

PES Institute of Technology and Management

NH-206, Sagar Road, Shivamogga-577204



Department of Computer Science and Engineering

Affiliated to

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka –590018



Laboratory Manual

VI Semester B.E REACT Laboratory (BCSL657B)

Prepared By,

**Mr. Ragavendra K,
Assistant Professor,
Department of CSE**

**Mr. Shravan H S,
Assistant Professor,
Department of CSE**



PES Institute of Technology and Management

Department of Computer Science and Engineering

SYLLABUS

RECT- VI			
Course Code	BCSL657B	CIE Marks	50
Number of Contact Hours/Week	0:0:2:0	SEE Marks	50
Total Number of Lab Contact Hours	40	Exam Hours	100
Credits-1			
Course Learning Objectives:			
<ul style="list-style-type: none">• Enable students to develop React applications utilizing functional and class-based components, effectively managing state with hooks and lifecycle methods.• Introduce, how to pass data dynamically between parent and child components using props, ensuring modular and reusable component design.• Create dynamic and responsive applications, integrating forms, validation, task management systems, and styled components.• Use React Router for navigation, external API integration for dynamic data handling, and CSS styling techniques for modern UI/UX design.			
Descriptions(if any):			
<ul style="list-style-type: none">• Implement all the programs in "REACT" and Linux OS.			
Programs List:			
1.	Use create-react-app to set up a new project. Edit the App.js file to include a stateful component with useState . <ol style="list-style-type: none">1. Add an input field and a <h1> element that displays text based on the input.2. Dynamically update the <h1> content as the user types.		
2.	Develop a React application that demonstrates the use of props to pass data from a parent component to child components. <ol style="list-style-type: none">1. The application should include the parent component named App that serves as the central container for the application.2. Create two separate child components,<ol style="list-style-type: none">a. Header: Displays the application title or heading.b. Footer: Displays additional information, such as copyright details or a tagline.3. Pass data (e.g., title, tagline, or copyright information) from the App component to the Header and Footer components using props.4. Ensure that the content displayed in the Header and Footer components is dynamically updated based on the data received from the parent component.		
3.	Create a Counter Application using React that demonstrates state management with the useState hook <ol style="list-style-type: none">1. Display the current value of the counter prominently on the screen.2. Add buttons to increase and decrease the counter value.3. Add a "Reset" button to set the counter back to its initial value. Include functionality to specify a custom increment or decrement step value.		



PES Institute of Technology and Management

Department of Computer Science and Engineering

	<ol style="list-style-type: none">4. Ensure the counter updates dynamically when the buttons are clicked. Use the useState hook to manage the counter's state within the component.5. Prevent the counter from going below a specified minimum value (e.g., 0).
4.	<p>Develop a To-Do List Application using React functional components that demonstrates the use of the useState hook for state management.</p> <ol style="list-style-type: none">1. Create a functional component named ToDoFunction to manage and display the to do list.2. Provide an input field for users to add new tasks.3. Maintain a list of tasks using state.4. Mark tasks as completed or pending and visually differentiate them.5. Allow users to delete tasks from the list.6. Dynamically render the list of tasks below the input field. Ensure each task is displayed in a user-friendly manner.
5.	<p>Develop a React application that demonstrates component composition and the use of props to pass data.</p> <ol style="list-style-type: none">1. Create two components: FigureList: A parent component responsible for rendering multiple child components.2. BasicFigure: A child component designed to display an image and its associated caption.3. Use the FigureList component to dynamically render multiple BasicFigure components.4. Pass image URLs and captions as props from the FigureList component to each BasicFigure component.5. Style the BasicFigure components to display the image and caption in an aesthetically pleasing manner.6. Arrange the BasicFigure components within the FigureList in a grid or list format.7. Allow users to add or remove images dynamically. Add hover effects or animations to the images for an interactive experience.
6.	<p>Design and implement a React Form that collects user input for name, email, and password. Form Fields are Name, Email, Password.</p> <ol style="list-style-type: none">1. Ensure all fields are filled before allowing form submission.2. Validate the email field to ensure it follows the correct email format (e.g., example@domain.com).3. Optionally enforce a minimum password length or complexity.4. Display error messages for invalid or missing inputs.5. Provide visual cues (e.g., red borders) to highlight invalid fields.6. Prevent form submission until all fields pass validation. Log or display the entered data upon successful submission (optional).7. Add a "Show Password" toggle for the password field. Implement client side sanitization to ensure clean input.



PES Institute of Technology and Management

Department of Computer Science and Engineering

7.	<p>Develop a React Application featuring a ProfileCard component to display a user's profile information, including their name, profile picture, and bio.</p> <ol style="list-style-type: none">1. The component should demonstrate flexibility by utilizing both external CSS and in-line styling for its design.2. Display the following information: Profile picture, User's name, A short bio or description Use an external CSS file for overall structure and primary styles, such as layout, colors, and typography. Apply inline styles for dynamic or specific styling elements, such as background colors or alignment.3. Design the ProfileCard to be visually appealing and responsive. Ensure the profile picture is displayed as a circle, and the name and bio are appropriately styled. Add hover effects or animations to enhance interactivity. Allow the background color of the card to change dynamically based on a prop or state.
8.	<p>Develop a Reminder Application that allows users to efficiently manage their tasks.</p> <ol style="list-style-type: none">1. The application should include the following functionalities: Provide a form where users can add tasks along with due dates.2. The form includes task name, Due date, An optional description.3. Display a list of tasks dynamically as they are added. Show relevant details like task name, due date, and completion status. Include a filter option to allow users to view all Tasks and Display all tasks regardless of status. Show only tasks marked as completed. Show only tasks that are not yet completed.
9.	<p>Design a React application that demonstrates the implementation of routing using the react-router-dom library.</p> <ol style="list-style-type: none">1. The application should include the Navigation Menu: Create a navigation bar with links to three distinct pages, Home, About, Contact.2. Develop separate components for each page (Home, About, and Contact) with appropriate content to differentiate them.3. Configure routes using react-router-dom to render the corresponding page component based on the selected link. Use BrowserRouter and Route components for routing. Highlight the active link in the navigation menu to indicate the current page
10.	<p>Design a React application featuring a class-based component that demonstrates the use of lifecycle methods to interact with an external API.</p> <ol style="list-style-type: none">1. The component should fetch, and update data dynamically based on user interactions or state changes. Use the componentDidMount lifecycle method to fetch data from an API when the component is initially rendered. Display the fetched data in a structured format, such as a table or list.2. Use the componentDidUpdate lifecycle method to detect changes in the component's state or props.3. Trigger additional API calls to update the displayed data based on user input or actions (e.g., filtering, searching, or pagination).4. Implement errors in handling to manage issues such as failed API requests or empty data responses. Display appropriate error messages to the user when necessary.5. Allow users to perform actions like filtering, searching, or refreshing the data. Reflect changes in the data displayed based on these interactions.



