Frontend Assignment Set

Module 1 – Foundation

THEORY EXERCISE:

- What is a HTTP?
- What is a Browsers? How they works?
- What is Domain Name?
- What is hosting?

LAB EXERCISE: Create Github account & make repository.

Module 2 – Fundamentals of World Wide Web

THEORY EXERCISE:

- Difference between Web Designer and Web Developer
- What is a W3C?
- What is Domain?
- What SEO?
- What is SDLC life cycle?

LAB EXERCISE: Explain phases of SDLC life cycle.

Module 3 – Fundamentals of IT

What is a Program?

LAB EXERCISE: Write a simple "Hello World" program in two different programming languages ofyour choice. Compare the structure and syntax.

THEORY EXERCISE: Explain in your own words what a program is and how it functions.

What is Programming?

THEORY EXERCISE: What are the key steps involved in the programming process?

Types of Programming Languages

THEORY EXERCISE: What are the main differences between high-level and low-level programminglanguages?

World Wide Web & How Internet Works

LAB EXERCISE: Research and create a diagram of how data is transmitted from a client to a serverover the internet.

THEORY EXERCISE: Describe the roles of the client and server in web communication.

Network Layers on Client and Server

LAB EXERCISE: Design a simple HTTP client-server communication in any language.

THEORY EXERCISE: Explain the function of the TCP/IP model and its layers.

Client and Servers

THEORY EXERCISE: Explain Client Server Communication

Types of Internet Connections

LAB EXERCISE: Research different types of internet connections (e.g., broadband, fiber, satellite)and list their pros and cons.

THEORY EXERCISE: How does broadband differ from fiber-optic internet?

Protocols

LAB EXERCISE: Simulate HTTP and FTP requests using command line tools (e.g., curl).

THEORY EXERCISE: What are the differences between HTTP and HTTPS protocols?

Application Security

LAB EXERCISE: Identify and explain three common application security vulnerabilities. Suggestpossible solutions.

THEORY EXERCISE: What is the role of encryption in securing application, Software Applications and Its Types

LAB EXERCISE: Identify and classify 5 applications you use daily as either system software orapplication software.

THEORY EXERCISE: What is the difference between system software and application software?

Software Architecture

LAB EXERCISE: Design a basic three-tier software architecture diagram for a web application.

THEORY EXERCISE: What is the significance of modularity in software architecture?

Layers in Software Architecture

LAB EXERCISE: Create a case study on the functionality of the presentation, business logic, and dataaccess layers of a given software system.

THEORY EXERCISE: Why are layers important in software architecture?

Software Environments

LAB EXERCISE: Explore different types of software environments (development, testing, production).Set up a basic environment in a virtual machine.

THEORY EXERCISE: Explain the importance of a development environment in software production.

Source Code

LAB EXERCISE: Write and upload your first source code file to Github.

THEORY EXERCISE: What is the difference between source code and machine code?

Github and Introductions

LAB EXERCISE: Create a Github repository and document how to commit and push code changes.

THEORY EXERCISE: Why is version control important in software development?

Student Account in Github

LAB EXERCISE: Create a student account on Github and collaborate on a small project with aclassmate.

THEORY EXERCISE: What are the benefits of using Github for students?

Types of Software

LAB EXERCISE: Create a list of software you use regularly and classify them into the followingcategories: system, application, and utility software.

THEORY EXERCISE: What are the differences between open-source and proprietary software?

GIT and GITHUB Training

LAB EXERCISE: Follow a GIT tutorial to practice cloning, branching, and merging repositories.

THEORY EXERCISE: How does GIT improve collaboration in a software development team?

Application Software

LAB EXERCISE: Write a report on the various types of application software and how they improveproductivity.

THEORY EXERCISE: What is the role of application software in businesses?

Software Development Process

LAB EXERCISE: Create a flowchart representing the Software Development Life Cycle (SDLC).

THEORY EXERCISE: What are the main stages of the software development process?

Software Requirement

LAB EXERCISE: Write a requirement specification for a simple library management system.

THEORY EXERCISE: Why is the requirement analysis phase critical in software development?

Software Analysis

LAB EXERCISE: Perform a functional analysis for an online shopping system.

THEORY EXERCISE: What is the role of software analysis in the development process?

System Design

LAB EXERCISE: Design a basic system architecture for a food delivery app.

THEORY EXERCISE: What are the key elements of system design?

Software Testing

LAB EXERCISE: Develop test cases for a simple calculator program.

