React Fundamentals - Getting Started with React in Vite, ES6 Prerequisites, JSX, and Components

Instructor: Pramod Kumar Jena

Session Overview

Topics Covered:

1. Introduction to React and Its Importance
2. Setting up the Development Environment with Vite
3. Installing and Configuring Tailwind CSS
4. ES6 Prerequisites for React
5. Understanding the Virtual DOM and Single-Page Application (SPA) Architecture
6. JSX Basics
7. Introduction to React Components

Goal: By the end of today's session, students will understand React's purpose, be comfortable with foundational ES6 concepts, and set up a React project in Vite with Tailwind CSS. They will also build a small introductory project to practice these concepts.

Session Outline

---

1. Introduction to React

- **What is React?**
  - Define React as a JavaScript library for building user interfaces.
  - Created by Facebook; widely adopted for its efficiency and flexibility.
- **Why Learn React?**
  - React's popularity in the industry.
  - Demand in job markets.
  - Key benefits: component-based structure, reactivity, and ease of integration with libraries.

---

2. Setting Up the Development Environment with Vite

- **Why Vite for React Projects?**

    - Explain Vite as a fast build tool for modern web projects.
    - Discuss benefits: quick start, hot-reloading, better performance compared to Create React App (CRA).
- **Steps for Installation in Vite**

Open a terminal and create a new project using the following command:

```
npm create vite@latest my-react-app -- --template react
cd my-react-app
npm install
```

    -

Run the development server:

```
npm run dev
```

    -
    - Open the application in the browser at the specified localhost URL.

---

3. Installing and Configuring Tailwind CSS

- **Why Tailwind CSS with React?**
    - Utility-first framework; perfect for building responsive UIs quickly.
- **Steps to Install Tailwind CSS in Vite**

Install Tailwind CSS:

```
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

    -
    - Configure Tailwind in the `tailwind.config.js` file.

Add Tailwind directives to `index.css`:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

    -

- Demonstrate adding some Tailwind classes in the `App.js` component to confirm the setup.

---

4. ES6 Prerequisites for React

Key ES6 Concepts to Understand for React:

Arrow Functions:

```
const greet = (name) => `Hello, ${name}`;
```

-

Destructuring:

```
const person = { name: 'Alice', age: 25 };
const { name, age } = person;
```

-

Spread and Rest Operators:

```
const numbers = [1, 2, 3];
const newNumbers = [...numbers, 4, 5];
```

-

Modules (import/export):

```
export const greet = () => 'Hello';
import { greet } from './greet.js';
```

-

Template Literals:

```
const greeting = `Hello, ${name}`;
```

-

---

5. Understanding Virtual DOM and Single-Page Application (SPA)

- **What is the Virtual DOM?**
    - Explain the concept of the virtual DOM as a lightweight representation of the actual DOM.
    - React updates the virtual DOM, identifies changes, and then efficiently updates the real DOM, improving performance.
- **Single-Page Application (SPA)**
    - SPAs load a single HTML page and dynamically update the content without full-page reloads.
    - Benefits: fast and responsive user experience.

---

6. Introduction to JSX

- **What is JSX?**
    - JSX stands for JavaScript XML, allowing HTML to be written directly within JavaScript.

Show a simple example of JSX syntax:

```
const element = <h1>Hello, World!</h1>;
```

    -
- **JSX and JavaScript Expression Embedding**

Demonstrate embedding JavaScript expressions in JSX with `{ }`:

```
const userName = 'John';
const greeting = <p>Hello, {userName}</p>;
```

    -

---

7. Introduction to Components

- **What is a Component?**
    - A component is a reusable, independent part of the UI in React.
    - Types: Functional and Class Components.
- **Building a Simple Functional Component**

Show the syntax for creating a basic functional component:

```
function Greeting() {
  return <h1>Hello, welcome to React!</h1>;
```

```
}
```

    -
    - **Example Project**: Build a `UserCard` component displaying user data (name and bio) using Tailwind CSS classes for styling.
- **Using Components in the App**
    - Demonstrate importing and using the `UserCard` component within the main `App` component.

Creating a Basic Profile Card

Objective: To reinforce today's learning by building a simple profile card that displays a user's name and bio.

1. **Instructions**:
    - Create a `ProfileCard` component in a new file (`ProfileCard.js`).
    - Style it with Tailwind CSS for a modern, card-based look.

Code:

```
import React from 'react';

const ProfileCard = ({ name, bio }) => {
  return (
    <div className="bg-blue-100 p-4 rounded-lg shadow-md">
      <h2 className="text-2xl font-bold">{name}</h2>
      <p className="text-gray-700">{bio}</p>
    </div>
  );
};

export default ProfileCard;
```

2.
3. **Rendering the Component**:
    - Import and render `ProfileCard` in the `App.js` file, passing in name and bio props.

Summary and Recap

- **React Fundamentals**: Introduction, Virtual DOM, JSX, and Components.
- **Vite Setup**: Vite and Tailwind installation.
- **ES6 Concepts**: Essential JavaScript ES6 features.
- **Mini Project**: A `ProfileCard` component to practice component creation, styling, and JSX.

Homework Assignment

1. Review ES6 concepts covered today and create a small file with examples of each.
2. Customize the `ProfileCard` component by adding more properties such as location, skills, or profile image.