PES Institute of Technology and Management

Department of Computer Science and Engineering

Laboratory Outcomes:

The student should be able to:

- Illustrate React basics and state components.
- Develop React applications that utilize component composition, passing data through props.
- Use dynamic state updates, event handling, and custom logic to increment, decrement, and re-set state values.
- Implement forms in React that collect and validate user input.
- Demonstrate interaction with external APIs, dynamic content generation and manage state in real-time applications.

Conduct of Practical Examination:

- Experiment distribution
 - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
 - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (*Need to change in accordance within university regulations*)
 - c) For laboratories having only one part–Procedure+Execution+Viva-Voice: 15+70+15= 100 Marks
 - d) For laboratories having PARTA and PARTB
 - i. Part A–Procedure + Execution +Viva =6+28+6=40 Marks
 - ii. Part B–Procedure + Execution +Viva=9+ 42+9=60 Marks



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program 1. Use create-react-app to set up a new project. Edit the App.js file to include a stateful component with useState.

1. Add an input field and a `<h1>` element that displays text based on the input. Dynamically update the `<h1>` content as the user types.

Step 1:- Create React Application (open terminal and type below command)

➤ `npx create-react-app myprog1`

Step 2:- go to that folder

➤ `cd myprog1`

Step 3:- Type below command to open react application in visual studio

➤ `code .`

Step 4:- expand the folder (myprog1)

Step 5:- Goto **src** folder present in **myprog1** folder

Step 6:- Expand src folder and click on app.js file present inside src folder and paste the below code and save it.

```
import React, { useState } from 'react';
import './App.css';

function App() {
  const [inputText, setInputText] = useState("");
  return (
    <div className="App">
      <header className="App-header">
        <input
          type="text"
          placeholder="Type something..."
          value={inputText}
          onChange={(event) => {setInputText(event.target.value)}}
        />
        <p>You typed: {inputText}</p>
      </header>
    </div>
  );
}

export default App;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 7:- Open terminal in visual studio and type the below command

➤ npm run start

Step 8:- A browser will open up and shows the below output in the browser

You typed:

Step 9:- Type Something inside a box and In the beside you typed we can see text entered in the box. It is getting updated dynamically.

You typed: Good morning



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program:- 2 Develop a React application that demonstrates the use of props to pass data from a parent component to child components.

1. The application should include the parent component named App that serves as the central container for the application.
2. Create two separate child components,
 - a) Header: Displays the application title or heading.
 - b) Footer: Displays additional information, such as copyright details or a tagline.
3. Pass data (e.g., title, tagline, or copyright information) from the App component to the Header and Footer components using props.
4. Ensure that the content displayed in the Header and Footer components is dynamically updated based on the data received from the parent component.

Step 1:- Create React Application (open terminal and type below command)

➤ `npx create-react-app myprog2`

Step 2:- go to that folder

➤ `cd myprog2`

Step 3:- Type below command to open react application in visual studio

➤ `code .`

Step 4:- expand the folder (myprog2)

Step 5:- Goto **src** folder present in **myprog2** folder

Step 6:- Expand src folder and click on app.js file present inside src folder and paste the below code and save it.



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
import React from 'react';
import Header from './Header';
import Footer from './Footer';
import './App.css';

function App() {
  const title = "Welcome to My React App";
  const tagline = "Building great apps with React";
  const copyright = "© 2025 MyApp, All Rights Reserved";

  return (
    <div className="App">
      <Header title={title} />
      <Footer tagline={tagline} copyright={copyright} />
    </div>
  );
}

export default App;
```

Step 7:- Expand src folder and right click on src folder and create a new file with name Header.js and Note down the below code



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
import React from 'react';

function Header(props) {
  return (
    <header>
      <h1>{props.title}</h1>
    </header>
  );
}

export default Header;
```

Step 8:- Expand src folder and right click on src folder and create a new file with name Footer.js and Note down the below code

```
import React from 'react';

function Footer(props) {
  return (
    <footer>
      <p>{props.tagline}</p>
      <p>{props.copyright}</p>
    </footer>
  );
}

export default Footer;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 9:- Expand src folder and click on app.css file and note down below code inside app.css

```
.App {  
    text-align: center;  
    font-family: Arial, sans-serif;  
}
```

```
header {  
    background-color: #282c34;  
    padding: 20px;  
    color: white;  
}
```

```
footer {  
    background-color: #282c34;  
    padding: 10px;  
    color: white;  
    position: absolute;  
    bottom: 0;  
    width: 100%;  
    text-align: center;  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 10:- Run the react application using below command

➤ npm run start

Step 11:- a browser will be opened and in the browser below output can be seen

Welcome to My React App

Building great apps with React
© 2025 MyApp, All Rights Reserved



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program: - 3 Develop Create a Counter Application using React that demonstrates state management with the useState hook

1. Display the current value of the counter prominently on the screen.
2. Add buttons to increase and decrease the counter value.
3. Add a "Reset" button to set the counter back to its initial value. Include functionality to specify a custom increment or decrement step value.
4. Ensure the counter updates dynamically when the buttons are clicked. Use the useState hook to manage the counter's state within the component.
5. Prevent the counter from going below a specified minimum value (e.g., 0).

Step 1:- Create React Application (open terminal and type below command)

➤ `npx create-react-app myprog3`

Step 2:- go to that folder

➤ `cd myprog3`

Step 3:- Type below command to open react application in visual studio

➤ `code .`

Step 4:- expand the folder (myprog3)

Step 5:- Goto **src** folder present in **myprog3** folder

Step 6:- Expand src folder and click on app.js file present inside src folder and paste the below code and save it.

