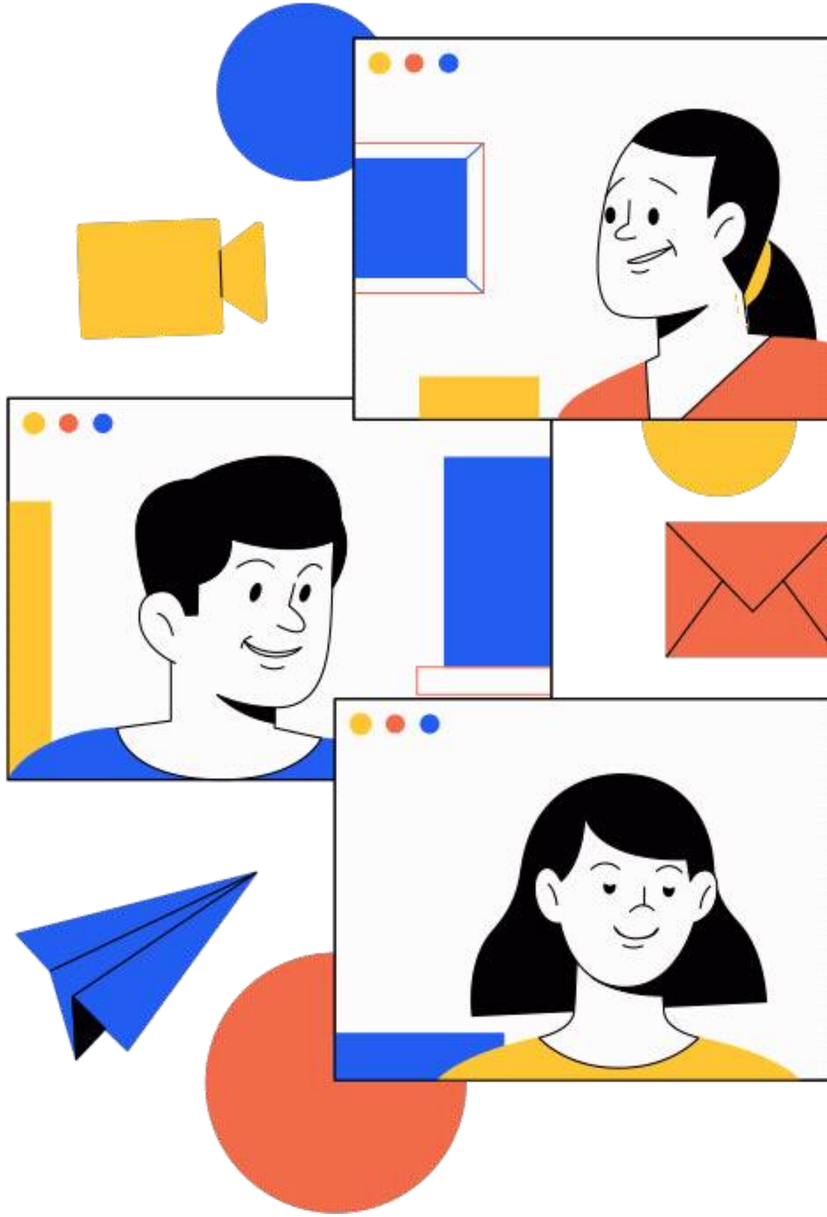
UDACITY

Week 6 - React ft. Redux

React creates; Redux orchestrates.