THEORY EXERCISE: Why is software testing important?

Maintenance

LAB EXERCISE: Document a real-world case where a software application required criticalmaintenance.

THEORY EXERCISE: What types of software maintenance are there?

Development

THEORY EXERCISE: What are the key differences between web and desktop applications?

Web Application

THEORY EXERCISE: What are the advantages of using web applications over desktop applications?

Designing

THEORY EXERCISE: What role does UI/UX design play in application development?

Mobile Application

THEORY EXERCISE: What are the differences between native and hybrid mobile apps?

DFD (Data Flow Diagram)

LAB EXERCISE: Create a DFD for a hospital management system.

THEORY EXERCISE: What is the significance of DFDs in system analysis?

Desktop Application

LAB EXERCISE: Build a simple desktop calculator application using a GUI library.

THEORY EXERCISE: What are the pros and cons of desktop applications compared to webapplications?

Flow Chart

LAB EXERCISE: Draw a flowchart representing the logic of a basic online registration system.

THEORY EXERCISE: How do flowcharts help in programming and system design?

Module 2 – Frontend- HTML

HTML Basics

Theory Assignment

- **Question 1**: Define HTML. What is the purpose of HTML in web development?
- **Question 2**: Explain the basic structure of an HTML document. Identify the mandatory tagsand their purposes.
- **Question 3**: What is the difference between block-level elements and inline elements in HTML? Provide examples of each.
- **Question 4**: Discuss the role of semantic HTML. Why is it important for accessibility andSEO? Provide examples of semantic elements.

Lab Assignment

Task:

Create a simple HTML webpage that includes:

⇒ A header (`<header>`), footer (`<footer>`), main section (`<main>`), and aside section (`<aside>`).

⇒ A paragraph with some basic text.

⇒ A list (both ordered and unordered).

⇒ A link that opens in a new tab.

HTML Forms

Theory Assignment

- **Question 1:** What are HTML forms used for? Describe the purpose of the input, textarea, select, and button elements.
- **Question 2:** Explain the difference between the GET and POST methods in form submission. When should each be used?
- **Question 3:** What is the purpose of the label element in a form, and how does it improve accessibility?

Lab Assignment

Task:

Create a contact form with the following fields:

⇒ Full name (text input)

⇒ Email (email input)

⇒ Phone number (tel input)

⇒ Subject (dropdown menu)

⇒ Message (textarea)

⇒ Submit button

Additional Requirements:

- ⇒ Use appropriate form validation using `required`, `minlength`, `maxlength`, and `pattern`.
- ⇒ Link form labels with their corresponding inputs using the `for` attribute.

HTML Tables

Theory Assignment

- **Question 1**: Explain the structure of an HTML table and the purpose of each of the following elements: `<table>`, `<tr>`, `<th>`, `<td>`, and `<thead>`.
- **Question 2**: What is the difference between `colspan` and `rowspan` in tables? Provide examples.
- **Question 3**: Why should tables be used sparingly for layout purposes? What is a better alternative?

Lab Assignment

Task:

Create a product catalog table that includes the following columns:

- ⇒ Product Name
- ⇒ Product Image (use placeholder image URLs)
- ⇒ Price
- ⇒ Description
- ⇒ Availability (in stock, out of stock)

Additional Requirements:

- ⇒ Use `thead` for the table header.
- ⇒ Add a border and some basic styling using inline CSS.
- ⇒ Use `colspan` or `rowspan` to merge cells where applicable.

Module 3 – Frontend – CSS and CSS3

CSS Selectors & Styling

Theory Assignment

- **Question 1**: What is a CSS selector? Provide examples of element, class, and ID selectors.
- **Question 2**: Explain the concept of CSS specificity. How do conflicts between multiple stylesget resolved?
- **Question 3**: What is the difference between internal, external, and inline CSS? Discuss the advantages and disadvantages of each approach.

Lab Assignment

Task:

Style the contact form (created in the HTML Forms lab) using external CSS. Thefollowing should be implemented:

⇒ Change the background color of the form.
⇒ Add padding and margins to form fields.
⇒ Style the submit button with a hover effect.
⇒ Use class selectors for styling common elements and ID selectors for uniqueelements.

CSS Box Model

Theory Assignment

- **Question 1**: Explain the CSS box model and its components (content, padding, border, margin). How does each affect the size of an element?
- **Question 2**: What is the difference between `border-box` and `content-box` box-sizing inCSS? Which is the default?