```
import React, { useState } from 'react';
import './App.css';

function App() {
  const [counter, setCounter] = useState(0);
  const [step, setStep] = useState(1);
  const minValue = 0;
  return (
    <div style={{ textAlign: 'center', marginTop: '50px' }}>
      <h1>Counter Application</h1>
      <div style={{ fontSize: '48px', margin: '20px' }}>
        <span>{ counter }</span>
      </div>
    </div>
  );
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
</div>

<div>

  <button onClick={() => {setCounter(counter + step);}}>Increase</button>

  <button onClick={() => {if (counter - step >= min Value) {setCounter(counter - step); }};
}>Decrease</button>

  <button onClick={() => { setCounter(0);}}>Reset</button>
</div>

<div style={{ marginTop: '20px' }}>

  <label>

    Set Increment/Decrement Step:

    <input

      type="number"

      value={step}

      onChange={(event) => {setStep(Number(event.target.value));}} min="1" style={{
marginLeft: '10px' }} />

    </label>

  </div>

</div>

);

}

export default App;
```

Step 7:- Run the React application using below command in Visual studio terminal
➤ npm run start

Step 8:- A browser will open up. In the browser we can see the below output. Before incrementing/decrementing the counter please provide a step value.



PES Institute of Technology and Management

Department of Computer Science and Engineering

Counter Application

0

Increase by 1

Decrease by 1

Reset

Set Increment/Decrement Step:



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program: - 4 Develop a To-Do List Application using React functional components that demonstrates the use of the useState hook for state management.

1. Create a functional component named ToDoFunction to manage and display the to do list.
2. Provide an input field for users to add new tasks.
3. Maintain a list of tasks using state.
4. Mark tasks as completed or pending and visually differentiate them.
5. Allow users to delete tasks from the list.
6. Dynamically render the list of tasks below the input field. Ensure each task is displayed in a user-friendly manner.

Step 1:- Create React Application (open terminal and type below command)

➤ npx create-react-app myprog4

Step 2:- go to that folder

➤ cd myprog4

Step 3:- Type below command to open react application in visual studio

➤ code .

Step 4:- expand the folder (myprog4)

Step 5:- Goto **src** folder present in **myprog4** folder

Step 6:- Expand src folder and click on app.js file present inside src folder and paste the below code and save it.

```
import React, {useState} from 'react'

function App() {

  /*Variables */

  /*1) Variables to hold the tasks */

  const [tasks, setTask] = useState([])

  /*2) Variable to hold the new task */

  const [newTask, setNewTask] = useState("")

  /*Arrow Functions */

  /*1) Add a task */

  const addTask = () => {

    if(tasks.trim!==""){

      setTask(tasks.concat({id:Date.now(), text:newTask, completed:false}))

      setNewTask("");

    }

  };

};
```




PES Institute of Technology and Management

Department of Computer Science and Engineering

/*2) Mark a particular task Completed using id property*/

```
const markComplete = (id) => {  
  setTask(tasks.map((task) => {  
    if(task.id === id){  
      return Object.assign({}, task, {completed: !task.completed})  
    }  
    else {  
      return task; } }));  
};
```

/*3) Remove a Task */

```
const removeTask = (id) =>  
  setTask(tasks.filter((task) => task.id !== id))
```

```
return (  
  <div>
```

{/*Input text box for taking task name from user */}

```
<input  
  type='text'  
  value={newTask}  
  onChange={(event) => setNewTask(event.target.value)}  
  style={{ marginLeft: '250px', borderRadius: '10px', border: '10px solid green', padding: '10px',  
    textAlign:'center'}} >  
</input>
```

{/*Button to add task to list or array */}

```
<button onClick={addTask} style={{ borderRadius:'10px', backgroundColor: 'orange'}}>Add Task  
</button>
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

`{/*List Of Task using unordered list */}`

``

`{tasks.map(task=>`

`<li key={task.id}`

`style={{ color: task.completed? 'green' : 'red',`

`padding: task.completed? '20px' : '20px',`

`backgroundColor: task.completed? 'grey': 'orange',`

`margin: task.completed? '10px' : '10px',`

`fontSize: task.completed? '30px' : '30px',`

`fontWeight: task.completed? 'bold' : 'bold',`

`textDecoration: task.completed? 'line-through' : 'none',`

`border: task.completed? '4px solid grey' : '6px solid yellow'`

`}} >`

` markComplete(task.id)}>{task.text}`

`<button onClick={()=> removeTask(task.id)}`

`style={{ backgroundColor:'white', marginLeft:'100px', padding:'10px',
fontSize:'30px', borderRadius: '10px' }}>`

`Delete</button>`

``

`}}`

``

`</div>`

`}}`

`export default App`

Step 7:- Run the React application using below command

➤ `npm run start`

Step 8:- A browser will open up. In the browser we can see the below output





PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 9:- Enter a task name inside the green colour text box

Add Task

Step 10:- Click on Add Task Button and below output can be seen after clicking on add task button

Add Task

- Reading A Book Every Day (10 Pages)

Delete

Step 11:- If we want to make particular task as completed and we need to click on task name. (for ex:- we need to click on “**Reading a Book Every Day**”). Below output can be seen after clicking on task name (marking task as completed)

Add Task

- Reading A Book Every Day (10 Pages)

Delete

Step 12:- We can add Multiple tasks by providing different task name based on the user prioretess (See the below output)

Add Task

- Reading A Book Every Day (10 Pages)

Delete
- Daily assign Two hours for coding

Delete

Step 13:- Delete a task can be done by clicking on delete button present in particular task



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program: - 5 Develop a React application that demonstrates component composition and the use of props to pass data.

1. Create two components: **FigureList**: A parent component responsible for rendering multiple child components.
2. **BasicFigure**: A child component designed to display an image and its associated caption.
3. Use the **FigureList** component to dynamically render multiple **BasicFigure** components.
4. Pass image URLs and captions as props from the **FigureList** component to each **BasicFigure** component.
5. Style the **BasicFigure** components to display the image and caption in an aesthetically pleasing manner.
6. Arrange the **BasicFigure** components within the **FigureList** in a grid or list format.
7. Allow users to add or remove images dynamically. Add hover effects or animations to the images for an interactive experience.

