Training Document

React JS & React Native (Beginner Level)

🎯 Objective

This document is designed to learn React JS (Web) and React Native (Mobile) from scratch.

🌐 PART 1: React JS (Web Development)

1️⃣ Prerequisites JavaScript Tutorial

Video Tutorial: https://www.youtube.com/watch?v=W6NZfCO5SIk

should have knowledge of:

- HTML
- CSS
- JavaScript basics

2️⃣ React JS Fundamentals React Tutorial

Video Tutorial: https://www.youtube.com/watch?v=Ke90Tje7VS0

Topics to be covered:

- What is React & why use it
- Setting up React project (Vite) Build a React app from Scratch – React
- JSX syntax
- Components
- Props
- State
- Event handling
- Conditional rendering
- Lists & keys

3️⃣ React JS Intermediate Concepts

- Hooks
- Form handling & validation
- Controlled components

- Component reusability
- Basic folder structure
- React Router (basic navigation)

4️⃣ React JS Beginner Projects

1. Counter App

This project allows users to increase, decrease, and reset a numeric value using buttons. The counter updates instantly on user interaction without refreshing the page. will learn how to manage component state, handle button click events, and update the UI dynamically.

---

2. To-Do List

The To-Do List application enables users to add new tasks, mark tasks as completed, and delete tasks when they are no longer needed. The task list updates in real time based on user actions. This project helps understand list rendering, conditional UI changes, and state management.

---

3. Notes App

The Notes App allows users to create, edit, and delete notes within the application. Notes are stored locally in the application state. This project focuses on handling text input, managing multiple data entries, and building reusable components.

---

4. Theme Toggle App

This application allows users to switch between light mode and dark mode. The theme change updates the background colors and text styles across the application. will learn how to apply conditional styling and manage global UI state.

---

5. Calculator App

The Calculator App performs basic arithmetic operations such as addition, subtraction, multiplication, and division. Users can input numbers using buttons and see results instantly. This project strengthens logic building, event handling, and state updates.

---

6. Quiz App

The Quiz App displays a set of predefined questions with multiple-choice options. Users select answers and receive a final score at the end of the quiz. This project helps practice conditional rendering, managing user selections, and implementing simple logic for score calculation.

---

7. Chat UI

The Chat UI project simulates a messaging interface where users can type and display messages in a chat layout. Messages are shown locally without real-time backend communication. This project will teach you how to set layout design, list rendering, and UI updates based on user input.

---

8. Image Gallery

The Image Gallery displays a collection of locally stored images in a grid or list format. Users can view images and navigate through them. This project focuses on component structure, styling, and rendering visual content efficiently.

---

9. Students Management System

This application allows users to add, view, edit and delete and manage student details such as name, roll number, and class. The data is stored locally in the application state.

Got it! You want the **Weather App description formatted like your Students Management System entry (#9)**—concise, clean, and structured. Here's the revised version:

---

10. Weather App Using ReactJS

This application allows users to enter the name of any city and fetch the current weather information for that location. The displayed details include temperature (in Celsius), weather description (sunny, cloudy, rain, etc.), and humidity. The app uses a free public API that does not require an API key, making it easy for learning and demonstration purposes.

Features:

- Enter city name to get current weather.
- Displays temperature in Celsius.
- Shows weather description (sunny, cloudy, rain, etc.).
- Shows humidity level.
- Free to use, no API key required.

API Used:

- **API Name:** wttr.in
- **API URL Format:** https://wttr.in/{CITY}?format=j1
- **Example:** https://wttr.in/London?format=j1
- **Response:** JSON data containing temperature, weather description, and humidity.

- **fetch** is a built-in JavaScript function to request data from a server or API. It returns a **promise** which resolves to the response.

Usage in React:
The app uses the fetch function to retrieve weather data from the API when a user enters a city name and presses Enter or clicks Submit button. The JSON response is then parsed and displayed in a user-friendly format.