Lab Assignment

Task

Create a profile card layout using the box model. The profile card shouldinclude:

⇒ A profile picture.
⇒ The user's name and bio.
⇒ A button to "Follow" the user.

Additional Requirements:

⇒ Add padding and borders to the elements.
⇒ Ensure the layout is clean and centered on the page using CSS margins.
⇒ Use the `box-sizing` property to demonstrate both `content-box` and `border-box` on different elements.

CSS Flexbox

Theory Assignment

- **Question 1**: What is CSS Flexbox, and how is it useful for layout design? Explain the terms `flex-container` and `flex-item`.
- **Question 2**: Describe the properties `justify-content`, `align-items`, and `flex-direction` used in Flexbox.

Lab Assignment

Task

Create a simple webpage layout using Flexbox. The layout should include:

⇒ A header.
⇒ A sidebar on the left.
⇒ A main content area in the center.
⇒ A footer.

Additional Requirements:

⇒ Use Flexbox to position and align the elements.
⇒ Apply different `justify-content` and `align-items` properties to observe theireffects.
⇒ Ensure the layout is responsive, adjusting for smaller screens.

CSS Grid

Theory Assignment

- **Question 1**: Explain CSS Grid and how it differs from Flexbox. When would you use Grid over Flexbox?
- **Question 2**: Describe the `grid-template-columns`, `grid-template-rows`, and `grid-gap` properties. Provide examples of how to use them.

Lab Assignment

Task

Create a 3x3 grid of product cards using CSS Grid. Each card should contain:

⇒ A product image.
⇒ A product title.
⇒ A price.

Additional Requirements:

⇒ Use `grid-template-columns` to create the grid layout.
⇒ Use `grid-gap` to add spacing between the grid items.
⇒ Apply hover effects to each card for better interactivity.

Responsive Web Design with Media Queries

Theory Assignment

- **Question 1**: What are media queries in CSS, and why are they important for responsive design?
- **Question 2**: Write a basic media query that adjusts the font size of a webpage for screens smaller than 600px

Lab Assignment(Task)

Build a responsive webpage that includes:

⇒ A navigation bar.

⇒ A content section with two columns.

⇒ A footer.

Additional Requirements:

⇒ Use media queries to make the webpage responsive for mobile devices.

⇒ On smaller screens (below 768px), stack the columns vertically.

⇒ Adjust the font sizes and padding to improve readability on mobile.

Typography and Web Fonts

Theory Assignment

- **Question 1**: Explain the difference between web-safe fonts and custom web fonts. Whymight you use a web-safe font over a custom font?
- **Question 2**: What is the `font-family` property in CSS? How do you apply a custom GoogleFont to a webpage?

Lab Assignment

Task

Create a blog post layout with the following:

⇒ A title, subtitle, and body content.

⇒ Use at least two different fonts (one for headings, one for body content).

⇒ Style the text to be responsive and easy to read.

Additional Requirements:

⇒ Use a custom font from Google Fonts.

⇒ Adjust line-height, font-size, and spacing for improved readability.

Module 5 – Frontend – HTML5

Theory Assignment

- **Question 1**: Difference b/w HTML & HTML5?
- **Question 2**: What are the additional tags used in HTML5?

Lab Assignment(Task)

Create a audio video tag

⇒ Also applied properties like muted loop autoplay
⇒ Create some shape using canvas tag in html
⇒ Create some shape using svg tag in html

Module 8) JavaScript

JavaScript Introduction

Theory Assignment

- **Question 1**: What is JavaScript? Explain the role of JavaScript in web development.
- **Question 2**: How is JavaScript different from other programming languages like Python orJava?
- **Question 3**: Discuss the use of `<script>` tag in HTML. How can you link an external JavaScript file to an HTML document?

Lab Assignment(Task)

⇒ Create a simple HTML page and add a `<script>` tag within the page.

⇒ Write JavaScript code to display an alert box with the message "Welcome toJavaScript!" when the page loads.

Variables and Data Types

Theory Assignment

- **Question 1**: What are variables in JavaScript? How do you declare a variable using `var`, `let`,and `const`?
- **Question 2**: Explain the different data types in JavaScript. Provide examples for each.
- **Question 3**: What is the difference between `undefined` and `null` in JavaScript?

Lab Assignment (Task)

⇒ Write a JavaScript program to declare variables for different data types (string,number, boolean, null, and undefined).

⇒ Log the values of the variables and their types to the console using `console.log()`.

