

Task 1: Mastering React Basics and CLI Tools for a Profile Website

Task 21:

Objective:

Set up a React project, create functional components, and explore basic React concepts with a focus on a user profile website. This task aims to enhance your understanding of React basics and how to implement them in a profile website context.

Pre-requisites:

- Basic understanding of HTML, CSS, and JavaScript
- Familiarity with a code editor like Visual Studio Code
- Basic knowledge of React

Concepts Covered:

- Setting Up a React Project
- Creating Functional Components
- Working with JSX
- Using Props and State
- Conditional Rendering and Event Handling

Concepts:

1. Setting Up a React Project:

Initialize a React project using Create React App, tailored for a profile website.

```
npx create-react-app profile-website
```

2. Creating Functional Components:

Create a functional component that returns a welcome message inside a JSX `<div>` for the website.

```
// src/components/WelcomeBanner.js
import React from 'react';

const WelcomeBanner = () => {
  return (
    <div>
      <h1>Welcome to My Profile</h1>
    </div>
  );
}

export default WelcomeBanner;
```

3. Experimenting with JSX:

Enhance the `WelcomeBanner` component with various JSX elements like buttons, images, and inputs relevant to the profile website theme.

```
// src/components/WelcomeBanner.js
import React from 'react';

const WelcomeBanner = () => {
  return (
    <div>
      <h1>Welcome to My Profile</h1>
      <button>Click Me</button>
      
      <input type="text" placeholder="Search..." />
    </div>
  );
}

export default WelcomeBanner;
```

4. Props and State in Action:

Pass a `username` prop to the `WelcomeBanner` component and display it. Introduce a state variable using `useState` to track a user's input (like a search bar) and display its value.

```
// src/components/WelcomeBanner.js
import React, { useState } from 'react';

const WelcomeBanner = ({ username }) => {
  const [search, setSearch] = useState('');

  return (
    <div>
      <h1>Welcome, {username}!</h1>
      <input
        type="text"
        placeholder="Search..."
        value={search}
        onChange={(e) => setSearch(e.target.value)}
      />
      <p>Searching for: {search}</p>
    </div>
  );
}

export default WelcomeBanner;
```

5. Conditional Rendering and Event Handling:

Implement a button in `WelcomeBanner` that toggles a state variable. Use this variable to conditionally render different elements in the component, like showing a login/logout button based on user state.

```
// src/components/WelcomeBanner.js
import React, { useState } from 'react';

const WelcomeBanner = ({ username }) => {
  const [search, setSearch] = useState('');
  const [loggedIn, setLoggedIn] = useState(false);

  return (
    <div>
      <h1>Welcome, {username}!</h1>
      <input
        type="text"
        placeholder="Search..."
        value={search}
        onChange={(e) => setSearch(e.target.value)}
      />
      <p>Searching for: {search}</p>
      <button onClick={() => setLoggedIn(!loggedIn)}>
        {loggedIn ? 'Logout' : 'Login'}
      </button>
      {loggedIn && <p>You are logged in</p>}
    </div>
  );
}

export default WelcomeBanner;
```

Setup:

1. Install Visual Studio Code (VS Code):

Download and install VS Code from [Visual Studio Code](#).

2. React Developer Tools:

Install the React Developer Tools browser extension for [Chrome](#) or [Firefox](#).

3. Git:

Install Git for version control. You can download it from [Git](#).

4. Node.js and npm:

Ensure Node.js and npm are installed on your machine. You can download and install them from [Node.js](#).

5. Setting Up the React Project:

- Open your terminal.
- Navigate to the directory where you want to create your project.
- Run the following command to create a new React project:

```
npx create-react-app profile-website
```

6. Installing Additional Dependencies:

- Navigate to the project directory:

```
cd profile-website
```

- Install any additional dependencies if required. For example:

```
npm install react-router-dom
```

Tasks:

Part 1: Setting Up and Exploring Basic React Concepts

1. Initialize a React Profile Project :

- Run the following command in the terminal to create a new React project:

```
npx create-react-app profile-website
```

- Goal: Learn how to set up a new React project using Create React App, tailored for a profile website.

2. Create a Functional Component

- In the project, create a file `WelcomeBanner.js` in the `src/components` directory and define a functional component that returns a welcome message inside a JSX `<div>` for the website.
- Example:

```
// src/components/WelcomeBanner.js
import React from 'react';

const WelcomeBanner = () => {
  return (
    <div>
      <h1>Welcome to My Profile</h1>
    </div>
  );
}

export default WelcomeBanner;
```

- Goal: Understand the syntax and structure of React functional components, and how they fit into a profile website.

3. Experiment with JSX:

- Enhance the `WelcomeBanner` component with various JSX elements (like buttons, images, and inputs) relevant to the website theme.
- Example:

```
// src/components/WelcomeBanner.js
import React from 'react';

const WelcomeBanner = () => {
  return (
    <div>
      <h1>Welcome to My Profile</h1>
      <button>Click Me</button>
      
      <input type="text" placeholder="Search..." />
    </div>
  );
}

export default WelcomeBanner;
```

- Goal: Get comfortable with JSX syntax and how it is used to build the UI for a specific website theme.