Ahmed Fouad Lotfy

React Session Lead

Agenda

What we'll cover in this session

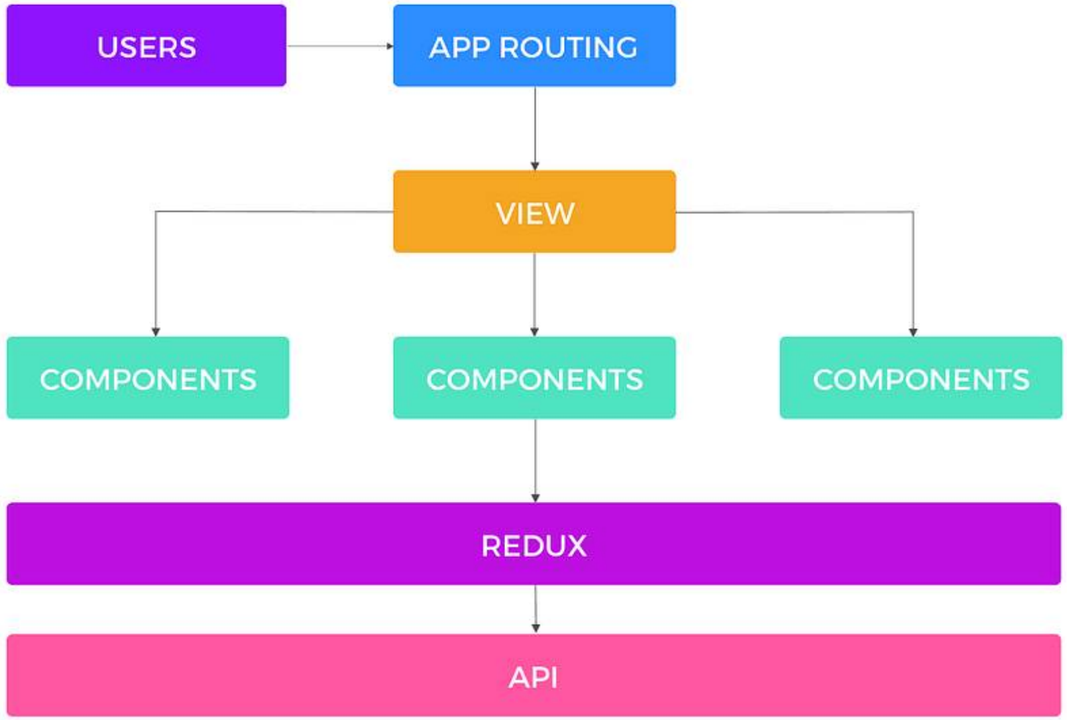
- Starting with React
- Combining Redux and React
- Asynchronous Redux
- Thunk
- Live Demo

Week
Prerequisites

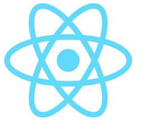
you don't have to be a master

1. **HTML/CSS**
2. **JavaScript**
3. **React**

Did you forget React?



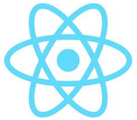
React.js Application Flow



Did you forget React?

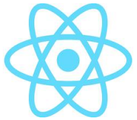
- Uses Virtual DOM
- Uni-directional Data flow
- Composition
- Uses JSX
- Most Famous Frontend Framework

What else do you remember about React?



Why Integrate Redux with React?

- **State Management:** React's local state can become difficult to manage in large apps. Redux provides a centralized solution.
- **Separation of Concerns:** React handles the view layer, while Redux manages the state, leading to cleaner and more maintainable code.
- **Improved Debugging:** Redux's developer tools allow you to inspect every state change, which is especially useful in React apps.



How to Integrate Redux with React?

- **Install Redux and React-Redux** Command: npm install redux react-redux
- **Create the Redux Store:** Create a store using createStore from Redux.
- **Define Reducers and Actions:** Brief explanation of how to create actions and reducers to manage state.
- **Connect React Components to Redux:** Connect function from react-redux to link React components with the Redux store.

Container

- **Purpose**: Handle the logic and state management of the app.
- **Focus**: How things work (data fetching, state updates).
- **Content**: Connect to Redux or other data sources.

Presentational

- **Purpose**: Focus on UI and display of the data. FAST on navigating.
- **Focus**: How things look (styling, layout).
- **Content**: Receive data via props from container components.

But this is not what we are going to do

But first lets talk about

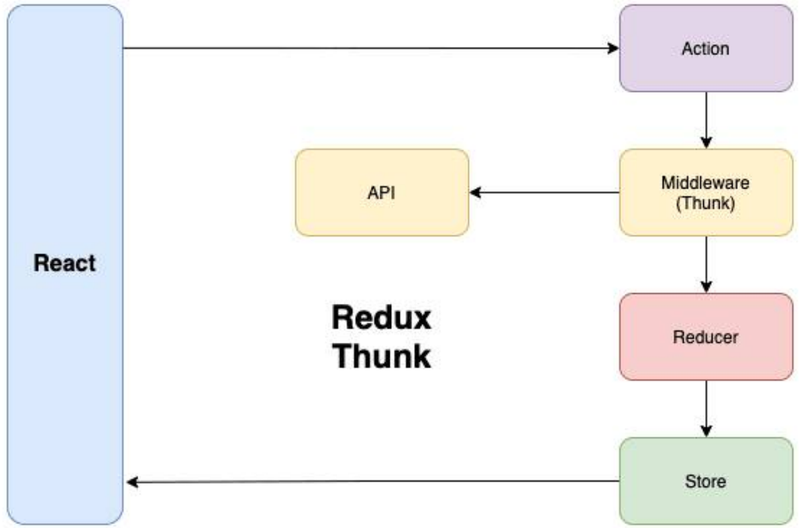
Asynchronous Redux

- **Why Async Operations Matters?** Real-World Scenarios applications need to interact with external APIs or services (e.g., fetching data, submitting forms, etc.) and Ensures data is loaded and updated in real-time, providing a seamless experience.