Step 1:- Create React Application (open terminal and type below command)

- `npx create-react-app myprog5`

Step 2:- go to that folder

- `cd myprog5`

Step 3:- Type below command to open react application in visual studio

- `code .`

Step 4:- expand the folder (myprog5)

Step 5:- Goto **src** folder present in **myprog5** folder

Step 6:- Right Click on **src** folder and create a new file named **BasicFigure.js**

Step 7:- Copy the below code inside the **BasicFigure.js** (This is Child Component)



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
import React from 'react'

import './FigureListStyles.css'

function BasicFigure ({imageSource, imageCaption, removeCard}) {

  return (

    <div className='figure-container'>

      /*Image Tag which takes source URL and Caption To display */

      <img

        src={imageSource} alt={imageCaption} className='figure-image'/>

      <p className='figure-caption'>{imageCaption}</p>

      <button onClick={removeCard} className='remove-button'>X</button>

    </div>

  )

}

export default BasicFigure;
```

Step 8:- Right Click on **src folder** and create a new file named **FigureList.js**

Step 9:- Copy the below code inside the **FigureList.js** (This is a Parent Component)



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
import React,{useState} from 'react'
import BasicFigure from './BasicFigure'
import './FigureListStyles.css'

function FigureList() {
  /*State variables to store data */
  /*Array to hold the list of images */
  const [figList, setFigList] = useState([])
  /*Varibale to hold the image source URL */
  const [imageSourceURL, setImageSourceURL] = useState("")
  /*Variable to hold the Image caption */
  const [imageCaptions, setImageCaptions] = useState("")
  /*Arrow Functions */
  /*1) To add image and caption to array */
  const addImage = () => {
    /*Check wheather image url and image caption are not empty */
    if(imageSourceURL.trim() !== "" && imageCaptions.trim() !== ""){
      setFigList(figList.concat({ id: Date.now(), imageSourceURL, imageCaptions}))
      setImageCaptions("")
      setImageSourceURL("")
    }
  };
  /*2) To remove particular image and caption from the array based on ID property */
  const removeImages = (id) => setFigList(figList.filter((figures)=> figures.id !== id))
  return (
    <div className='container'>
      <div className='input-container'>
        <span>Provide Image URL:- </span>
        <input
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
type='text'
    value={imageSourceURL}
    onChange={(event)=> setImageSourceURL(event.target.value)}
  ></input>
<span>Provide Caption/Description:- </span>
  <input
    type='text'
    value={imageCaptions}
    onChange={(event)=> setImageCaptions(event.target.value)}
  ></input>
</div>
<button onClick={addImage}>Add Image</button>
<div className='grid-layout'>
  {figList.map(({id, imageSourceURL, imageCaptions})=>
    <BasicFigure
      key={id}
      imageSource={imageSourceURL}
      imageCaption={imageCaptions}
      removeCard={() => removeImages(id)}
    ></BasicFigure>
  )}
</div>
</div>
)
}
export default FigureList
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 11:- Goto **App.js** file present in **src** folder and Copy the below code inside the **App.js** (This Main App component)

```
import React from 'react'
import FigureList from './FigureList'
function App() {
  return (
    <div>
      <FigureList></FigureLis>
    </div>
  )
}

export default App
```

Step 12:- Right Click on **src folder** and create a new file named **FigureListStyles.css**

Step 13:- Copy the below code inside the **FigureListStyles.css** (This is CSS file)



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
.figure-container {  
    border-radius: 1rem;  
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
    overflow: hidden;  
    position: relative;  
    border: 1px solid #e5e7eb;  
    padding: 0.5rem;  
    transition: transform 0.2s ease-in-out;  
}  
  
.figure-container:hover {  
    transform: scale(1.05);  
}  
  
.figure-image {  
    width: 100%;  
    height: 12rem;  
    object-fit: cover;  
    border-radius: 0.5rem;  
}  
  
.figure-caption {  
    text-align: center;  
    margin-top: 0.5rem;  
    color: #4b5563;  
    font-weight: 500;  
}  
  
.remove-button {  
    position: absolute;  
    top: 0.5rem;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
right: 0.5rem;  
background-color: #ef4444;  
color: white;  
border-radius: 50%;  
padding: 0.25rem 0.5rem;  
opacity: 0;  
transition: opacity 0.3s ease-in-out;  
}
```

```
.figure-container:hover .remove-button {  
  opacity: 1;  
}
```

```
.container {  
  padding: 1.5rem;  
  max-width: 48rem;  
  margin: auto;  
}
```

```
.input-container {  
  display: flex;  
  gap: 0.5rem;  
  margin-bottom: 1rem;  
}
```

```
.grid-layout {  
  display: grid;  
  grid-template-columns: repeat(2, 1fr);  
  gap: 1rem;  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 14:- open Terminal in visual studio and type below command

➤ npm run start

Step 15:- Chrome browser will open after typing npm start and Below output can be seen in the browser

Provide Image URL:-

Provide Caption/Description

Add Image

Step 16:- Connect to internet first and provide url links (ex:- <https://picsum.photos/seed/picsum/200/300>) (more image urls can be seen in this website:- <https://picsum.photos/>) and provide caption/ description based on image url.

Provide Image URL:-

Provide Caption/Description

Add Image



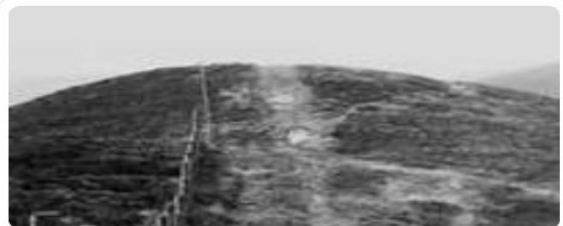
Image 1



Dog



Mountain



GreyScale Image



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 17:- Remove particular image and caption (using id property) can be done by placing mouse over that image (cancel button can be seen)

Provide Image URL:- Provide Caption/Description

Add Image



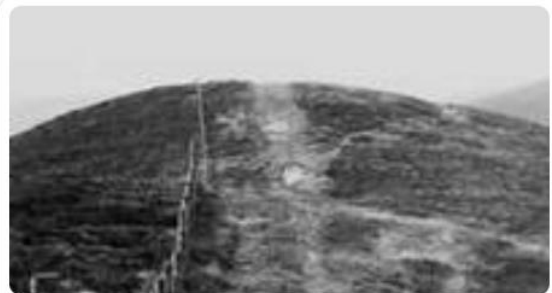
Image 1



Dog



Mountain



GreyScale Image



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program: - 6

Design and implement a React Form that collects user input for name, email, and password. Form Fields are Name, Email, Password.

1. Ensure all fields are filled before allowing form submission.
2. Validate the email field to ensure it follows the correct email format (e.g., example@domain.com).
3. Optionally enforce a minimum password length or complexity.
4. Display error messages for invalid or missing inputs.
5. Provide visual cues (e.g., red borders) to highlight invalid fields.
6. Prevent form submission until all fields pass validation. Log or display the entered data upon successful submission (optional).
7. Add a "Show Password" toggle for the password field. Implement client side sanitization to ensure clean input.

Step 1:- Create React Application (open terminal and type below command)

- `npx create-react-app myprog6`

Step 2:- go to that folder(In terminal)

- `cd myprog6`

Step 3:- Type below command to open react application in visual studio

- `code .`

Step 4:- expand the folder (myprog6)

Step 5:- Goto **src** folder present in **myprog6** folder

Step 6:- Goto **App.js** file present in **src** folder and Copy the below code inside the **App.js** (This Main App component)



PES Institute of Technology and Management

Department of Computer Science and Engineering

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

import './UseForm.css'; // Import the CSS file

const UserForm = () => {

  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [showPassword, setShowPassword] = useState(false);
  const [errors, setErrors] = useState({ });

  const sanitizedName = name.trim();
  const sanitizedEmail = email.trim();
  const sanitizedPassword = password.trim();

  const validateEmail = (email) => {
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return emailRegex.test(email);
  };

  const validateForm = () => {

    let isValid = true;
    const newErrors = { };

    if (!sanitizedName) {
      newErrors.name = 'Name is required';
      isValid = false;
    }
  }
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
if (!sanitizedEmail) {  
    newErrors.email = 'Email is required';  
    isValid = false;  
} else if (!validateEmail(sanitizedEmail)) {  
    newErrors.email = 'Invalid email format';  
    isValid = false;  
}  
  
if (!sanitizedPassword) {  
    newErrors.password = 'Password is required';  
    isValid = false;  
} else if (sanitizedPassword.length < 6) {  
    newErrors.password = 'Password must be at least 6 characters long';  
    isValid = false;  
}  
  
setErrors(newErrors);  
return isValid;  
};  
  
const handleSubmit = (event) => {  
    event.preventDefault();  
    if (validateForm()) {  
        const sanitizedName = name.trim();  
        const sanitizedEmail = email.trim();  
        const sanitizedPassword = password.trim();  
  
        console.log('Form submitted:', {  
            name: sanitizedName,  
            email: sanitizedEmail,  
            password: sanitizedPassword,
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
    });  
  }  
};  
  
const handleInputChange = (event) => {  
  const { name, value } = event.target;  
  switch (name) {  
    case 'name':  
      setName(value);  
      break;  
    case 'email':  
      setEmail(value);  
      break;  
    case 'password':  
      setPassword(value);  
      break;  
    default:  
      break;  
  }  
  setErrors(prevErrors => {  
    const updatedErrors = Object.assign({}, prevErrors);  
    updatedErrors[name] = "";  
    return updatedErrors;  
  });  
};  
  
const toggleShowPassword = () => {  
  setShowPassword(!showPassword);  
};
```