JavaScript Operators

Theory Assignment

- **Question 1**: What are the different types of operators in JavaScript? Explain with examples.
  - Arithmetic operators
  - Assignment operators
  - Comparison operators
  - Logical operators
- **Question 2**: What is the difference between == and === in JavaScript?

Lab Assignment(Task)

Create a JavaScript program to perform the following:

- ⇒ Add, subtract, multiply, and divide two numbers using arithmetic operators.
- ⇒ Use comparison operators to check if two numbers are equal and if onenumber is greater than the other.
- ⇒ Use logical operators to check if both conditions (e.g., `a > 10` and `b < 5`)are true.

Control Flow (If-Else, Switch)

Theory Assignment

- **Question 1**: What is control flow in JavaScript? Explain how `if-else` statements work withan example.
- **Question 2**: Describe how `switch` statements work in JavaScript. When should you use a `switch` statement instead of `if-else`?

Lab Assignment

- ⇒ Task 1:

  Write a JavaScript program to check if a number is positive, negative, or zero usingan `if-else` statement.
- ⇒ Task 2:

  Create a JavaScript program using a `switch` statement to display the day of theweek based on the user input (e.g., 1 for Monday, 2 for Tuesday, etc.).

Loops (For, While, Do-While)

Theory Assignment

- **Question 1**: Explain the different types of loops in JavaScript (for, while, do-while). Provide abasic example of each.
- **Question 2**: What is the difference between a `while` loop and a `do-while` loop?

Lab Assignment

- ⇒ Task 1:

  Write a JavaScript program using a `for` loop to print numbers from 1 to 10.
- ⇒ Task 2:

  Create a JavaScript program that uses a `while` loop to sum all even numbers between 1 and 20.
- ⇒ Task 3:

  Write a `do-while` loop that continues to ask the user for input until they enter a number greater than 10.

Functions

Theory Assignment

- **Question 1**: What are functions in JavaScript? Explain the syntax for declaring and calling a function.
- **Question 2**: What is the difference between a function declaration and a function expression?
- **Question 3**: Discuss the concept of parameters and return values in functions.

Lab Assignment

⇒ Task 1:

Write a function `greetUser` that accepts a user's name as a parameter and displaysa greeting message (e.g., "Hello, John!").

⇒ Task 2:

Create a JavaScript function `calculateSum` that takes two numbers as parameters,adds them, and returns the result.

Arrays

Theory Assignment

- **Question 1**: What is an array in JavaScript? How do you declare and initialize an array?
- **Question 2**: Explain the methods `push()`, `pop()`, `shift()`, and `unshift()` used in arrays.

Lab Assignment

⇒ Task 1:

- Declare an array of fruits (`["apple", "banana", "cherry"]`). Use JavaScript to:
- Add a fruit to the end of the array.
- Remove the first fruit from the array.
- Log the modified array to the console.

⇒ Task 2:

- Write a program to find the sum of all elements in an array of numbers.

Objects

Theory Assignment

- **Question 1**: What is an object in JavaScript? How are objects different from arrays?
- **Question 2**: Explain how to access and update object properties using dot notation and bracket notation.

Lab Assignment

Task:

⇒ Create a JavaScript object `car` with properties `brand`, `model`, and `year`. UseJavaScript to:

- Access and print the car's brand and model.
- Update the `year` property.
- Add a new property `color` to the car object.

JavaScript Events

Theory Assignment

- **Question 1**: What are JavaScript events? Explain the role of event listeners.
- **Question 2**: How does the `addEventListener()` method work in JavaScript? Provide an example.

Lab Assignment

Task

⇒ Create a simple webpage with a button that, when clicked, displays an alert saying"Button clicked!" using JavaScript event listeners.

DOM Manipulation

Theory Assignment

- **Question 1**: What is the DOM (Document Object Model) in JavaScript? How does JavaScript interact with the DOM?
- **Question 2**: Explain the methods `getElementById()`, `getElementsByClassName()`,and `querySelector()` used to select elements from the DOM.

Lab Assignment

Task:

⇒ Create an HTML page with a paragraph (`<p>`) that displays "Hello, World!".

⇒ Use JavaScript to:

- Change the text inside the paragraph to "JavaScript is fun!".
- Change the color of the paragraph to blue.

JavaScript Timing Events (setTimeout, setInterval)

Theory Assignment

- **Question 1**: Explain the `setTimeout()` and `setInterval()` functions in JavaScript. Howare they used for timing events?
- **Question 2**: Provide an example of how to use `setTimeout()` to delay an action by 2 seconds.

