LAB ASSIGNMENT NO – 6b

Aim: Write a program to implement the concept of forms and events. Create a React JS registration Form consisting of textbox, textarea, selection input, check box, radio button, submit button and reset button handling onSubmit, onClick and keyDown events.

Theory:

React Forms:

React provides a way to create and manage forms in web applications. Forms are used to collect user input and submit it to the server or perform actions based on the input. React forms can be controlled or uncontrolled, depending on how you manage the form data.

Controlled Forms: In controlled forms, React components (usually input elements) are controlled by React state. This means that the form elements' values are stored in React state, and any changes to the form elements trigger state updates, causing the UI to re-render with updated values. Controlled forms are typically implemented using the useState hook or class component state.

Uncontrolled Forms: Uncontrolled forms are less common in React but can be used when you want to work with form elements directly without managing their values through React state. In uncontrolled forms, the form elements hold their values in the DOM, and you can access their values using refs or other DOM manipulation techniques.

React Events:

React events allow you to respond to user interactions or actions within a React application. React uses a synthetic event system that normalizes browser-specific events, making it consistent and cross-browser compatible.

Event Handling: In React, you can attach event handlers to elements using JSX syntax. Common events include onClick, onChange, onSubmit, onKeyDown, and many others. When an event occurs, the associated event handler function is executed.

Event Object: React event handlers receive a synthetic event object as an argument. This object contains information about the event, such as the target element, event type, and event properties.

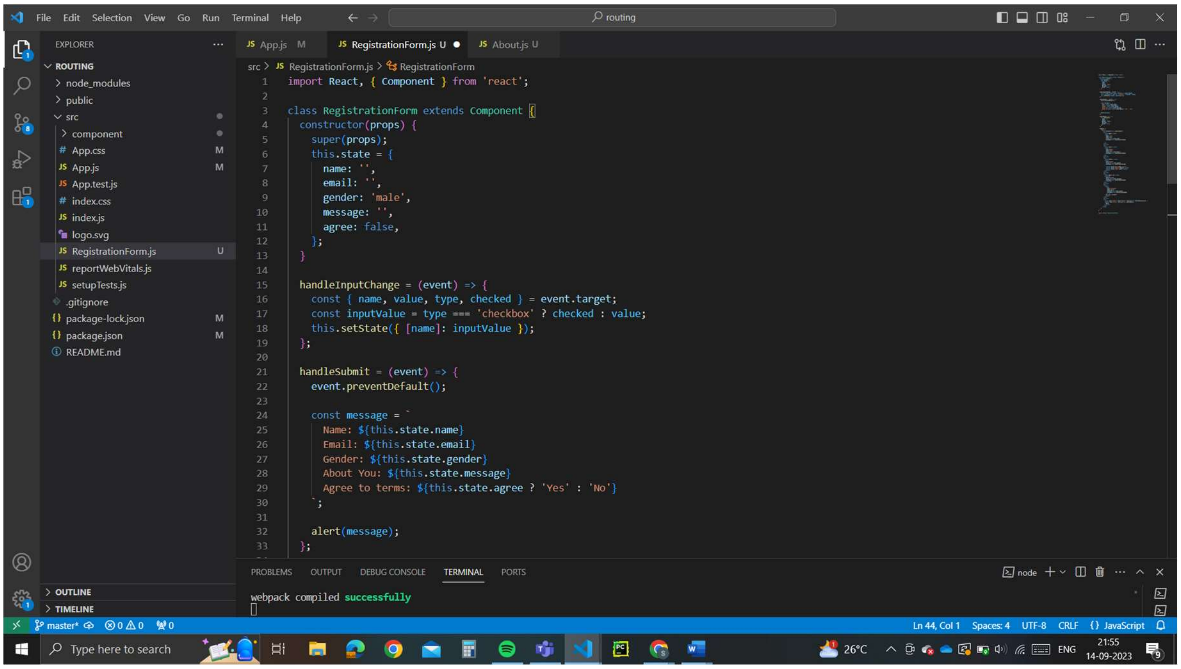
Preventing Default Behavior: You can use the event.preventDefault() method to prevent the default behavior of an event, such as preventing a form submission or link navigation.

