WEB APP DEVELOPMENT WITH REACTJS (INT252)

Lecture 19: Redux & Global State (Introduction)

Unit V – State Management & Advanced React

Syllabus Mapping:

- Redux basics
- Global state management
- Store, actions, reducers (conceptual)

Target Learner: Beginner → Intermediate

Why Redux Exists

You already know:

- Props → local
- Context → global (small apps)

Problem:

- Large apps become hard to manage

When Context Becomes Difficult

```
AuthContext
ThemeContext
CartContext
UserContext
```

Too many contexts ❌

What is Redux?

Redux is:

- A predictable state container
- Centralized global store

Single source of truth ✅

Redux Mental Model

```
UI → Action → Reducer → Store → UI
```

One-way data flow

Real-World Analogy

Redux Store = Bank Ledger

- One place
- Controlled updates

Core Redux Concepts

1. Store – holds data
2. Action – what happened
3. Reducer – how state changes

Redux Store

The store:

- Contains entire app state
- Read-only

Action

Action is:

- Plain JS object
- Describes event

```
{ type: 'INCREMENT' }
```

Reducer

Reducer is:

- A pure function
- Takes state + action
- Returns new state

Reducer Example

```
function counterReducer(state = 0, action) {
  switch (action.type) {
    case 'INCREMENT':
      return state + 1;
    default:
      return state;
  }
}
```

Redux Flow Explained

```
Button Click
   ↓
Dispatch Action
   ↓
Reducer updates state
   ↓
Store changes
   ↓
UI updates
```

Why Redux Is Predictable

- No direct state mutation
- Only reducers update state

Debug-friendly ✅

Redux vs Context

Context	Redux
Simple	Scalable
Few updates	Complex apps
Less tooling	Strong tooling

Redux Toolkit (Modern Redux)

Modern Redux uses:

```
@reduxjs/toolkit
```

Simpler, safer, recommended

When to Use Redux

✅ Large applications
✅ Many shared states
✅ Complex updates

When NOT to Use Redux

❌ Small apps
❌ Simple state

Common Beginner Mistakes

❌ Using Redux too early

❌ Mutating state

❌ Overengineering

Practice Exercises (Conceptual)

1. Identify store data in an app
2. Write action for login
3. Explain reducer role

✅ Answers – Practice

1. User, cart, theme
2. `{ type: 'LOGIN' }`
3. Reducer updates state

Key Takeaways

- Redux manages global state
- Uses actions and reducers
- Predictable one-way flow

Next Lecture

Lecture 20: Redux Toolkit & React Integration (Basics)

Unit V – State Management & Advanced React