PES Institute of Technology and Management

Department of Computer Science and Engineering

```
return (  
  <form onSubmit={handleSubmit} className="form-container">  
    <div className="form-group">  
      <label htmlFor="name">Name:</label>  
      <input  
        type="text"  
        id="name"  
        name="name"  
        value={name}  
        onChange={handleInputChange}  
        className={errors.name ? 'invalid' : ''}  
      />  
      {errors.name && <p className="error-message">{errors.name}</p>}  
    </div>  
  
    <div className="form-group">  
      <label htmlFor="email">Email:</label>  
      <input  
        type="email"  
        id="email"  
        name="email"  
        value={email}  
        onChange={handleInputChange}  
        className={errors.email ? 'invalid' : ''}  
      />  
      {errors.email && <p className="error-message">{errors.email}</p>}  
    </div>  
  
    <div className="form-group">  
      <label htmlFor="password">Password:</label>  
      <input
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
type={ showPassword ? 'text' : 'password'}
id="password"
name="password"
value={ password}
onChange={ handleInputChange}
className={errors.password ? 'invalid' : ''}
/>
{errors.password && <p className="error-message">{ errors.password}</p>}
<div className="password-toggle">
  <input
    type="checkbox"
    id="showPassword"
    checked={ showPassword}
    onChange={ toggleShowPassword}
  />
  <label htmlFor="showPassword">Show Password</label>
</div>
</div>

<button type="submit" >
  Submit
</button>
</form>
);
};

export default UserForm;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 7:- Create a file in src folder with name UseForm.css and add the below content

```
.form-container {  
    max-width: 500px;  
    margin: 0 auto;  
    padding: 20px;  
    background: #f9f9f9;  
    border-radius: 8px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
.form-group {  
    margin-bottom: 15px;  
}  
.form-group label {  
    display: block;  
    margin-bottom: 5px;  
    font-weight: bold;  
}  
.form-group input {  
    width: 100%;  
    padding: 8px;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    box-sizing: border-box;  
}  
.password-input {  
    position: relative;  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
.toggle-password {  
    position: absolute;  
    right: 10px;  
    top: 50%;  
    transform: translateY(-50%);  
    background: none;  
    border: none;  
    cursor: pointer;  
    color: #666;  
}
```

```
.password-hint {  
    font-size: 0.8em;  
    color: #666;  
    margin-top: 5px;  
}
```

```
.submit-btn {  
    background-color: #4CAF50;  
    color: white;  
    padding: 10px 15px;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    font-size: 16px;  
}
```

```
.submit-btn:hover {  
    background-color: #45a049;  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
.submit-btn:disabled {  
  background-color: #cccccc;  
  cursor: not-allowed;  
}
```

```
.error-message {  
  color: #d9534f;  
  font-size: 0.8em;  
  margin-top: 5px;  
  display: block;  
}
```

