

Industrial Training Daily Diary

MERN Stack Training

Company: Sensation Software Solutions

Student Name: Gurvinder Singh

Training Duration: 6 Months

Day: 17

Day 17 – State Management in React & Introduction to Redux

Objective of the Day

The objective of Day 17 was to understand the **state management problem in React**, the limitations of props and local state, and to get an **introduction to Redux** as a centralized state management solution used in large-scale applications.

Understanding State Management in React

The trainer explained that as React applications grow, managing state across multiple components becomes complex. Passing data through multiple levels using props leads to problems in maintainability.

Key issues discussed:

- Props drilling
 - Difficulty in managing shared state
 - Tight coupling between components
-

Props vs State Revisited

We revised the difference between props and state:

- Props are read-only and passed from parent to child
- State is mutable and managed within components
- Excessive prop passing makes code complex

This revision helped in understanding the need for better state management solutions.

Introduction to Redux

The trainer introduced **Redux** as a predictable state container for JavaScript applications.

Main concepts explained:

- Single source of truth
- Global application state
- Centralized data flow

Redux helps manage shared state efficiently across the entire application.

Core Concepts of Redux

The following Redux concepts were introduced:

- Store – holds the global state
- Actions – describe what happened
- Reducers – define how state changes
- Dispatch – sends actions to reducers

These concepts were explained using simple flow diagrams and examples.

Redux Data Flow

We learned about the unidirectional data flow in Redux:

Action → Dispatch → Reducer → Store → UI

This predictable flow makes debugging and testing easier.

Practical Demonstration

The trainer demonstrated:

- Creating a Redux store
- Writing basic actions and reducers
- Connecting Redux with a React application

This demo helped us understand how Redux integrates with React.

Weekly Task – Week 4 (Assigned)

As part of the weekly task, we were assigned to:

- Study Redux core concepts
- Create a small demo for global state management
- Compare Context API and Redux
- Update diary and code on GitHub Pages

Importance in MERN Stack Development

Redux is widely used in MERN stack applications to manage complex state and ensure data consistency across components.

Conclusion of Day 17

Day 17 gave me a clear understanding of why state management is important and how Redux helps solve common problems in large React applications.

Outcome of the Day:

- Understanding of state management challenges
- Basic knowledge of Redux
- Ability to explain Redux data flow
- Preparation for advanced R