Lab Assignment

⇒ Task 1:

- Write a program that changes the background color of a webpage after 5 secondsusing `setTimeout()`.

⇒ Task 2:

- Create a digital clock that updates every second using `setInterval()`.

JavaScript Error Handling

Theory Assignment

- **Question 1**: What is error handling in JavaScript? Explain the `try`, `catch`, and `finally` blocks with an example.
- **Question 2**: Why is error handling important in JavaScript applications?

Lab Assignment

Task:

- Write a JavaScript program that attempts to divide a number by zero. Use `try-catch` to handle the error and display an appropriate error message.

Module 9- Introduction to React.js

THEORY EXERCISE

- **Question 1**: What is React.js? How is it different from other JavaScript frameworks and libraries?
- **Question 2**: Explain the core principles of React such as the virtual DOM and component-based architecture.
- **Question 3**: What are the advantages of using React.js in web development?

LAB EXERCISE

- Task:
  - Set up a new React.js project using `create-react-app`.
  - Create a basic component that displays "Hello, React!" on the web page.

JSX (JavaScript XML)

THEORY EXERCISE

- **Question 1**: What is JSX in React.js? Why is it used?
- **Question 2**: How is JSX different from regular JavaScript? Can you write JavaScript insideJSX?
- **Question 3**: Discuss the importance of using curly braces `{}` in JSX expressions.

LAB EXERCISE

- Task:
  - Create a React component that renders the following JSX elements:
    - A heading with the text "Welcome to JSX".
    - A paragraph explaining JSX with dynamic data (use curly braces to insert variables).

Components (Functional & Class Components)

THEORY EXERCISE

- **Question 1**: What are components in React? Explain the difference between **functional components** and **class components**.
- **Question 2**: How do you pass data to a component using props?
- **Question 3**: What is the role of `render()` in class components?

LAB EXERCISE

- Task 1:
  - Create a functional component `Greeting` that accepts a `name` as a prop and displays "Hello, [name]!".
- Task 2:
  - Create a class component `WelcomeMessage` that displays "Welcome to React!" anda `render()` method.

Props and State

THEORY EXERCISE

- **Question 1**: What are props in React.js? How are props different from state?
- **Question 2**: Explain the concept of state in React and how it is used to manage component data.
- **Question 3**: Why is `this.setState()` used in class components, and how does it work?

LAB EXERCISE

- Task 1:
  - Create a React component `UserCard` that accepts `name`, `age`, and `location` as props and displays them in a card format.
- Task 2:
  - Create a `Counter` component with a button that increments a count value using React state. Display the current count on the screen.

Handling Events in React

THEORY EXERCISE

- **Question 1**: How are events handled in React compared to vanilla JavaScript? Explain the concept of synthetic events.
- **Question 2**: What are some common event handlers in React.js? Provide examples of `onClick`, `onChange`, and `onSubmit`.
- **Question 3**: Why do you need to bind event handlers in class components?

LAB EXERCISE

- Task 1:
  - Create a button in a React component that, when clicked, changes the text from "Not Clicked" to "Clicked!" using event handling.
- Task 2:
  - Create a form with an input field in React. Display the value of the input field dynamically as the user types in it.

Conditional Rendering

THEORY EXERCISE

- **Question 1**: What is conditional rendering in React? How can you conditionally render elements in a React component?
- **Question 2**: Explain how `if-else`, ternary operators, and `&&` (logical AND) are used in JSX for conditional rendering.

LAB EXERCISE

- Task 1:
  - Create a component that conditionally displays a login or logout button based on the user's login status.
- Task 2:
  - Implement a component that displays a message like "You are eligible to vote" if the user is over 18, otherwise display "You are not eligible to vote."

Lists and Keys

THEORY EXERCISE

- **Question 1**: How do you render a list of items in React? Why is it important to use keys when rendering lists?
- **Question 2**: What are keys in React, and what happens if you do not provide a unique key?

LAB EXERCISE

- Task 1:
  - Create a React component that renders a list of items (e.g., a list of fruit names). Use the `map()` function to render each item in the list.
- Task 2:
- Create a list of users where each user has a unique `id`. Render the user list using React and assign a unique `key` to each user.

Forms in React

THEORY EXERCISE

- **Question 1**: How do you handle forms in React? Explain the concept of controlled components.
- **Question 2**: What is the difference between controlled and uncontrolled components in React?

