1.Define SPA and its benefits

**SPA (Single-Page Application):**  
A **Single-Page Application** is a web application that loads a single HTML page and dynamically updates content as the user interacts with the app, without reloading the entire page.

**Benefits:**

**Faster performance** after initial load (no full-page reloads)

**Seamless user experience** similar to desktop apps

**Reduces server load** by fetching only required data

**Improved mobile performance**

**Easy to convert into Progressive Web Apps (PWAs)**

2.Define React and identify its working

ANS:

**React:**  
React is an open-source JavaScript library developed by Facebook used for building user interfaces, especially for SPAs. It uses components to create reusable UI pieces.

**How React Works:**

Uses **components** to build UI blocks.

Uses a **virtual DOM** to detect changes in the UI.

Updates only the parts of the DOM that actually changed (efficient rendering).

Supports **unidirectional data flow** (from parent to child).

React uses **JSX** (JavaScript + XML) to describe UI elements.

3.Identify the differences between SPA and MPA

ANS:

| **Feature** | **SPA (Single-Page Application)** | **MPA (Multi-Page Application)** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| Page Reloads | No full reloads (dynamic content) | Reloads entire page on each interaction |

|  |  |  |
| --- | --- | --- |
| Speed | Faster after first load | Slower, multiple requests per page |

|  |  |  |
| --- | --- | --- |
| User Experience | Smooth, app-like | Traditional website experience |

|  |  |  |
| --- | --- | --- |
| SEO | Harder to optimize without SSR | Easier for SEO |

|  |  |  |
| --- | --- | --- |
| Complexity | More client-side complexity | Simpler logic, but more backend work |

|  |  |  |
| --- | --- | --- |
| Examples | Gmail, Facebook, Twitter | Amazon, Wikipedia, eBay |

4.Explain Pros & Cons of Single-Page Application

ANS:

**Pros:**

Fast and responsive user experience

Smooth navigation (no page reloads)

Efficient data transfer (only fetch data, not HTML)

Ideal for mobile and dynamic web apps

**Cons:**

SEO limitations (needs extra setup like SSR or pre-rendering)

Initial load can be heavy

Harder to implement proper browser history and routing

JavaScript dependency (breaks if JS fails)

5.Explain about React

ANS:

React is a **component-based JavaScript library** for building fast, dynamic, and scalable user interfaces.

It was created by Facebook and is widely used in industry.

It promotes building **modular, reusable components**.

Supports **declarative programming**, where you describe what the UI should look like, and React takes care of the how.

React can be used for **web** (via ReactJS) and **mobile** (via React Native) development.

6.Define virtual DOM

ANS:

**Virtual DOM (VDOM):**  
The virtual DOM is a lightweight **in-memory representation of the real DOM** used by React.

**How it works:**

When the app state changes, React creates a **new virtual DOM**.

It **compares (diffs)** the new VDOM with the previous one.

React then **calculates the minimal set of changes** needed.

It **efficiently updates** only those parts of the actual DOM (real DOM).

This approach improves **performance** and makes UI rendering faster.

7.Explain Features of React

ANS:

**Component-Based Architecture** – Reusable and isolated building blocks

**Virtual DOM** – Efficient UI updates

**JSX (JavaScript XML)** – Write HTML inside JavaScript

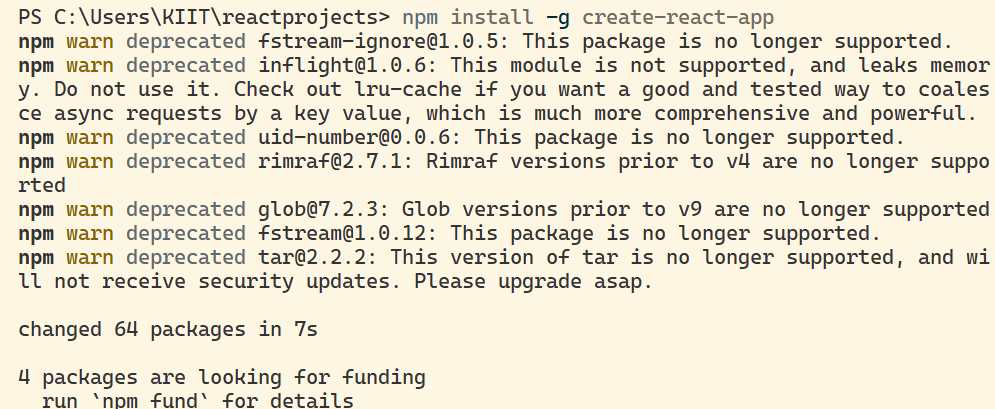
**One-Way Data Binding** – Predictable flow of data

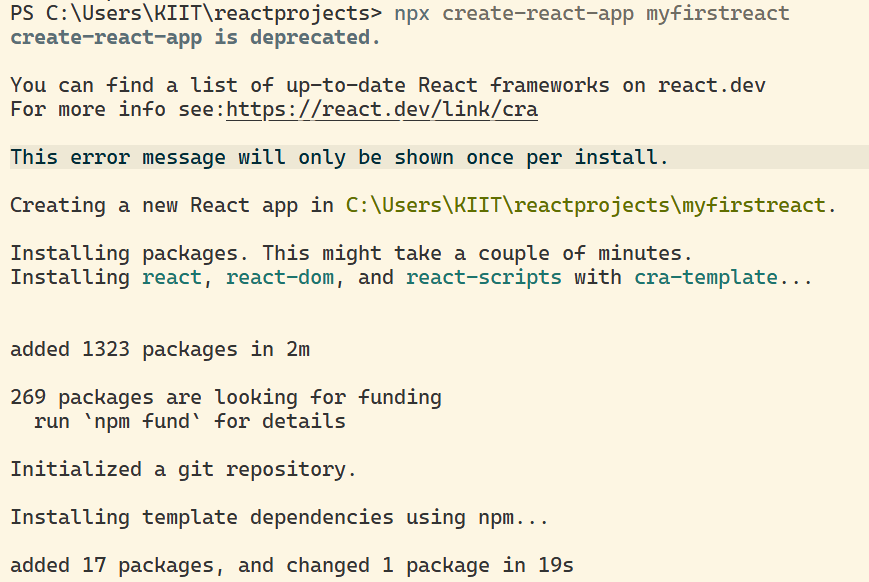
**React Hooks** – Manage state and side effects in functional components

**Rich Ecosystem** – Includes React Router, Redux, Next.js, etc.

**Fast Rendering** – Optimized rendering using virtual DOM

**Cross-Platform Development** – Via React Native for mobile







**CODE:**

import './App.css';

**function** App() {

  return (

     <h1>Welcome to the first session of React</h1>

  );

}

export default App;