Part 2: Diving Deeper into React Features

1. Props and State in Action:

- Pass a username prop to the WelcomeBanner component and display it. Then, introduce a state variable using useState to track a user's input (like a search bar) and display its value.
- Example:

```
// src/components/WelcomeBanner.js
import React, { useState } from 'react';

const WelcomeBanner = ({ username }) => {
  const [search, setSearch] = useState('');

  return (
    <div>
      <h1>Welcome, {username}!</h1>
      <input
        type="text"
        placeholder="Search..."
        value={search}
        onChange={(e) => setSearch(e.target.value)}
      />
      <p>Searching for: {search}</p>
    </div>
  );
}

export default WelcomeBanner;
```

- Goal: Learn the difference between props and state and how to use them in components, especially in the context of a profile website.

2. Conditional Rendering and Event Handling :

- Implement a button in `WelcomeBanner` that toggles a state variable. Use this variable to conditionally render different elements in the component, like showing a login/logout button based on user state.
- Example:

```
// src/components/WelcomeBanner.js
import React, { useState } from 'react';

const WelcomeBanner = ({ username }) => {
  const [search, setSearch] = useState('');
  const [loggedIn, setLoggedIn] = useState(false);

  return (
    <div>
      <h1>Welcome, {username}!</h1>
      <input
        type="text"
        placeholder="Search..."
        value={search}
        onChange={(e) => setSearch(e.target.value)}
      />
      <p>Searching for: {search}</p>
      <button onClick={() => setLoggedIn(!loggedIn)}>
        {loggedIn ? 'Logout' : 'Login'}
      </button>
      {loggedIn && <p>You are logged in</p>}
    </div>
  );
}

export default WelcomeBanner;
```

- Goal: Practice conditional rendering and handling user interactions in React, with a focus on typical profile website interactions.









Instructions:

1. Write the required code in `src/components/WelcomeBanner.js`.
2. Open the terminal and navigate to the project directory.
3. Run the React project using:

```
npm start
```

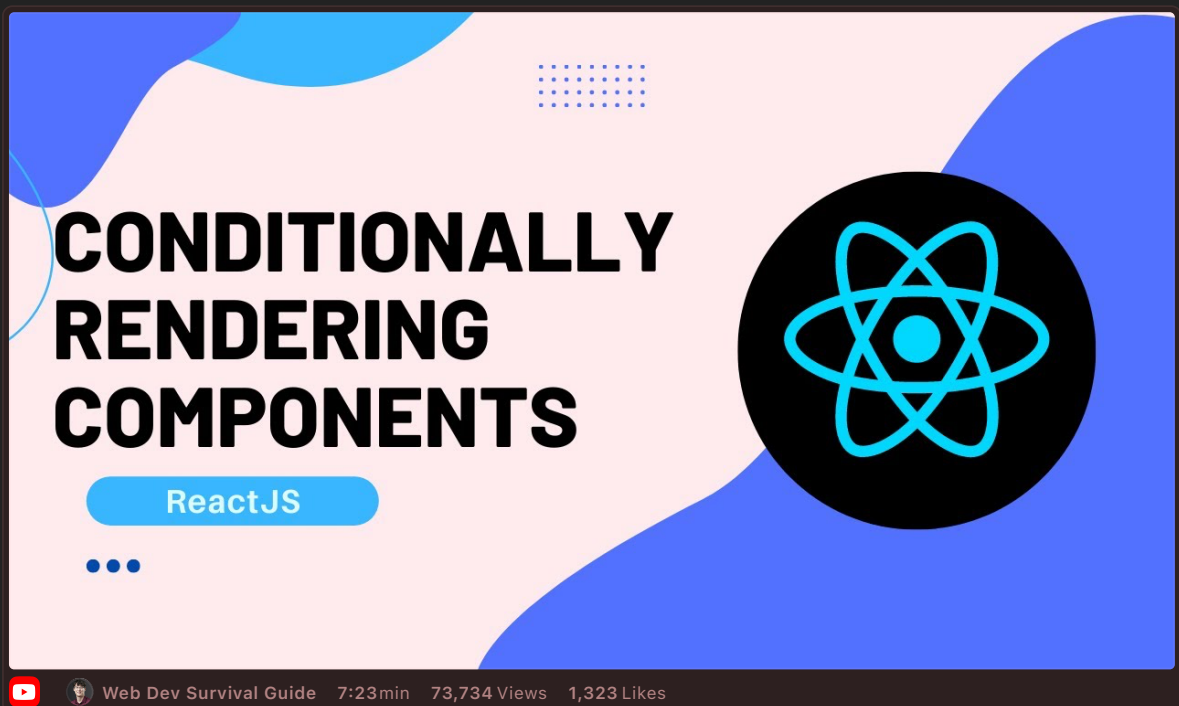
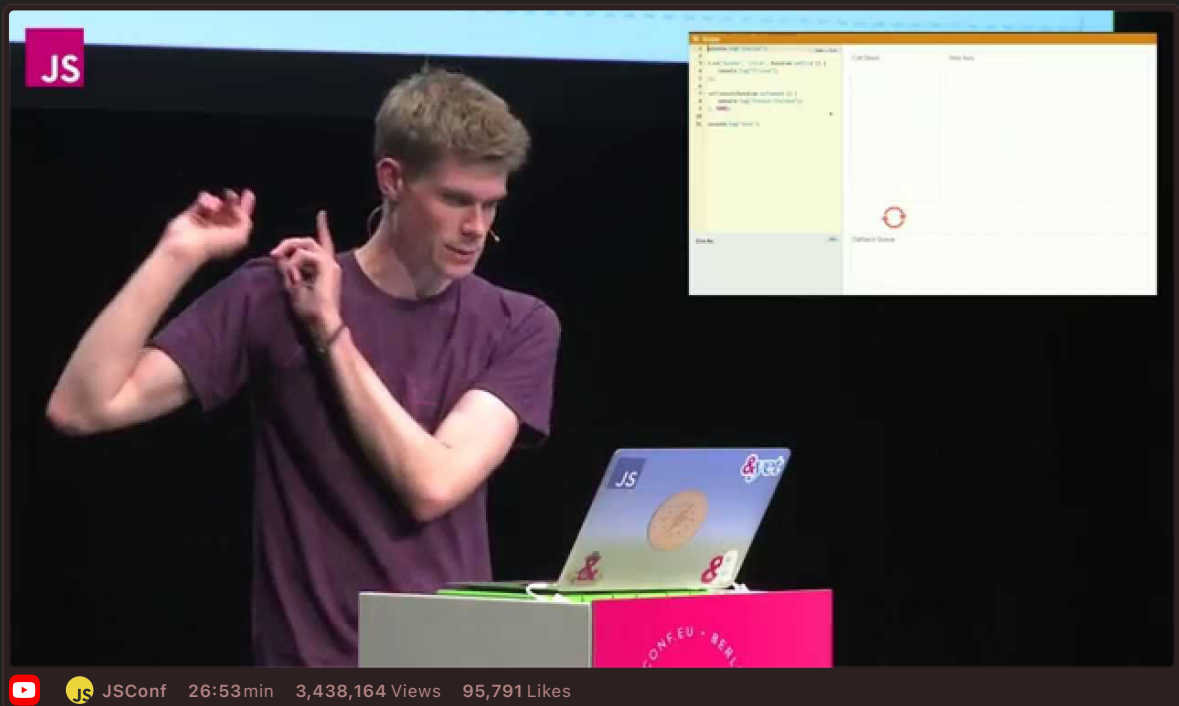
4. Open the project in a web browser to ensure the code displays correctly.
5. Use the browser's developer tools and React Developer Tools to debug and inspect the elements.