State Updates: Event handlers are often used to update the component's state, which triggers a re-render of the component. For example, you can use the onChange event to update the state when a user types in an input field.

Conclusion:

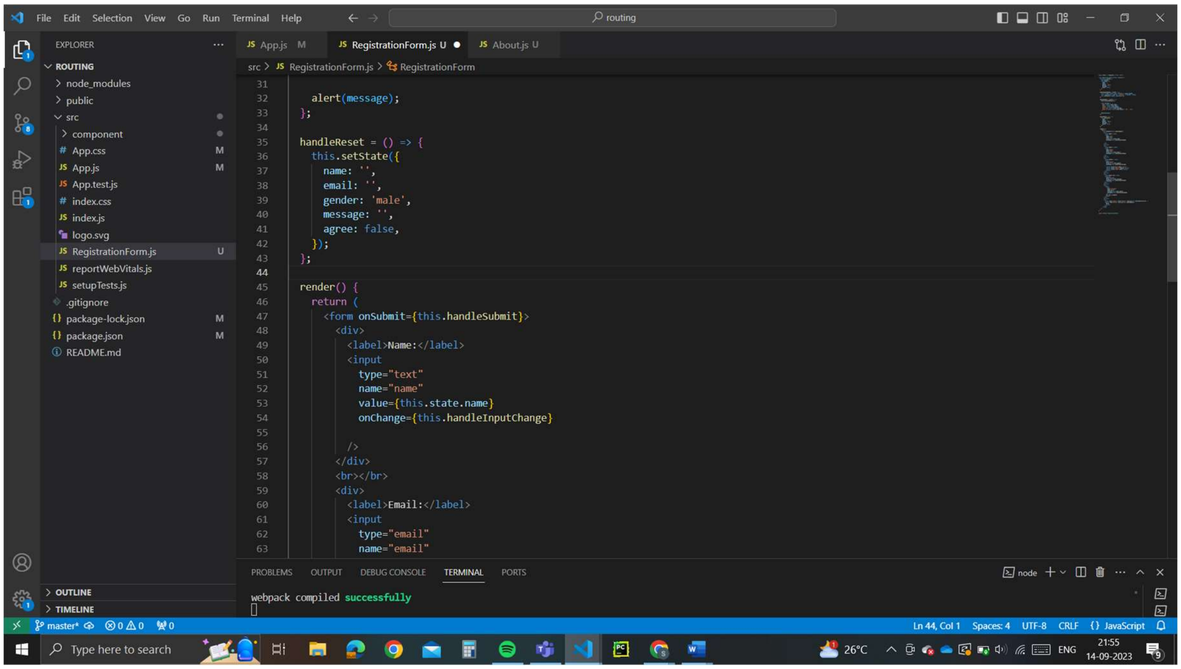
We learnt that React forms and events are fundamental concepts for building interactive and user-friendly web applications. React provides tools and patterns for handling user input, managing form data, and responding to user interactions through event handling. Whether building simple forms or complex user interfaces, understanding and using React forms and events effectively is crucial for creating dynamic and responsive applications.

Code:



```
import React, { Component } from 'react';

class RegistrationForm extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: '',
      email: '',
      gender: 'male',
      message: '',
      agree: false,
    };
  }

  handleInputChange = (event) => {
    const { name, value, type, checked } = event.target;
    const inputValue = type === 'checkbox' ? checked : value;
    this.setState({ [name]: inputValue });
  };

  handleSubmit = (event) => {
    event.preventDefault();

    const message = `
      Name: ${this.state.name}
      Email: ${this.state.email}
      Gender: ${this.state.gender}
      About You: ${this.state.message}
      Agree to terms: ${this.state.agree ? 'Yes' : 'No'}
    `;

    alert(message);
  };
```