```
.error input {  
  border-color: #d9534f;  
}
```

```
.success-message {  
  background: #dff0d8;  
  padding: 15px;  
  border-radius: 4px;  
  color: #3c763d;  
}
```

Step 8:- open Terminal in visual studio and type below command

➤ npm run start

Step 9:- Below output can be seen in browser and enter the values in the respective field and click on submit and after submitting press ctrl+shift+I to see the output in console or right click on the page and click on inspect and then click on console.



PES Institute of Technology and Management

Department of Computer Science and Engineering

Name:

Suresh Reddy

Email:

sureshreddy13@gmail.com

Password:

.....

☐

Show Password

Submit

Name:

Name is required

Email:

sureshreddy13@gmail.com

Password:

.....

☐

Show Password

Submit



PES Institute of Technology and Management

Department of Computer Science and Engineering

Name:

Suresh Readdy

Email:

sureshreddy13|



Please include an '@' in the email address. 'sureshreddy13' is missing an '@'.

.....

☐

Show Password

Submit

Name:

Suresh Readdy

Email:

sureshreddy13@gmail.com

Password:

....

Password must be at least 6 characters long

☐

Show Password

Submit



PES Institute of Technology and Management

Department of Computer Science and Engineering

Name:

Suresh Readdy

Email:

sureshreddy13@gmail.com

Password:

qwerty..



Show Password

Submit

Name:

Suresh Readdy

Email:

sureshreddy13@gmail.com

Password:



Show Password

Submit

react-dom-client.development.js:24593

Download the React DevTools for a better development experience:

<https://react.dev/link/react-devtools>

Form submitted:

App.jsx:59

```
▼ {name: 'Suresh Readdy', email: 'sureshreddy13@gmail.com', password: 'qwerty..'} ⓘ  
  email: "sureshreddy13@gmail.com"  
  name: "Suresh Readdy"  
  password: "qwerty.."
```

```
▶ [[Prototype]]: Object
```

>



PES Institute of Technology and Management

Department of Computer Science and Engineering

Name:

Email:

Password:

☐

Show Password

Submit

Program: - 7

Develop a React Application featuring a ProfileCard component to display a user's profile in-formation, including their name, profile picture, and bio.

1. The component should demonstrate flexibility by utilizing both external CSS and inline styling for its design.
2. Display the following information: Profile picture, User's name, A short bio or de-scription Use an external CSS file for overall structure and primary styles, such as layout, colors, and typography. Apply inline styles for dynamic or specific styling el-ements, such as background colors or alignment.
3. Design the ProfileCard to be visually appealing and responsive. Ensure the profile pic-ture is displayed as a circle, and the name and bio are appropriately styled. Add hover effects or animations to enhance interactivity. Allow the background color of the card to change dynamically based on a prop or state.

Step 1:- Create React Application (open terminal and type below command)

➤ npx create-react-app myprog7



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 2:- go to that folder(In terminal)

➤ `cd myprog7`

Step 3:- Type below command to open react application in visual studio

➤ `code .`

Step 4:- expand the folder (myprog7)

Step 5:- Goto **src** folder present in **myprog7** folder

Step 6:- Goto **App.js** file present in **src** folder and Copy the below code inside the **App.js** (This Main App component)

```
import React, { useState } from 'react';
import ProfileCard from './ProfileCard';
import './ProfileCard.css';
function App() {
  /*Bgcolor state variable to hold the color */
  const [bgColor, setBgColor] = useState('#f0f0f0');
  return (
    <div className="app">
      <ProfileCard
        name="Jane Doe"
        bio="Front-end Developer. Loves React and design."
        image="https://avatar.iran.liara.run/public/boy"
        backgroundColor={bgColor}
      />
      <button onClick={() => setBgColor('#cce7ff')}>Change Background</button>
    </div>
  );
}
export default App;
```

Step 7:- Right Click on **src folder** and create a new file named **ProfileCard.js**

Step 8:- Copy the below code inside the **ProfileCard.js**



PES Institute of Technology and Management

Department of Computer Science and Engineering

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

function ProfileCard({ name, bio, image, backgroundColor }) {
  const cardStyle = {
    backgroundColor: backgroundColor,
    textAlign: 'center'
  };
  return (
    <div className="profile-card" style={cardStyle}>
      <img src={image} alt="Profile" className="profile-image" />
      <h2 className="profile-name">{name}</h2>
      <p className="profile-bio">{bio}</p>
    </div>
  );
}

export default ProfileCard;
```

Step 9:- Right Click on **src folder** and create a new file named **ProfileCard.css**

Step 10:- Copy the below code inside the **ProfileCard.css**



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
.profile-card {  
  width: 300px;  
  margin: 20px auto;  
  padding: 20px;  
  border-radius: 15px;  
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
  transition: transform 0.3s ease, box-shadow 0.3s ease;  
}  
  
.profile-card:hover {  
  transform: translateY(-5px);  
  box-shadow: 0 6px 12px rgba(0, 0, 0, 0.3);  
}  
  
.profile-image {  
  width: 100px;  
  height: 100px;  
  border-radius: 50%;  
  object-fit: cover;  
}  
  
.profile-name {  
  font-size: 1.5rem;  
  margin: 10px 0 5px;  
  color: #333;  
}  
  