---

5 Learning Outcomes (React JS)

- Understanding component-based architecture
- Managing state and props
- Building interactive UIs
- Handling user events

🌐 PART 2: React Native (Mobile Development) Set Up Your Environment · React Native

Video Tutorial: https://www.youtube.com/watch?v=Hf4MJH0jDb4

1️⃣ Prerequisites

should have knowledge of:

- JavaScript basics
- React JS fundamentals (components, state, props)
- Basic HTML/CSS concepts (for layout understanding)

2️⃣ React Native Fundamentals React Native · Learn once, write anywhere

Topics to be covered:

- What is React Native & why use it
- Setting up React Native project (npx react-native init)
- Components (View, Text, TextInput, Button, Image)
- Props and State
- Event handling
- Styling with StyleSheet
- Conditional rendering
- Lists (FlatList) & keys

3️⃣ React Native Intermediate Concepts

- Hooks
- Handling forms and inputs
- Controlled components
- Component reusability
- Navigation using **React Navigation**
- Fetching data from APIs

4️⃣ React Native Beginner Projects

1. **Counter App**
   This project allows users to increase, decrease, and reset a numeric value using buttons. This will help learn state management, event handling, and UI updates in a mobile environment.

2. **To-Do List**
   This application enables users to add new tasks, mark tasks as completed, and delete tasks. Helps understand list rendering, conditional UI updates, and managing state in mobile apps.

3. **Notes App**
   Allows creating, editing, and deleting notes. will handle text inputs, manage multiple data entries, and build reusable components.

4. **Theme Toggle App**
   Allows users to switch between light mode and dark mode. The theme change updates background colors and text styles across the app.will learn conditional styling and global UI state management.

5. **Calculator App**
   Performs basic arithmetic operations such as addition, subtraction, multiplication, and division. Users input numbers using buttons and see results instantly. This project strengthens logic building, event handling, and state updates.

6. **Quiz App**
   Displays a set of predefined questions with multiple-choice options. Users select answers and receive a final score at the end of the quiz. Helps practice conditional rendering, managing user selections, and implementing simple score calculation logic.

7. **Chat UI**
   Simulates a messaging interface where users can type and display messages in a chat layout. Messages are shown locally without real-time backend communication. Teaches layout design, list rendering, and UI updates based on user input.

8. **Image Gallery**
   Displays a collection of locally stored images in a grid or list format. Users can view images and navigate through them. Focuses on component structure, styling, and rendering visual content efficiently.

9. **Students Management System**
   Allows users to add, view, edit, and delete student details such as name, roll number, and class. Data is stored locally in the app state. learn CRUD operations, state management, and component reusability.

10. **Weather App Using React Native**

This application allows users to enter the name of any city and fetch the current weather information for that location. The displayed details include temperature (in Celsius), weather description (sunny, cloudy, rain, etc.), and humidity. The app uses a free public API that does not require an API key.

Features:

- Enter city name to get current weather
- Displays temperature in Celsius

- Shows weather description (sunny, cloudy, rain, etc.)
- Shows humidity level
- Free to use, no API key required

API Used:

- API Name: wttr.in
- API URL Format: https://wttr.in/{CITY}?format=j1
- Example: https://wttr.in/London?format=j1
- Response: JSON data containing temperature, weather description, and humidity

fetch in React Native:
fetch is a built-in JavaScript function to request data from a server or API. It returns a promise which resolves to the response.

Usage in React Native:

- Use fetch to retrieve weather data from the API when a user enters a city name and presses the Submit button
- Parse the JSON response
- Display the weather data using Text components inside View
- Example structure includes TextInput for city, Button for search, and a View to display results

---

5️⃣ Learning Outcomes (React Native)

- Understanding mobile app component structure
- Managing state and props in a mobile environment
- Building interactive UIs for mobile devices
- Handling user events and inputs
- Fetching and displaying data from APIs
- Implementing basic CRUD and interactive projects in mobile apps