Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2025-26

Class:	**TE**	**Semester:**	**V**
Course Code:	**CSL501**	**Course Name:**	**Web Computing and Network Lab**

Name of Student:	**Mitesh Doulat Pawar**
Roll No. :	**73**
Experiment No.:	**05**
Title of the Experiment:	**Perform ReactJS project initialization with some major functionalities.**
Date of Performance:	**05/08/2025**
Date of Submission:	**12/08/2025**

Evaluation

Performance Indicator	**Max. Marks**	**Marks Obtained**
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	**Exceed Expectations (EE)**	**Meet Expectations (ME)**	**Below Expectations**
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Ms Kshitija Gharat

Signature :

Date:

Experiment No. 5

Aim: React: Installation and Configuration, JSX, Components, Props, State, Forms, Events, Routers, Refs, Keys.

Objective:

1) To produce the most effective possible rendering performance.

2) Rather than being engaged on the whole web app, React JS allows a developer to break down the complex UI into simpler components.

Theory:

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front end library responsible only for the view layer of the application. It was created by Jordan Walke, who was a software engineer at Facebook. It was initially developed and maintained by Facebook and was later used in its products like WhatsApp & Instagram. Facebook developed ReactJS in 2011 in its newsfeed section, but it was released to the public in the month of May 2013.

Today, most of the websites are built using MVC (model view controller) architecture. In MVC architecture, React is the 'V' which stands for view, whereas the architecture is provided by the Redux or Flux.

A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks. ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time

Installation Reactjs on Windows:

Step 1: Install Node.js installer for windows. Once downloaded open NodeJS without disturbing other settings, click on the Next button until it's completely installed.

Step 2: Open command prompt to check whether it is completely installed or not type the command –>

```
node -v
```

If the installation went well it will give you the version you have installed

Step 3: Now in the terminal run the below command:

```
npm install -g create-react-app
```

It will globally install react app for you. To check everything went well run the command

```
create-react-app --version
```

If everything went well it will give you the installed version of react app

Step 4:Now Create a new folder where you want to make your react app using the below command:

```
mkdir newfolder.
```

Move inside the same folder using the below command:

```
cd newfolder (your folder name)
```

Step 5: Now inside this folder run the command –>

```
create-react-app reactfirst YOUR_APP_NAME
```

Step 6: Now open the IDE of your choice for eg. Visual studio code and open the folder where you have installed the react app newolder (in the above example) inside the folder you will see your app's name reactapp (In our example). Use the terminal and move inside your app name folder.Use command cd reactapp (your app name)

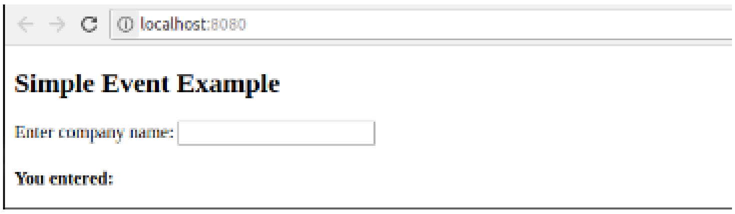
Step 7: To start your app run the below command :

```
npm start
```

React Components, State , Props and Events

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

class App extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      companyName: ''

    };

  }
  changeText(event) {
    this.setState({
      companyName: event.target.value
    });
  }
  render() {
    return (
      <div>
        <h2>Simple Event Example</h2>
        <label htmlFor="name">Enter company name: </label>
        <input type="text" id="companyName" onChange={this.changeText.bind(this)}/>
        <h4>You entered: { this.state.companyName }</h4>
      </div>
    );
  }
}
export default App;
```

Output



Simple Event Example

Enter company name:

You entered:

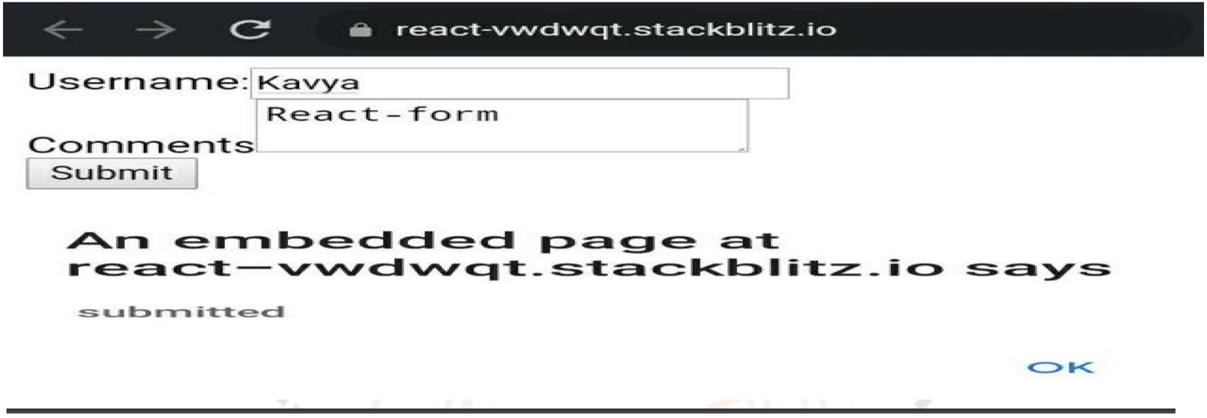
React Form and Router