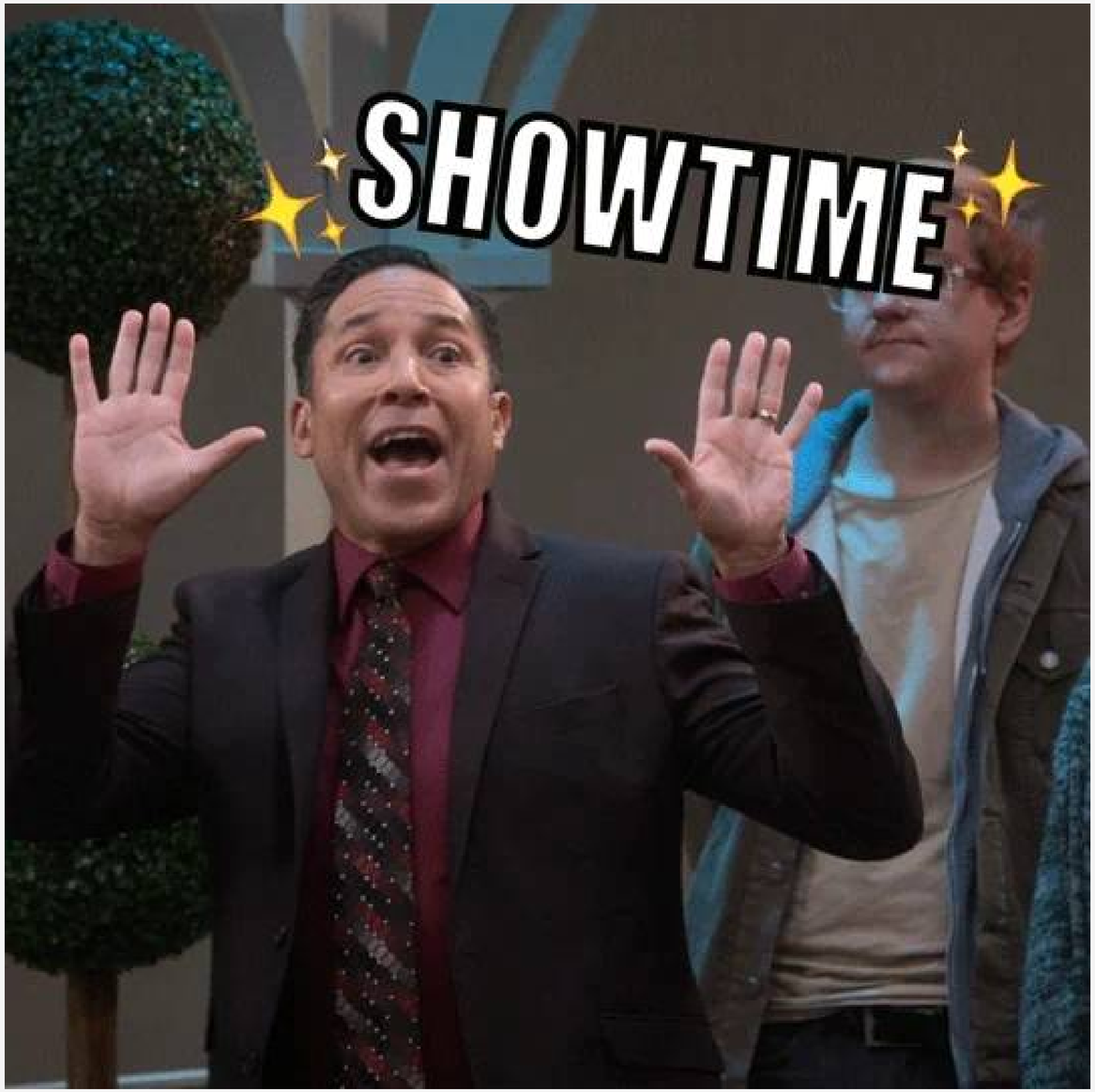
- **What are the Challenges with Async in Redux?** Redux by default is synchronous, so handling asynchronous tasks (like API calls) requires additional mechanisms and ensuring the state accurately reflects the status of async operations (e.g., loading, success, error) is crucial for a predictable app.

Redux Thunk

- **What is a Thunk?** A middleware that allows you to write action creators that return a function instead of an action. This function can then perform side effects, including async tasks, and dispatch actions based on the result.
- **How It Works?**
  - Action creator returns a function instead of an action object.
  - This function can dispatch other actions based on the async result (e.g., dispatching a success or failure action).