.profile-bio {  
  font-size: 1rem;  
  color: #666;  
}  
  
.app button {  
  display: block;  
  margin: 20px auto;  
  padding: 8px 16px;  
  font-size: 1rem; }
```



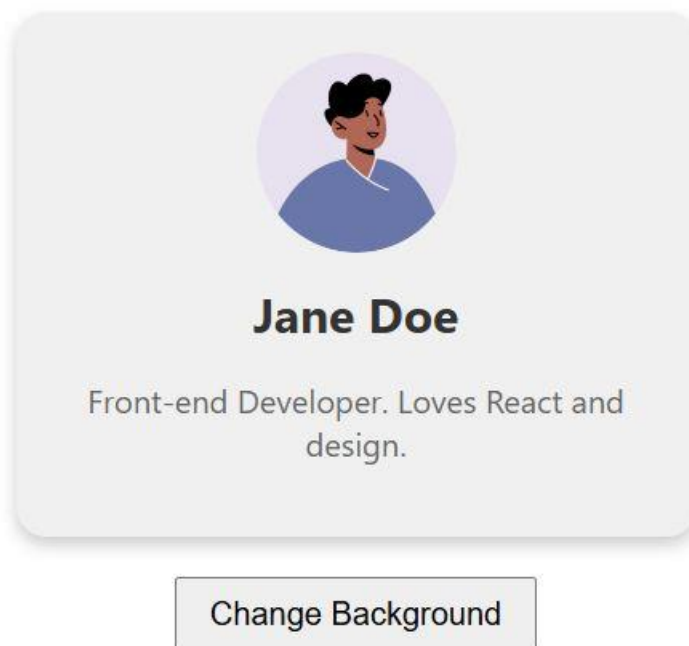
PES Institute of Technology and Management

Department of Computer Science and Engineering

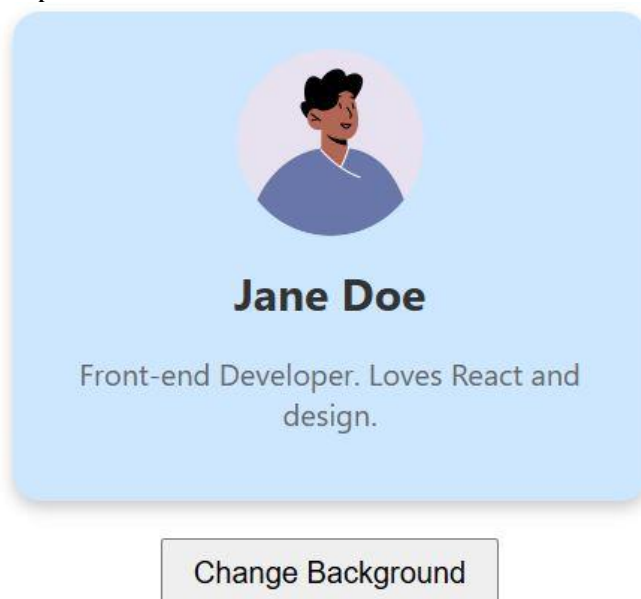
Step 11:- open Terminal in visual studio and type below command

➤ `npm run start`

Step 12:- Chrome browser will open after typing npm start and Below output can be seen in the browser



Step 13:- Click on change background button to see the background colour change. Below output can be seen in the browser.





PES Institute of Technology and Management

Department of Computer Science and Engineering

Program: - 8

Develop a Reminder Application that allows users to efficiently manage their tasks.

1. The application should include the following functionalities: Provide a form where users can add tasks along with due dates.
2. The form includes task name, Due date, An optional description.
3. Display a list of tasks dynamically as they are added. Show relevant details like task name, due date, and completion status. Include a filter option to allow users to view all Tasks and Display all tasks regardless of status. Show only tasks marked as completed. Show only tasks that are not yet completed.

Step 1:- Create React Application (open terminal and type below command)

- `npx create-react-app myprog8`

Step 2:- go to that folder(In terminal)

- `cd myprog8`

Step 3:- Type below command to open react application in visual studio

- `code .`

Step 4:- expand the folder (myprog8)

Step 5:- Goto **src** folder present in **myprog8** folder

Step 6:- Goto **App.js** file present in **src** folder and Copy the below code inside the **App.js** (This Main App component)



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
import React, { useState } from 'react';  
import TaskForm from './components/TaskForm';  
import TaskList from './components/TaskList';  
import './App.css';
```

```
function App() {  
  const [tasks, setTasks] = useState([]);  
  const [filter, setFilter] = useState('all');  
  
  // Add a new task  
  const addTask = (task) => {  
    setTasks(tasks.concat([task]));  
  };  
  
  // Toggle task completion  
  const toggleTaskCompletion = (id) => {  
    setTasks(tasks.map((task) => {  
      if (task.id === id) {  
        return {  
          id: task.id,  
          text: task.text,  
          completed: !task.completed  
        };  
      } else {  
        return task;  
      }  
    }));  
  };  
  
  // Delete a task  
  const deleteTask = (id) => {  
    setTasks(tasks.filter(task => task.id !== id));  
  };  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
};
```

```
// Filter tasks based on completion status
```

```
const filteredTasks = tasks.filter(task => {  
  if (filter === 'completed') return task.completed;  
  if (filter === 'pending') return !task.completed;  
  return true;  
});
```

```
return (
```

```
<div className="App">
```

```
<h1>Reminder Application</h1>
```

```
{/* Task Form */}
```

```
<TaskForm onAddTask={addTask} />
```

```
{/* Filter Options */}
```

```
<div className="filter">
```

```
<label htmlFor="status-filter">Filter Tasks:</label>
```

```
<select id="status-filter" onChange={(e) => setFilter(e.target.value)} value={filter}>
```

```
<option value="all">All Tasks</option>
```

```
<option value="completed">Completed Tasks</option>
```

```
<option value="pending">Pending Tasks</option>
```

```
</select>
```

```
</div>
```

```
{/* Task List */}
```

```
<TaskList
```

```
tasks={filteredTasks}
```

```
onToggleCompletion={toggleTaskCompletion}
```

```
onDelete={deleteTask}
```

```
/>
```




PES Institute of Technology and Management

Department of Computer Science and Engineering

```
</div>  
  
);  
  
}  
  
export default App;
```

Step 7:- right click on **src** folder and create a new folder with name **components**. Right click on **components** folder create a new file with name **TaskForm.js**. Copy the below code inside TaskForm.js file.

```
import React, { useState } from 'react';  
  
function TaskForm({ onAddTask }) {  
  const [taskName, setTaskName] = useState("");  
  const [dueDate, setDueDate] = useState("");  
  const [description, setDescription] = useState("");  
  
  const handleSubmit = (e) => {  
    e.preventDefault();  
  
    if (!taskName || !dueDate) return; // Ensure task name and due date are provided.  
  
    const newTask = {  
      id: Date.now(),  
      name: taskName,  
      dueDate,  
      description,  
      completed: false,  
    };  
  }  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
onAddTask(newTask);
setTaskName("");
setDueDate("");
setDescription("");
};
return (
  <form onSubmit={handleSubmit} className="task-form">
    <input
      type="text"
      placeholder="Task Name"
      value={taskName}
      onChange={(e) => setTaskName(e.target.value)}
      required
    />
    <input
      type="date"
      value={dueDate}
      onChange={(e) => setDueDate(e.target.value)}
      required
    />
    <textarea
      placeholder="Optional description"
      value={description}
      onChange={(e) => setDescription(e.target.value)}
    ></textarea>
    <button type="submit">Add Task</button>
  </form>
);
}
```

export default TaskForm;



