St. Francis Institute of Technology

(Engineering College)

An Autonomous Institute, Affiliated to University of Mumbai NAAC

A+ Accredited | CMPN, EXTC, INFT NBA Accredited | ISO 9001:2015 Certified

Department of Artificial Intelligence and Machine Learning

Academic Year: 2025-2026 **Term**: Even (Jan. 2026 – Jun. 2026) **Class / Branch**: SE – AIML
Semester: IV
Course: Web Programming Lab. (AI4VS_LR4)
Date of Assignment: / /2026 **Date of Submission:** / /2026

Pre-Lab Exercises for Experiment-9

Pre-Lab Activity 1: Understanding useEffect Execution

Task:

1. Create a functional component.
2. Display a simple message on screen.
3. Use `useEffect` to:
    - Print a message in the console when the component loads.

Pre-Lab Activity 2: Dependency Array Behavior

Task:

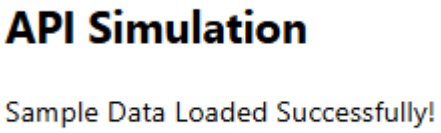
1. Create a counter using useState.
2. Add a useEffect that:
    - Logs “Counter Updated” whenever count changes.
3. Experiment with:
    - No dependency array
    - Empty dependency array
    - Dependency with count

Pre-Lab Activity 3: Simulating API Call using setTimeout

Before calling real APIs, simulate asynchronous behavior.

Task:

1. Create a component.
2. Create state called `data`.
3. Use useEffect to:
    - Simulate fetching data using setTimeout (2 seconds delay).
    - After delay, update state with sample data.
4. Show "Loading…" until data appears.



Pre-Lab Activity 4: Basic Fetch API (Without useEffect)

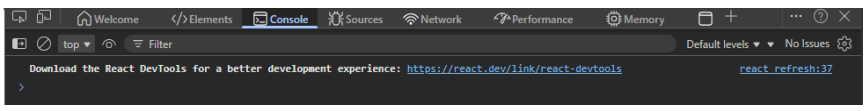
Task:

1. Create a button labeled "Fetch Data".
2. On button click:
    - Fetch data from a public API.
    - Print response in console.

Pre-Lab Activity 5: Async/Await Practice

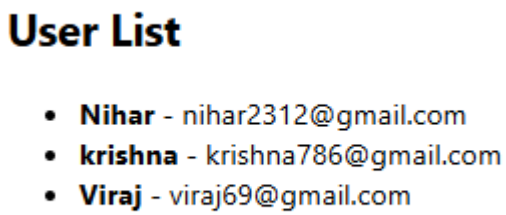
Task:

1. Create an async function.
2. Use async/await instead of .then().
3. Log fetched data in console.
4. Add try-catch for error handling.



Pre-Lab Activity 6: Rendering Array Data Dynamically Task:

1. Create an array of objects manually.
2. Store it in state.
3. Use map() to display:
    - Name
    - Email
4. Render items in list format.



Pre-Lab Record Questions

1 When does useEffect execute?

- `useEffect` executes **after the component renders**.
- By default (no dependency array), it runs **after every render**.
- With an **empty dependency array** `[]`, it runs **only once** when the component mounts.
- With a **dependency array** `[value]`, it runs whenever that specific value changes.
- It is mainly used for **side effects** like API calls, logging, timers, subscriptions, etc.
- It runs **after the DOM updates**, not during rendering.

---

2 What is the difference between `.then()` and `async/await`?

- `.then()` is a **promise-based method** used to handle asynchronous operations.

- `async/await` is a **modern syntax built on promises** that makes code look synchronous.
- `.then()` uses **chaining**, which can become harder to read with multiple calls.
- `async/await` makes code **cleaner and easier to understand**.
- Error handling in `.then()` uses `.catch()`.
- Error handling in `async/await` uses **try-catch block**.
- Both achieve the same result, but `async/await` improves readability.

3️⃣ Why is loading state important during API calls?

- API calls take time to fetch data.
- Loading state informs the user that **data is being fetched**.
- It improves **user experience**.
- Prevents the UI from appearing frozen.
- Helps avoid errors from accessing undefined data.
- Shows messages like **"Loading..."** until data arrives.

4️⃣ Why must fetched data be stored in state?

- React re-renders only when **state changes**.
- Storing fetched data in state updates the UI automatically.
- If data is not stored in state, it will not display properly.
- State ensures **data persistence inside the component**.
- It helps manage and render dynamic content.
- Without state, React cannot track changes in fetched data.