

# Industrial Training Daily Diary

## MERN Stack Training

**Company:** Sensation Software Solutions

**Student Name:** Tajinder Kaur

**Training Duration:** 6 Months

**Day:** 28

---

## Day 28 – Redux Form Integration with Async API Calls + Error Handling

### Objective of the Day

The objective of Day 28 was to integrate **React forms with Redux** for **asynchronous API calls**, and implement **error handling** to manage responses and display meaningful messages to users.

---

## Redux Form Integration

Trainer explained:

- Mapping form input fields to Redux store state
- Dispatching actions on input change
- Using Redux to store form submission data
- Benefits: centralized state, easier debugging, predictable data flow

Example: Registration form data stored in Redux.

---

## Async API Calls

Key points covered:

- Using Redux Thunk middleware for async actions
- Writing action creators that make API calls
- Dispatching actions on success/failure of API requests
- Managing loading states

Example: Submitting registration form data to a mock backend.

---

## Error Handling

Trainer explained how to handle API errors:

- Displaying field-specific errors returned from backend
  - Handling network errors and showing generic messages
  - Using Redux state to store error messages
  - Conditional rendering of error alerts in the form
- 

## Practical Work Done

During lab session:

- Connected registration form to Redux store
- Implemented async action to POST data to backend
- Added loading indicator during API call
- Displayed success or error messages based on API response
- Tested error handling for invalid input and network failures

This hands-on session helped understand real-world async form submissions.

---

## Weekly Task – Week 7 (Extended)

Tasks included:

- Complete Redux integration for all forms
  - Implement async API calls with Redux Thunk
  - Add proper error handling and display messages
  - Push code and diary updates to GitHub Pages
- 

## Importance in MERN Stack Development

Async API calls with Redux integration are essential for handling backend communication efficiently, providing better UX, and maintaining predictable state management.

---

## Conclusion of Day 28

Day 28 gave practical experience in managing form data via Redux, making asynchronous API calls, and handling errors effectively. I can now ensure reliable data flow and responsive UI behavior.

---

### **Outcome of the Day:**

- Redux integration for forms completed
- Async API call implemented with Redux Thunk
- Error handling added for user feedback
- Week 7 extended task worked on and documented