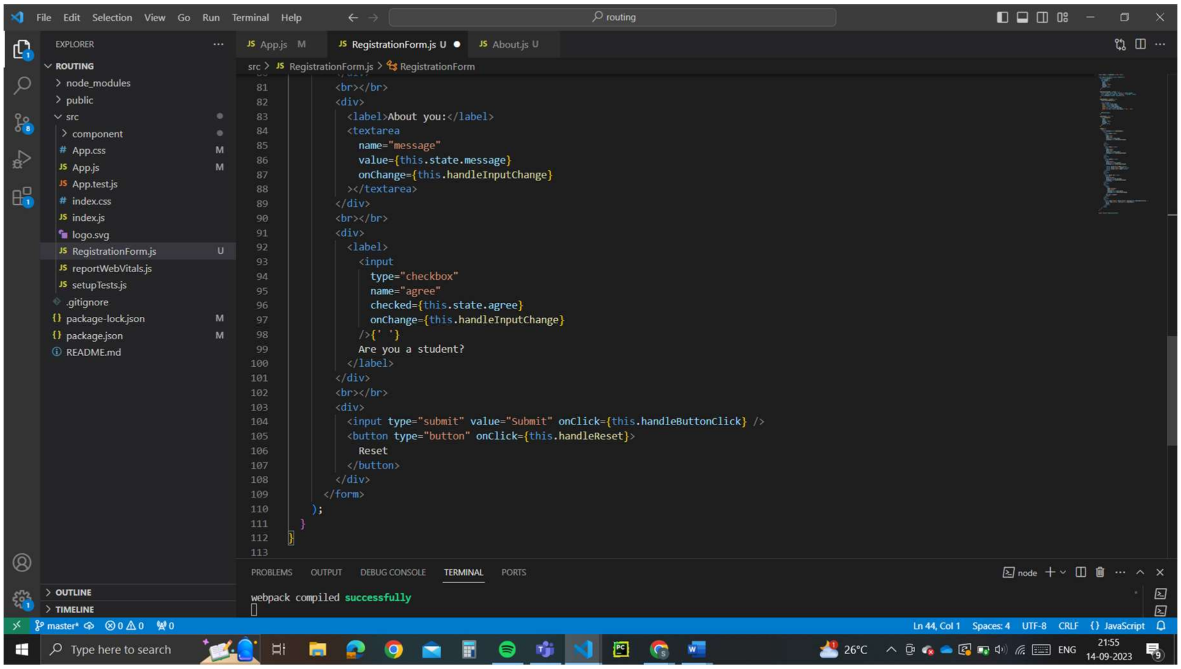
```
webpack compiled successfully
```



```

    alert(message);
  };

  handleReset = () => {
    this.setState({
      name: '',
      email: '',
      gender: 'male',
      message: '',
      agree: false,
    });
  };

  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <div>
          <label>Name:</label>
          <input
            type="text"
            name="name"
            value={this.state.name}
            onChange={this.handleInputChange}

          />
        </div>
        <br></br>
        <div>
          <label>Email:</label>
          <input
            type="email"
            name="email"
```