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 8:- Right click on **components** folder create a new file with name **TaskList.js**. Copy the below code inside **TaskList.js** file.

```
import React from 'react';

function TaskList({ tasks, onToggleCompletion, onDelete }) {
  return (
    <ul className="task-list">
      {tasks.map((task) => (
        <li
          key={task.id}
          className={task.completed ? 'completed' : 'pending'} >
            <div>
              <h3>{task.name}</h3>
              <p>Due Date: {task.dueDate}</p>
              {task.description && <p>{task.description}</p>}
            </div>
            <div>
              <button onClick={() => onToggleCompletion(task.id)}>
                {task.completed ? 'Mark as Pending' : 'Mark as Completed'}
              </button>
              <button onClick={() => onDelete(task.id)}>Delete</button>
            </div>
          </li>
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

}}}

);} }

export default TaskList;



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 9:- Goto app.css in src folder note down the below code.

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f9;  
    margin: 0;  
    padding: 0;  
}  
  
.App {  
    max-width: 600px;  
    margin: 50px auto;  
    padding: 20px;  
    background-color: #fff;  
    border-radius: 8px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}  
  
h1 {  
    text-align: center;  
    margin-bottom: 20px;  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
.task-form {  
  display: flex;  
  flex-direction: column;  
  gap: 10px;  
}  
  
input,  
textarea,  
button {  
  padding: 10px;  
  border-radius: 5px;  
  border: 1px solid #ccc;  
}  
  
button {  
  background-color: #4caf50;  
  color: white;  
  border: none;  
  cursor: pointer;  
}  
  
button:hover {  
  background-color: #45a049;  
}  
  
.filter {  
  margin: 20px 0;  
  display: flex;  
  justify-content: space-between;  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
.task-list {  
  list-style: none;  
  padding: 0;  
}  
  
.task-item {  
  display: flex;  
  justify-content: space-between;  
  padding: 10px;  
  border-bottom: 1px solid #ddd;  
}  
  
.task-item.completed {  
  background-color: #d4edda;  
  text-decoration: line-through;  
}  
  
.task-item.pending {  
  background-color: #fff3cd;  
}  
  
.task-item button {  
  background-color: red;  
  color: white;  
  border: none;  
  cursor: pointer;  
}  
  
.task-item button:hover {  
  background-color: #dc3545;  
}
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 10:- open Terminal in visual studio and type below command

- `npm run start`

Step 11:- Chrome browser will open after typing npm start and Below output can be seen in the browser

Reminder Application



Add Task

Filter Tasks:

All Tasks



Step 12:- Create a task with task name, due date , small description.

Reminder Application



Add Task

Filter Tasks:

All Tasks







PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 13:- Click on add task button. Below output can be seen

Reminder Application





Add Task

Filter Tasks:

All Tasks ▼

Order a Titan Watch from Amazon

Mark as CompletedDelete

Due Date: 2025-05-15

Check the price, color, wrist size, dial pad.

Step 14:- Initially when we create a task , it will be in pending task stage. Filter the pending task. Below output can be seen (Click on the button if we want make particular task as completed or u can delete the task)

Filter Tasks:

Pending Tasks ▼

Order a Titan Watch from Amazon

Mark as CompletedDelete

Due Date: 2025-05-15

Check the price, color, wrist size, dial pad.



PES Institute of Technology and Management

Department of Computer Science and Engineering

0Step 15:- Click on the button if we want make particular task as completed

Filter Tasks:

Completed Tasks ▼

Due Date:

Mark as Pending

Delete



PES Institute of Technology and Management

Department of Computer Science and Engineering

Program 9. Design a React application that demonstrates the implementation of routing using the react-router-dom library.

1. The application should include the Navigation Menu: Create a navigation bar with links to three distinct pages, Home, About, Contact.
2. Develop separate components for each page (Home, About, and Contact) with appropriate content to differentiate them.
3. Configure routes using react-router-dom to render the corresponding page component based on the selected link. Use BrowserRouter and Route components for routing. Highlight the active link in the navigation menu to indicate the current page.

Step 1:- Create React Application (open terminal and type below command)

➤ `npx create-react-app myprog9`

Step 2:- go to that folder(In terminal)

➤ `cd myprog9`

Step 3:- Type below command to open react application in visual studio

➤ `code .`

Step 4:- expand the folder (myprog9)

Step 5:- Goto **src** folder present in **myprog9** folder

Step 6:- Goto **App.js** file present in **src** folder and Copy the below code inside the **App.js** (This Main App component)



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
import React from "react";

import { BrowserRouter as Router, Routes, Route, NavLink } from "react-router-dom";

import Home from "../pages/Home";

import About from "../pages/About";

import Contact from "../pages/Contact";

import "../App.css";

const App = () => {

  return (

    <Router>

      <nav className="nav">

        <NavLink to="/" end className="nav-link">Home</NavLink>

        <NavLink to="/about" className="nav-link">About</NavLink>

        <NavLink to="/contact" className="nav-link">Contact</NavLink>

      </nav>

      <div className="page-content">

        <Routes>

          <Route path="/" element={ <Home /> } />

          <Route path="/about" element={ <About /> } />

          <Route path="/contact" element={ <Contact /> } />

        </Routes>

      </div>

    </Router>

  );

};

export default App;
```



PES Institute of Technology and Management

Department of Computer Science and Engineering

Step 7:- Goto **src** folder and right click on **src** folder create a folder with name **pages**. Right click on **pages** folder and create a new file with name **Home.js** and paste the below content.

```
import React from "react";  
  
const Home = () => <h1>Welcome to the Home Page</h1>;  
  
export default Home;
```

Step 8:- Right click on **pages** folder and create a new file with name **About.js** and paste the below content.

```
import React from "react";  
  
const About = () => <h1>About Us</h1>;  
  
export default About;
```

Step 9:- Right click on **pages** folder and create a new file with name **Contact.js** and paste the below content.

```
import React from "react";  
  
const Contact = () => <h1>Contact Us</h1>;  
  
export default Contact;
```

Step 10:- Goto **src** folder then goto to **app.css** file and write the below content.



PES Institute of Technology and Management

Department of Computer Science and Engineering

```
App.css.nav {  
  background-color: #282c34;  
  padding: 1rem;  
  display: flex;  
  gap: 1rem;  
}  
  
.nav-link {  
  color: white;  
  text-decoration: none;  
  font-weight: 500;  
}  
  
/* Automatically applied by NavLink when active */  
.nav-link.active {  
  border-bottom: 2px solid yellow;  
}  
  
.page-content {  
  padding: 2rem;  
}
```

Step 11:- open Terminal in visual studio and type below command

- **npm run start**

Step 12:- Below output can be seen in the browser