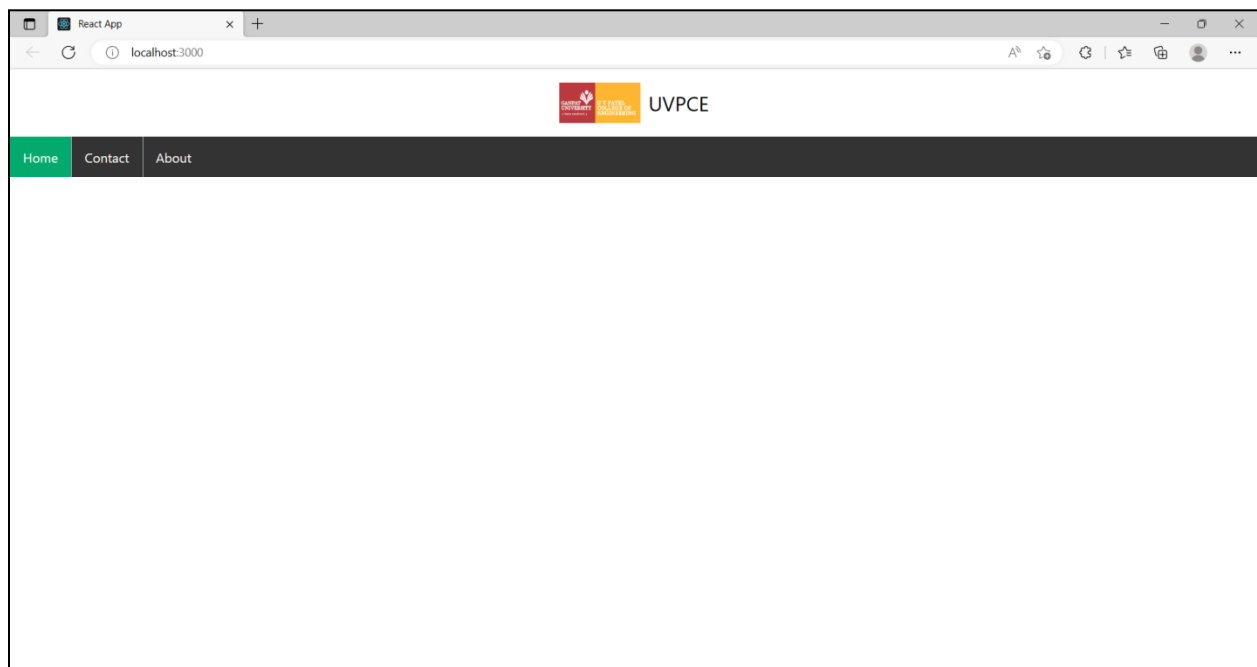
Import **pr8/src/menubar.js** file in **pr8/src/index.js** file like this.

```
import MenuBar from './menubar';
```

Add below line of code in **pr8/src/index.js** file after importing the **pr8/src/menubar.js** file

```
const navbar = ReactDOM.createRoot(document.getElementById('navbar'));
navbar.render(
  <React.StrictMode>
    <MenuBar />
  </React.StrictMode>
);
```

Output:



Que:3 Make registration form having minimum 4-5 different elements and submit value of elements.

Que:4 Show working of events: onClick(), onChange() and onSubmit().

pr8/src/form.js

```
import React from 'react'
import './form.css';

class form extends React.Component {
  constructor(props) {
    super(props);
    this.state = {name: "",email:"",mobile:"",
gender:"", sem:"",lan1:"",lan2:"",lan3:""};
  }
  onHandleClick=()=>{
    const { name, email, mobile, gender,
sem, lan1, lan2, lan3 } = this.state;
    alert("OnClick\n"+name+"\n"+email+"\n"+
mobile+"\n"+gender+"\n"+sem+"\n"+lan1+" "+lan2+" "+lan3);
  }
  onSubmitHandleClick=(e)=>{
    e.preventDefault();
    const { name, email, mobile, gender,
sem, lan1, lan2, lan3 } = this.state;
    alert("OnSubmit\n"+name+"\n"+email+"\n"+
mobile+"\n"+gender+"\n"+sem+"\n"+lan1+" "+lan2+" "+lan3);
  }
  render() {
    return (
      <div className='mainLayout'>
        <form className='form' method="post"
onSubmit={this.onSubmitHandleClick.bind(this)}>
          <table className='table'>
            <tr>
              <h1 className='formTitle'>Registration Form</h1>
            </tr>
          </table>
        </div>
      </div>
    )
  }
}
```

```
<td><p>Name: </p></td>
<td>
  <input
    className='input'
    type="text"
    placeholder='Your Name'
    value={this.state.name}
    name="name"
    onChange={ (e) => this.setState(
      { name: e.target.value } ) } />
</td>
</tr>
<tr>
<td><p>Email: </p></td>
<td>
  <input
    className='input'
    type="text"
    placeholder='Your Email'
    value={this.state.email}
    name="email"
    onChange={ (e) => this.setState(
      { email: e.target.value } ) }
  />
</td>
</tr>
<tr>
<td><p>Mobile Number: </p></td>
<td>
  <input
    className='input'
    type="number"
    placeholder='Your Contact'
    value={this.state.mobile}
    name="mobile"
    onChange={ (e) => this.setState(
      { mobile: e.target.value } ) }
  />
</td>
```

```

</tr>
<tr>
  <td><p>Gender: </p></td>
  <td onChange={ (e) => this.setState(
    { gender: e.target.value } )}>
    <input className='input' type="radio"
      name="gen" value="Male"/>Male
    <input className='input' type="radio"
      name="gen" value="Female"/>Female
  </td>
</tr>
<tr>
  <td><p>Semester: </p></td>
  <td>
    <select name="sem" className='inputSem'
      onChange={ (e) => this.setState(
        { sem: e.target.value } )}>
      <option value="null">Select</option>
      <option value="Sem 1">1</option>
      <option value="Sem 2">2</option>
      <option value="Sem 3">3</option>
      <option value="Sem 4">4</option>
      <option value="Sem 5">5</option>
      <option value="Sem 6">6</option>
      <option value="Sem 7">7</option>
      <option value="Sem 8">8</option>
    </select>
  </td>
</tr>
<tr>
  <td><p>Language: </p></td>
  <td>
    <input name="lan1" className='input'
      type="checkbox" value="C" onChange={
        (e)=> {let isChecked =
          e.target.checked;if (isChecked) {
            this.setState({lan1: e.target.value})}
            else{this.setState({lan1:""})} } }>C
  </td>
</tr>

