

Industrial Training Daily Diary

MERN Stack Training

Company: Sensation Software Solutions

Student Name: Tajinder Kaur

Training Duration: 6 Months

Day: 29

Day 29 – Redux with React Hooks: useSelector, useDispatch & Middleware Enhancements

Objective of the Day

The objective of Day 29 was to learn **React-Redux hooks** for managing state, using `useSelector` and `useDispatch`, and enhancing middleware for better state and API management.

useSelector & useDispatch Hooks

Trainer explained:

- `useSelector` to select and access Redux state in functional components
- `useDispatch` to dispatch actions from components
- Advantages over `connect` HOC: simpler, cleaner code, easier to manage functional components

Example: Selecting counter value and dispatching increment/decrement actions.

Middleware Enhancements

Key points covered:

- Using Redux Thunk for async API calls
 - Combining multiple middleware for logging, error handling, and async calls
 - Understanding the order of middleware execution
 - Practical demonstration with API requests
-

Practical Work Done

During lab session:

- Converted previous forms to use React-Redux hooks
- Replaced `connect` with `useSelector` and `useDispatch`
- Enhanced middleware by adding logging of actions and API responses
- Tested async form submissions and observed middleware handling

Hands-on practice helped in understanding modern Redux patterns with hooks.

Weekly Task – Week 7 (Advanced)

Tasks included:

- Update all forms to use Redux hooks
 - Implement enhanced middleware logging and error handling
 - Test async API calls using Redux hooks
 - Push code and diary updates to GitHub Pages
-

Importance in MERN Stack Development

Using Redux hooks simplifies state management in functional components, improves code readability, and allows better control over middleware, enhancing the scalability of MERN applications.

Conclusion of Day 29

Day 29 gave practical knowledge of React-Redux hooks and middleware enhancements. I can now manage global state more efficiently and handle async actions with logging and error management.

Outcome of the Day:

- React-Redux hooks implemented
- Middleware enhanced for logging and error handling
- Async API calls tested using hooks
- Week 7 advanced tasks progressed and documented