**Batch T7**

**Practical No. 5**

**Title of Assignment: Study and implementation of ReactJs**

**Student Name: Shreyash Patil**

**Student PRN: 22510039**

**Problem Statement 0: Basics of ReactJs**

Q. What is React and what problem does it solve?

* React is a JavaScript library for building user interfaces, primarily for single-page applications (SPAs). It allows developers to create reusable UI components and manage dynamic data effectively. React solves the problem of efficiently updating the DOM by using a virtual DOM, improving performance and scalability.

Q. What are React components and how are they used?

* React components are the building blocks of a React application. Each component is an independent, reusable piece of UI that can manage its own state and be composed together to create complex UIs. Components can be class-based or functional.

Q. What is JSX in React?

* JSX (JavaScript XML) is a syntax extension for JavaScript used in React. It allows developers to write HTML-like code within JavaScript, making it easier to build UI components.

Q. What are props in React and how do they differ from state?

* **Props:** Short for "properties," they are read-only inputs passed to components to configure them.
* **State:** The internal data management for a component that can change over time.
* **Key Differences:**
* Props are passed from parent to child components and are immutable.
* State is internal to a component and can change dynamically.

Q. What is state in React and how does it work?

* State represents data that changes over time in a React component. Each component can manage its own state using useState (in functional components) or this.state (in class components). When state updates, React re-renders the component.

Q. What are React lifecycle methods, and why are they important?

* Lifecycle methods are hooks in class components that allow you to run code at specific stages in a component's lifecycle (e.g., mounting, updating, unmounting). They are important for managing side effects and optimizing performance.
* **Key Lifecycle Methods:**
* componentDidMount()
* componentDidUpdate()
* componentWillUnmount()

Elaborate following with respect to ReactJs

o Event Handling

React uses synthetic events to handle user interactions. Event handlers are passed as functions, e.g., onClick, onChange.

o Conditional Rendering

Render components or elements conditionally using JavaScript logical operators like if or ternary expressions.

o Lists and Keys

When rendering lists, each item must have a unique "key" to help React identify which items have changed.

o Forms

Handling form inputs requires using onChange event handlers and updating state accordingly.

o Hooks

Hooks allow functional components to manage state and side effects. Common hooks include useState, useEffect, and useContext.

o React Router

React Router is used for navigation between different components or views in a single-page application.

o State Management

State can be managed globally using libraries like Redux or React Context API for more complex applications.

o React Context API

Context API allows you to share state across the entire app without passing props down manually.

Q. How can you optimize the performance of a React application?

**Use React.memo:** Prevent unnecessary re-renders for functional components.

* **Code Splitting:** Load code dynamically using React.lazy() and Suspense.
* **Use Virtualization:** Use libraries like react-window for rendering large lists.
* **Avoid Inline Functions:** Move functions outside render methods to prevent re-creation on every render.
* **Optimize State Management:** Avoid deep state nesting and update only necessary parts of the state.
* **Use** useCallback **and** useMemo**:** Memoize functions and values to avoid unnecessary recalculations.

Problem Statement 1: Star Wars character app

(In this problem statement, example of Star Wars is given, you may choose any characters

from the series of the movie like Harry Potter, etc. Every group in a batch will have

different characters.)