```

```

        <input name="lan2" className='input'
        type="checkbox" value="Java" onChange={
        (e)=> {let isChecked =
        e.target.checked;if(isChecked)
        {this.setState({lan2: e.target.value})}
        else{this.setState({lan2:""})}}}}/>Java

        <input name="lan3" className='input'
        type="checkbox" value="Python" onChange={
        (e)=> {let isChecked =
        e.target.checked;if(isChecked) {
        this.setState({lan3: e.target.value})}
        else{this.setState({lan3:""})}}}}/>Python
    </td>
</tr>
<tr>
    <td></td>
    <td><input className="btnSubmit"
    type="submit" value="Submit"/></td>
</tr>
</table>
</table>
</form>
<button className="btnOnClick" onClick={
this.onHandleClick.bind(this)}>onClick</button>
<form className='form'>
    <table className='table'>
        <tr>
            <h1 className='formTitle'>On Change Event</h1>
        </tr>
        <table>
            <tr>
                <td><p>Name: </p></td>
                <td><p>{this.state.name}</p></td>
            </tr>
            <tr>
                <td><p>Email: </p></td>
                <td><p>{this.state.email}</p></td>
            </tr>
        </table>
    </table>

```

```
        <tr>
            <td><p>Mobile Number: </p></td>
            <td><p>{this.state.mobile}</p></td>
        </tr>
        <tr>
            <td><p>Gender: </p></td>
            <td><p>{this.state.gender}</p></td>
        </tr>
        <tr>
            <td><p>Semester: </p></td>
            <td><p>{this.state.sem}</p></td>
        </tr>
        <tr>
            <td><p>Language: </p></td>
            <td><p>{this.state.lan1} {this.state.lan2}
                {this.state.lan3}</p></td>
        </tr>
    </table>
</table>
</form>
</div>
    );
}
}

export default form;
```

pr8/src/form.css

```
.form{
  display: flex;
  flex-direction: row;
  margin-left: 20px;
  margin-top: 10px;
  justify-content: center;
}
.input{
  border: none;
  border-bottom-style: solid;
  border-bottom-width: 3px;
  border-color: #f00;
  margin-left: 10px;
  padding: 5px;
}
.inputSem{
  width: 100px;
  border: none;
  border-bottom-style: solid;
  border-bottom-width: 3px;
  border-color: #f00;
  margin-left: 10px;
  padding: 5px;
}
.btnSubmit{
  margin-left: 10px;
  border: none;
  border-bottom-style: solid;
  border-bottom-width: 3px;
  padding: 10px;
}
.btnOnClick{
  margin-top: 10px;
  margin-left: 10px;
  border: none;
  border-bottom-style: solid;
  border-bottom-width: 3px;
  padding: 10px;
}
```

```
}  
.formTitle{  
  text-align: center;  
}  
.table{  
  border: 1px solid;  
  padding: 30px;  
}  
.mainLayout{  
  display: flex;  
  flex-direction: row;  
  justify-content: center;  
}
```

pr8/public/index.html

```
<div id="form"></div>
```

Add above line of code in **pr8/public/index.html** after **<div id="navbar"></div>** this tag.

pr8/src/index.js

Import **pr8/src/form.js** file in **pr8/src/index.js** file like this.

```
import Form from './form';
```

Add below line of code in **pr8/src/index.js** file after importing the **pr8/src/form.js** file

```
const form = ReactDOM.createRoot(document.getElementById('form'));  
form.render(  
  <React.StrictMode>  
    <Form />  
  </React.StrictMode>  
) ;
```

Output:

The screenshot shows a web browser window with the URL `localhost:3000`. The page has a header with the UVPC logo and the text "UVPC". Below the header is a navigation bar with links for "Home", "Contact", and "About". The main content area is divided into two columns. The left column contains a "Registration Form" with the following fields: "Name:" (text input), "Email:" (text input), "Mobile Number:" (text input), "Gender:" (radio buttons for "Male" and "Female"), "Semester:" (dropdown menu), and "Language:" (checkboxes for "C", "Java", and "Python"). A "Submit" button is at the bottom of the form. The right column contains an "On Change Event" section, which is currently empty. A vertical grey bar with the text "onClick" is positioned between the two columns.

The screenshot shows the same web browser window after the registration form has been submitted. The "Registration Form" on the left now contains the following data: "Name:" (ABC), "Email:" (abc@gmail.com), "Mobile Number:" (1234567890), "Gender:" (radio button for "Male" is selected), "Semester:" (5), and "Language:" (checkboxes for "C" and "Java" are selected). The "On Change Event" section on the right now displays the submitted data: "Name: ABC", "Email: abc@gmail.com", "Mobile Number: 1234567890", "Gender: Male", "Semester: Sem 5", and "Language: C Java". The vertical grey bar with the text "onClick" remains between the two columns.