Resources:

- **Getting Started – React**
A JavaScript library for building user interfaces
 <https://reactjs.org/docs/getting-started.html>
- **Create React App**
Set up a modern web app by running one command.
 <https://create-react-app.dev/>
- **New React Developer Tools – React Blog**
This blog site has been archived. Go to react.dev/blog to see the recent posts. A month a...
 <https://reactjs.org/blog/2015/09/02/new-react-developer-tools.html>
-  **Documentation for Visual Studio Code**
Find out how to set-up and get the most from Visual Studio Code. Optimized for building and d...
 <https://code.visualstudio.com/docs>

Videos:

- 
  **Programming with Mosh** 1:20 hours 3,108,249 Views 55,795 Likes



GitHub Instructions:

1. Open in Visual Studio Code:

After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, VSCode will open the repository directly. If prompted, select "Open" or "Allow" to open the repository in VSCode.

2. Open the Terminal in VSCode:

In VSCode, open a terminal by selecting Terminal > New Terminal from the top menu.

3. Complete the Task:

In VSCode, write your solution in the `src/components/WelcomeBanner.js` file.

4. Run and Test Your Code:

Open your terminal, navigate to your project directory, and run:

```
npm start
```

5. Commit Your Changes:

In the VSCode terminal, add your changes to git:

```
git add src/components/WelcomeBanner.js
```

Commit your changes with a meaningful message:

```
git commit -m "Completed task 21"
```

6. Push Your Changes to Your Repository:

Push your changes to your forked repository:

```
git push origin main
```

7. Create a Pull Request:

Go to your repository on GitHub.

Click on the "Pull Requests" tab.

Click the "New Pull Request" button.

Ensure the base repository is the original template repository and the base branch is `main`.

Ensure the head repository is your forked repository and the compare branch is `main`.

Click "Create Pull Request".

Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Open in Visual Studio Code

# Open terminal in VSCode

# Complete the task by editing src/components/WelcomeBanner.js

# Navigate to the project directory
cd profile-website

# Run your code
npm start

# Add, commit, and push your changes
git add src/components/WelcomeBanner.js
git commit -m "Completed task 1"
git push origin main

# Create a pull request on GitHub
```