Redux

Redux is a **state management library** for JavaScript apps (most commonly with React). It provides a **centralized store** where all application state lives, and a strict way to update it.

Real World Example:

Without Redux→ Everyone keeps money at home (local state). To transfer, they must directly exchange cash.
With Redux→ All money is kept in the **bank (store)**. Transactions happen via deposit/withdraw slips (actions).

The **bank manager (reducer)** processes the request and updates the balance.

Why Redux

Without Redux (Normal React State): Imagine you're in a **big school** 🏫.

- Each **classroom (React component)** has its own **attendance register (state)**.
- If the **Principal (App)** wants to know total students, he must **ask every classroom separately**.
- If a student **moves from Class A to Class B**, both teachers must **update their registers manually**.

-> Messy, confusing, and prone to errors.

With Redux (Central Store)

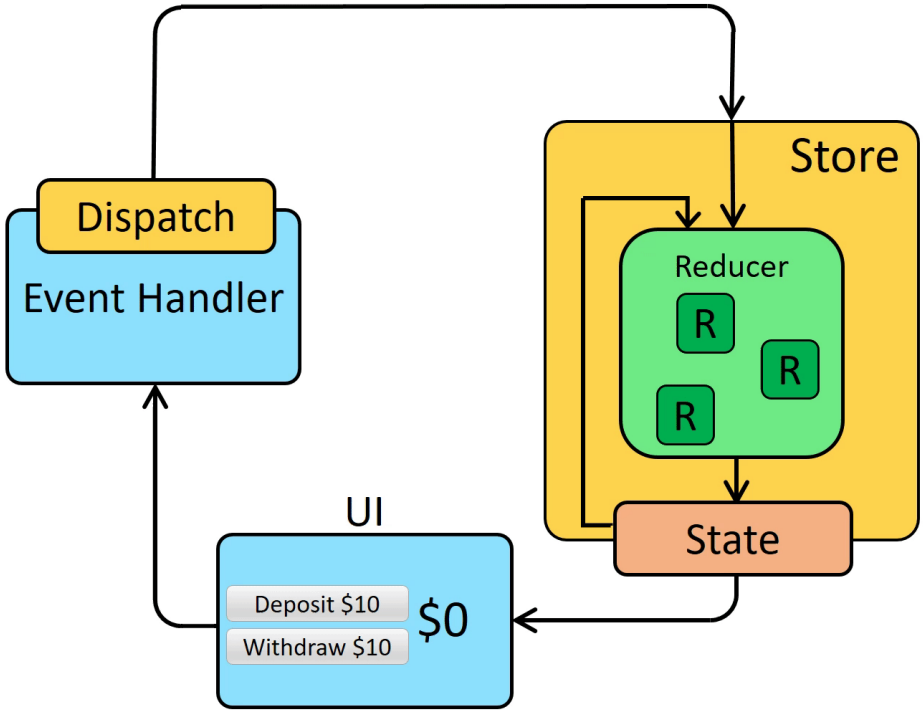
Now, the school decides to use a **single Central Attendance Office (Redux Store)**.

- All classrooms report student changes (add/remove) to this **one office**.
- Teachers don't keep separate registers — they just **query the office** when needed.
- The office updates records **in one place**, so the information is **always correct and consistent**.

-> Clean, predictable, and reliable.

Redux Data Flow (One-Way)

1. **View (React Component)** → dispatches an **Action**.
2. **Action** → sent to the **Store** via dispatch().
3. **Store** → runs the correct **Reducer** to update state.
4. **React Components** subscribed to the store → re-render with new data.



Redux Terminology

Store: The single source of truth (holds the whole app's state).
->The **bank vault** 🏦 where all money (data) is stored.
Slice: A piece of the store that manages **one feature's state + actions + reducers** (created with createSlice).
->A **slice of pizza** 🍕 → Each slice is a feature, together they form the full pizza (store).
State: The actual data stored inside Redux at a given time.
->Your **bank balance** right now.
Action: An object that describes **what happened** (must have a type).
->A **deposit/withdraw slip** you give to the bank.
Reducer: A pure function inside the slice that changes the state based on actions.
->The **bank manager** 👨‍💼 who updates your account balance.
Dispatch: A method used to send actions to the store.
->Putting your slip into the bank's counter box.
Provider: A React component that makes the store available to all components.
->A **school notice board** 🏫 that shares info everywhere.
Selector: A function that extracts a specific part of state from the store.
->Looking up **only your balance** in the ledger instead of all accounts.
Middleware: Functions that run between dispatching an action and reaching the reducer.
->A **security guard** 🔒 who checks slips before giving them to the manager.

Install and Configure

Step 1: **Create your project**

```
npm create vite@latest  my-redux-project
cd my-redux-project
```

Step 2: **Install Redux and Toolkit**

```
npm install @reduxjs/toolkit  react-redux
```

Step 3: **Start using Redux**

Project Work

Build a **Shopping Cart App** using Flux:

- **Actions:** ADD_ITEM, REMOVE_ITEM, CLEAR_CART.
- **Store:** Manages cart items & total price.
- **Components:** Cart view + Add item form.