React JS Notes

1. Introduction to React

- **React** is a powerful JavaScript library developed by Facebook for building fast, interactive, and scalable user interfaces.

- It follows a **component-based architecture**, which allows reusability and better code management.

- Primarily used for creating **Single Page Applications (SPAs)** that offer seamless user experiences without full page reloads.

---

2. MPA vs SPA

🔹 Multi-Page Application (MPA)

- Each user action loads a **new HTML page** from the server.

- Traditional approach used in older websites.

- Slower performance due to repeated full-page reloads.

- Example: Legacy e-commerce sites, banking portals.

🔹 Single Page Application (SPA)

- Loads a single HTML page and dynamically updates content as the user interacts.

- Faster, smoother navigation and modern UX.

- Example: Gmail, Instagram, Facebook.

---

3. Prerequisites

Before diving into React, ensure you are comfortable with:

- ✅ **HTML5**
- ✅ **CSS3**
- ✅ **JavaScript**
- ✅ **Node.js & npm** basics

---

4. Setup & Installation

Step 1: Install Node.js

- Download from: https://nodejs.org

Verify installation:

```
node -v
npm -v
```

-

Step 2: Create React App using Vite

- **Vite** is a next-generation build tool that is faster and lighter than Create React App.

```
# Create new project
npm create vite@latest my-app

# Choose framework:
# → React
# → Variant: JavaScript

cd my-app

# Install dependencies
npm install

# Run development server
npm run dev
```

- Open your project in the browser: http://localhost:5173/

---

5. React Components

React applications are built using **components**. Each component is an independent piece of UI.

🔹 Functional Component (Modern Approach)

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

export default Welcome;
```

🔹 Class Component (Older Approach)

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

class Welcome extends Component {
  render() {
    return <h1>Hello, Welcome to React!</h1>;
  }
}

export default Welcome;
```

📌 Today, **functional components with hooks** are the industry standard.

---

6. JSX (JavaScript XML)

- **JSX** is a syntax extension for JavaScript used in React.
- It allows writing **HTML-like code** inside JavaScript.
- Makes UI creation more readable and intuitive.

Example:

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

👉 JSX is not HTML – it gets transpiled to JavaScript by tools like **Babel**.

---

7. Git & GitHub Basics

🔹 What is Git?

- Git is a **version control system** that tracks changes in code.
- Helps in collaboration and rollback if something goes wrong.

🔹 What is GitHub?

- GitHub is a **cloud-based hosting platform** for Git repositories.
- Used to store, share, and collaborate on projects.

🔹 Common Git Commands

```
# Initialize git inside a project
git init

# Add files to staging area
git add .

# Commit changes
git commit -m "Initial commit"

# Add remote repository
git remote add origin https://github.com/username/repo.git

# Push code to GitHub
git push -u origin main
```

👉 Replace username/repo.git with your actual repository URL.