```
import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';
import reportWebVitals from './reportWebVitals';
ReactDOM.render(
  <App/>,
  document.getElementById('root')
);
import React from "react";
import {BrowserRouter as Router,Switch,Route,Link} from "react-router-dom"
function App() {
  return (
            <Router>
            <div>
            <ul>
            <li>
            <Link to="/">Home</Link>
            </li>
            <li>
            <Link to="/about">About</Link>
            </li>
            <li>
            <Link to="/dashboard">Dashboard</Link>
            </li>
```

```
        </ul>
        <hr/>
        <Switch>
        <Route exact path="/">
        <Home />
        </Route>
        <Route path="/about">
        <About />
        </Route>
        <Route path="/dashboard">
        <Dashboard />
        </Route>
        </Switch>
        </div>
        </Router>
  );
}
function Home() {
  return (
        <div>
        <h2>Home</h2>
```

```
            </div>
  );
}
function About() {
  return (
            <div>
            <h2>About</h2>
            </div>
  );
}
function Dashboard() {
  return (
            <div>
      <h2>Dashboard</h2>
            </div>
  );
}
export default App
```

Output



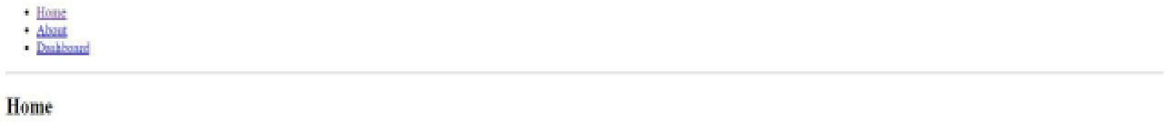
react-vwdwqt.stackblitz.io

Username: Kavya

Comments React-form

An embedded page at react-vwdwqt.stackblitz.io says

submitted



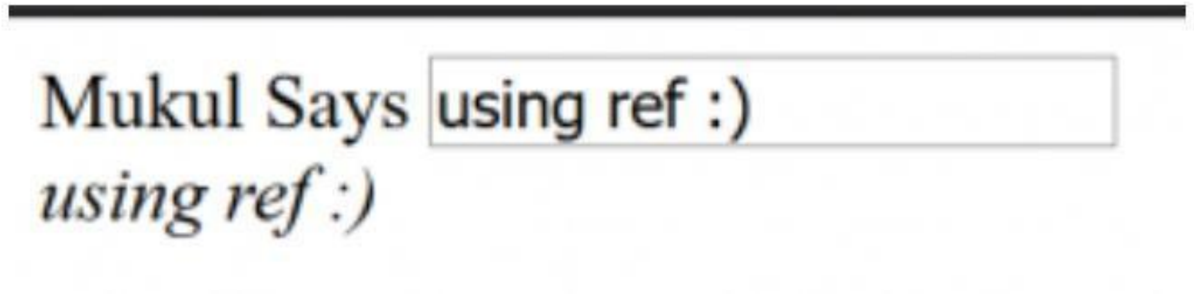
- Home
- About
- Dashboard

Home

React Refs

```
// using refs
class App extends React.Component {
 constructor(){
  super();
  this.state = { sayings: ""};
 }
 update(e){
  this.setState({ sayings: this.refs.anything.value});
 }
 render(){
  return (
   <div>
    Mukul Says <input type="text" ref="anything"
      onChange = {this.update.bind(this)}/>
   <br/>
   <em>{this.state.sayings}</em>
  </div>
  );
 }
}
 ReactDOM.render(< App />, document.getElementById('root'));
```

output

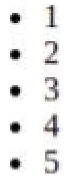


```
Mukul Says using ref :)
using ref :)
```

React keys

```
import React from "react";
import ReactDOM from "react-dom";
// Component to be extracted
function MenuItems(props) {
    const item = props.item;
     return <li>{item}</li>;
}
// Component that will return an
// unordered list
function Navmenu(props) {
    const list = props.menuitems;
    const updatedList = list.map((listItems) => {

        return <MenuItems key={listItems.toString()} item={listItems} />;

    });

    return <ul>{updatedList}</ul>;

}

const menuItems = [1, 2, 3, 4, 5];

ReactDOM.render(

    <Navmenu menuitems={menuItems} />,

    document.getElementById("root")

);
```

Output



```
• 1
• 2
• 3
• 4
• 5
```

Conclusion: ReactJS serves as a powerful and efficient JavaScript library for building dynamic and performant user interfaces, particularly focused on the view layer of an application within an MVC-like architecture. Its core strength lies in its component-based structure, enabling developers to create reusable and manageable UI elements, which can be nested to construct complex applications.