LAB EXERCISE

- Task 1:
  - Create a form with inputs for `name`, `email`, and `password`. Use state to control the form and display the form data when the user submits it.
- Task 2:
  - Add validation to the form created above. For example, ensure that the `email` input contains a valid email address.

Lifecycle Methods (Class Components)

THEORY EXERCISE

- **Question 1**: What are lifecycle methods in React class components? Describe the phases of a component's lifecycle.

- **Question 2**: Explain the purpose of `componentDidMount()`, `componentDidUpdate()`,and `componentWillUnmount()`.

LAB EXERCISE

- Task 1:
  - Create a class component that fetches data from an API when the component mounts using `componentDidMount()`. Display the data in the component.
- Task 2:
  - Implement a component that logs a message to the console when it updates using `componentDidUpdate()`. Log another message when the component unmounts using `componentWillUnmount()`.

Hooks (useState, useEffect, useReducer, useMemo, useRef, useCallback)

THEORY EXERCISE

- **Question 1**: What are React hooks? How do `useState()` and `useEffect()` hooks work in functional components?
- **Question 2**: What problems did hooks solve in React development? Why are hooks considered an important addition to React?
- **Question 3**: What is useReducer ? How we use in react app?
- **Question 4:** What is the purpose of useCallback & useMemo Hooks?
- **Question 5:** What's the Difference between the useCallback & useMemo Hooks?
- **Question 6 :** What is useRef ? How to work in react app?

LAB EXERCISE

- Task 1:
  - Create a functional component with a counter using the `useState()` hook. Include buttons to increment and decrement the counter.
- Task 2:
  - Use the `useEffect()` hook to fetch and display data from an API when the component mounts.
- Task 3:
  - Create react app with use of useSelector & useDispatch.

- Task 4:
  - Create react app to avoid re-renders in react application by useRef?

Routing in React (React Router)

THEORY EXERCISE

- **Question 1**: What is React Router? How does it handle routing in single-page applications?
- **Question 2**: Explain the difference between `BrowserRouter`, `Route`, `Link`, and `Switch` components in React Router.

LAB EXERCISE

- Task 1:
    - Set up a basic React Router with two routes: one for a Home page and one for an About page. Display the appropriate content based on the URL.
- Task 2:
    - Create a navigation bar using React Router's `Link` component that allows users to switch between the Home, About, and Contact pages.

React – JSON-server and Firebase Real Time Database

THEORY EXERCISE

- **Question 1:** What do you mean by RESTful web services?
- **Question 2:** What is Json-Server? How we use in React ?
- **Question 3**: How do you fetch data from a Json-server API in React? Explain the role of `fetch()` or `axios()` in making API requests.
- **Question 4:** What is Firebase? What features does Firebase offer?
- **Question 5**: Discuss the importance of handling errors and loading states when working with APIs in React

LAB EXERCISE

- Task 1:
    - Create a React component that fetches data from a public API (e.g., a list of users)and displays it in a table format.
    - Create a React app with Json-server and use Get , Post , Put , Delete & patch method on Json-server API.
- Task 2:
    - Create a React app crud and Authentication with firebase API.
    - Implement google Authentication with firebase API.

- Task 3:
    - Implement error handling and loading states for the API call. Display a loading spinner while the data is being fetched.

Context API

THEORY EXERCISE

- **Question 1**: What is the Context API in React? How is it used to manage global state across multiple components?
- **Question 2**: Explain how `createContext()` and `useContext()` are used in React for sharing state.

LAB EXERCISE

- Task 1:
    - Create a simple theme toggle (light/dark mode) using the Context API. The themestate should be shared across multiple components.
- Task 2:
    - Use the Context API to create a global user authentication system. If the user is logged in, display a welcome message; otherwise, prompt them to log in.

State Management (Redux, Redux-Toolkit or Recoil)

THEORY EXERCISE

- **Question 1**: What is Redux, and why is it used in React applications? Explain the core concepts of actions, reducers, and the store.
- **Question 2**: How does Recoil simplify state management in React compared to Redux?

LAB EXERCISE

- Task 1:
    - Create a simple counter application using Redux for state management. Implement actions to increment and decrement the counter.
- Task 2:
    - Build a todo list application using Recoil for state management. Allow users to add, remove, and mark tasks as complete.
- Task 3:
    - Build a crud application using Redux-Toolkit for state management. Allow users to add, remove, delete and update.