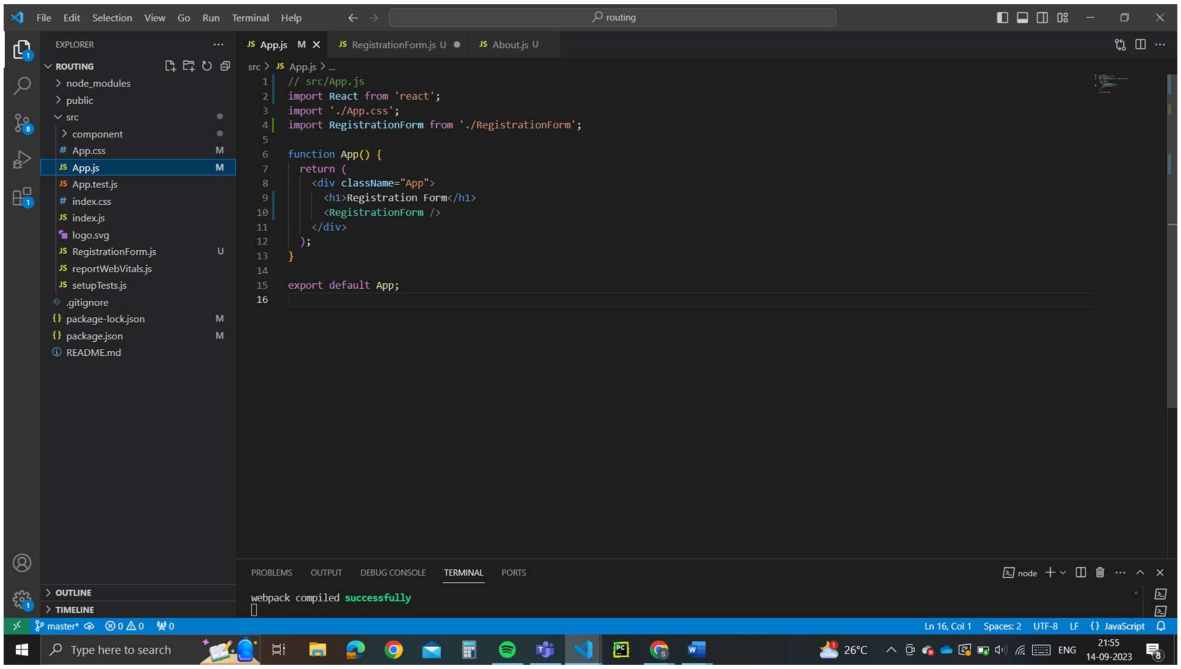
```
webpack compiled successfully
```

Shreya Bagade/ SEM V/ TE.IT/ T11/ Roll No-09/ Assignment no-6b



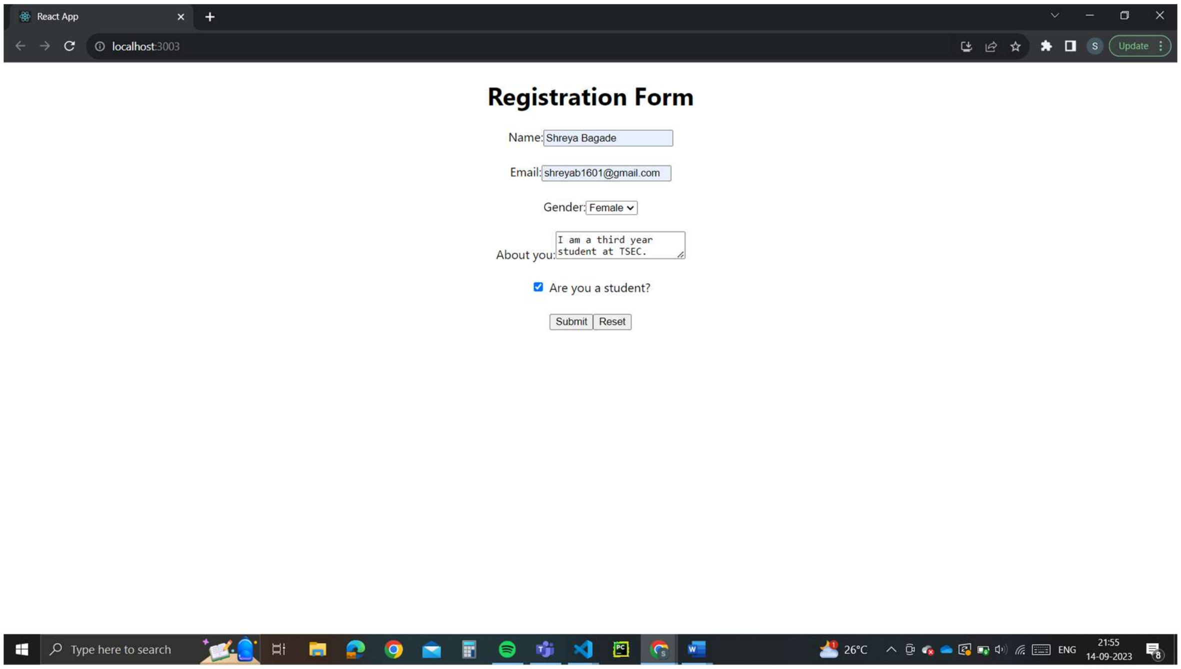
```
        <br></br>
        <div>
          <label>About you:</label>
          <textarea
            name="message"
            value={this.state.message}
            onChange={this.handleInputChange}
          ></textarea>
        </div>
        <br></br>
        <div>
          <label>
            <input
              type="checkbox"
              name="agree"
              checked={this.state.agree}
              onChange={this.handleInputChange}
            />{' '}
            Are you a student?
          </label>
        </div>
        <br></br>
        <div>
          <input type="submit" value="Submit" onClick={this.handleButtonClick} />
          <button type="button" onClick={this.handleReset}>
            Reset
          </button>
        </div>
      </form>
    );
  }
}
```

```
webpack compiled successfully
```



```
// src/App.js
import React from 'react';
import './App.css';
import RegistrationForm from './RegistrationForm';

function App() {
  return (
    <div className="App">
      <h1>Registration Form</h1>
      <RegistrationForm />
    </div>
  );
}

export default App;
```

```
webpack compiled successfully
```