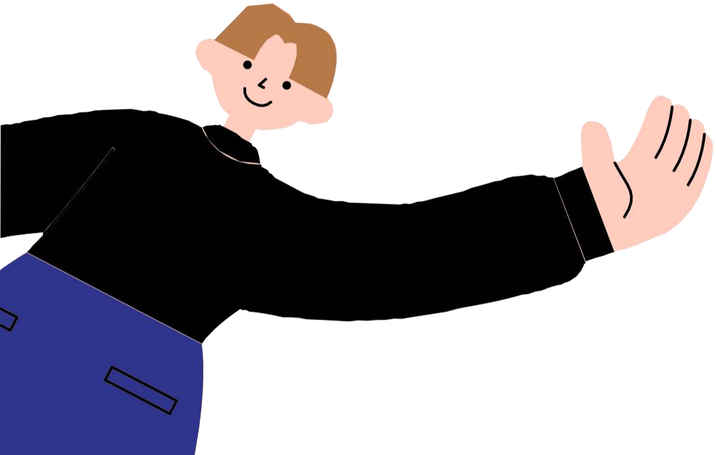
Its Demo Time

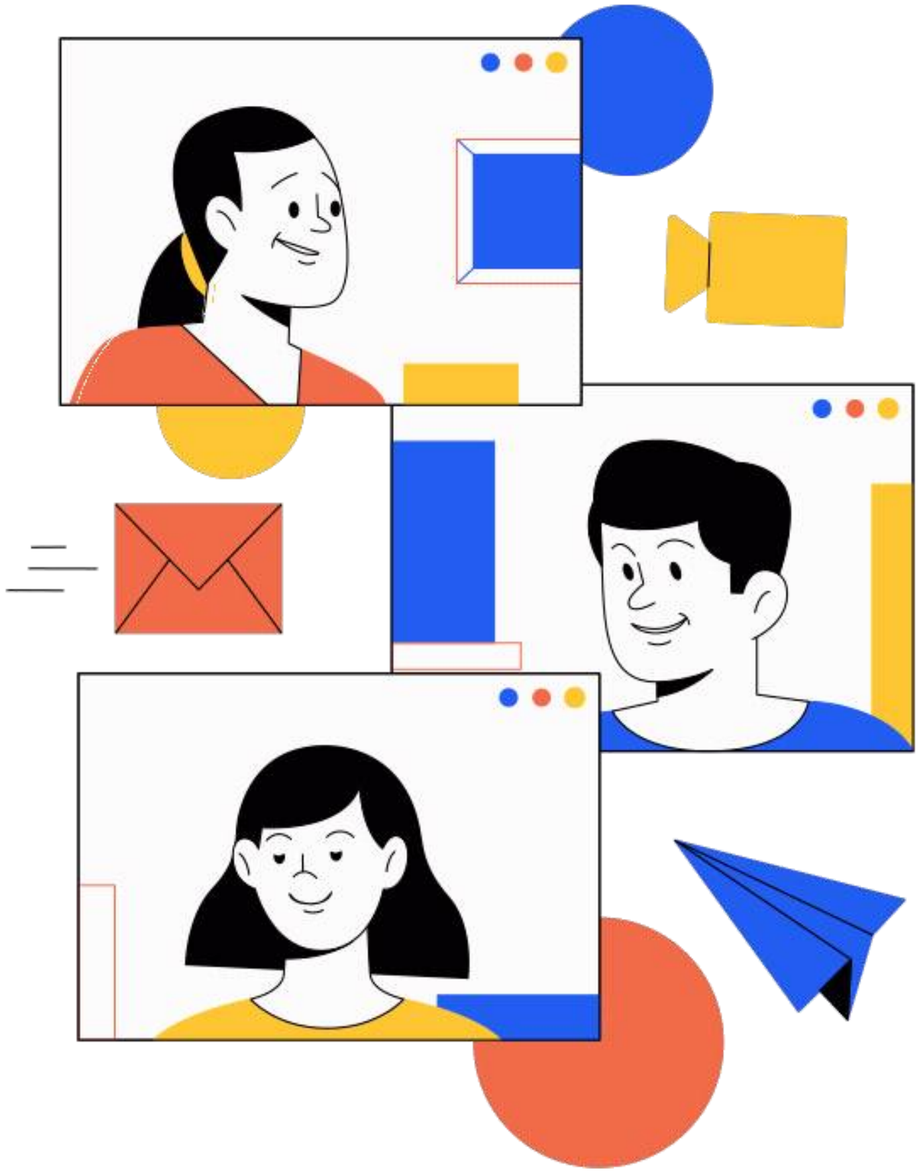


Next Session:

More React and Redux

Any Questions?





Thank you for attending!

Feel free to email at **a.lotfy@fci-cu.edu.eg** or reach me at circle anytime for any questions or clarifications!

Follow me on Github **@ahmeddxfouad**

code and slides are found at this **github repo**