**Assignment 2**

**Web Engineering – Spring 2025**

|  |  |  |
| --- | --- | --- |
| Date: 2-June-2025 | Due Date: 10-June-2025 | Total Marks = 10 |
|  |  | Marks Obtained = |

Teacher Name**: Dr. Azhar Dilshad**

Instructions:

1. Submit the outcome of assignment solution in zip file in the assignments e folder.
2. Submit the scanned copy of your answers in a single PDF to your assignments e-folder.
3. Must write the file name as “yourname-CMSID-WebEngg-Assignemt2-Spring2025.PDF”.
4. Must write your name, CMS ID and WebEngg Assignment No. 2 in the title page.

Assignment Title: React Fundamentals and Application Development

Learning Objectives:

By the end of this assignment, students should be able to:

* + Understand and implement React components, props, and state.
  + Utilize React hooks effectively (useState, useEffect, useContext).
  + Fetch data from REST and GraphQL APIs.
  + Implement state management using Context API or Redux.
  + Build and document a small, functional React application.

Part 1: React Fundamentals

**Question 1:** Create three functional React components:

* + A Header component that receives a title via props.
  + A Counter component that uses local state to increment/decrement a value.
  + A UserCard component that displays user data (name, email, etc.) passed via props.

# Deliverables:

Code + Screenshots of rendered components.

Part 2: React Hooks

# Question 2:

* + Use useState to create a form with controlled inputs (name, age, email).
  + Use useEffect to log to the console every time the form is updated.
  + Use useContext to share theme information (e.g., dark/light mode) between components.

# Deliverables:

Code + Explanation of how each hook works and why it was used.

Part 3: API Fetching

# Question 3:

* + Fetch user data using a free REST API (e.g., https://jsonplaceholder.typicode.com/users) and display the first 5 users.
  + (Optional/Advanced) Use a GraphQL public API (e.g., https://countries.trevorblades.com/) to fetch and display country names and capitals.

# Deliverables:

Code + Screenshots of rendered data + Short paragraph explaining REST vs GraphQL.

Part 4: State Management

# Question 4:

* + Use **Context API or Redux** to manage global state for user authentication (mock login/logout).
  + Display login state across multiple components (e.g., Navbar, Profile).

# Deliverables:

Code + Diagram showing the state flow across components.

Part 5: Mini Project – Build a React App

# Question 5:

Build a small React application that includes:

* + A form for input
  + Data fetching from an API
  + State management (Context API or Redux)
  + At least 3 components and 1 custom hook (optional) Example apps:
  + Weather Dashboard
  + Simple Task Tracker
  + Country Info Viewer (REST/GraphQL)

# Deliverables:

* + Source code on GitHub
  + Demo video or screenshots
  + Brief documentation (README.md with features, setup instructions, and folder structure)

# Submission Guidelines:

* + Submit a single zipped folder containing:
    - Source code
    - README.md file
    - Screenshots or demo video link
    - GitHub repo link (optional but recommended)

**End of Assignment 2**