Registration Form

Name: Shreya Bagade

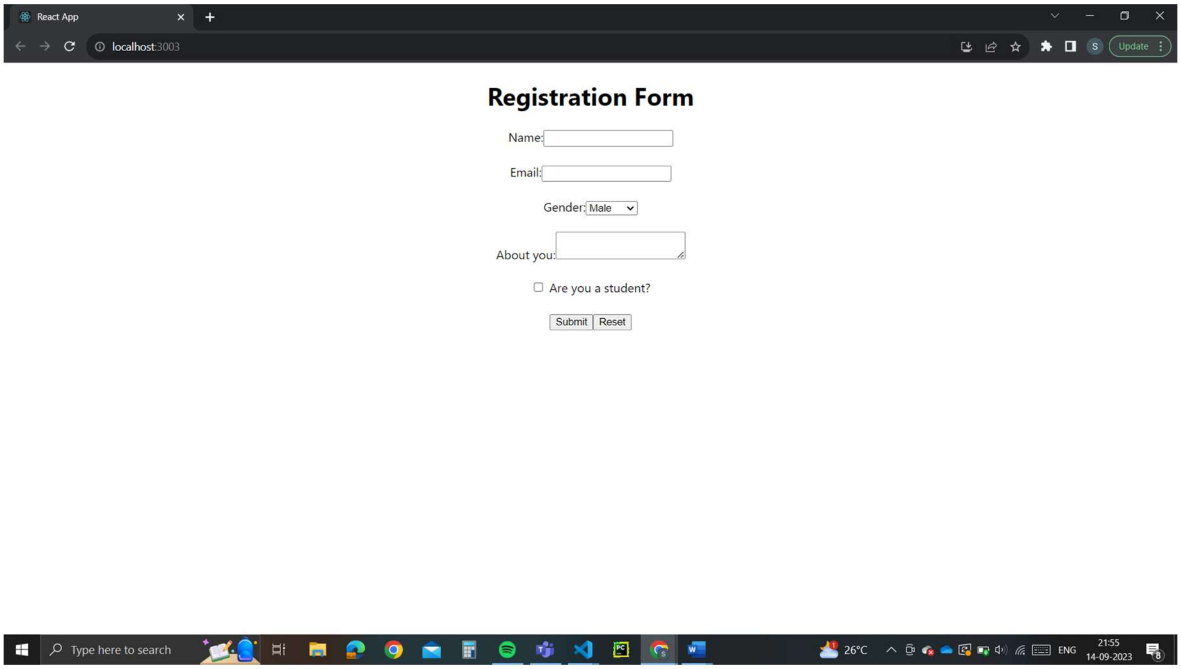
Email: shreyab1601@gmail.com

Gender: Female

About you: I am a third year student at TSEC.

☑ Are you a student?

Submit Reset



Registration Form

Name:

Email:

Gender: Male

About you:

□ Are you a student?

Submit Reset

__**Lab Outcome :**__ **LO-5** : Construct front end applications using React.