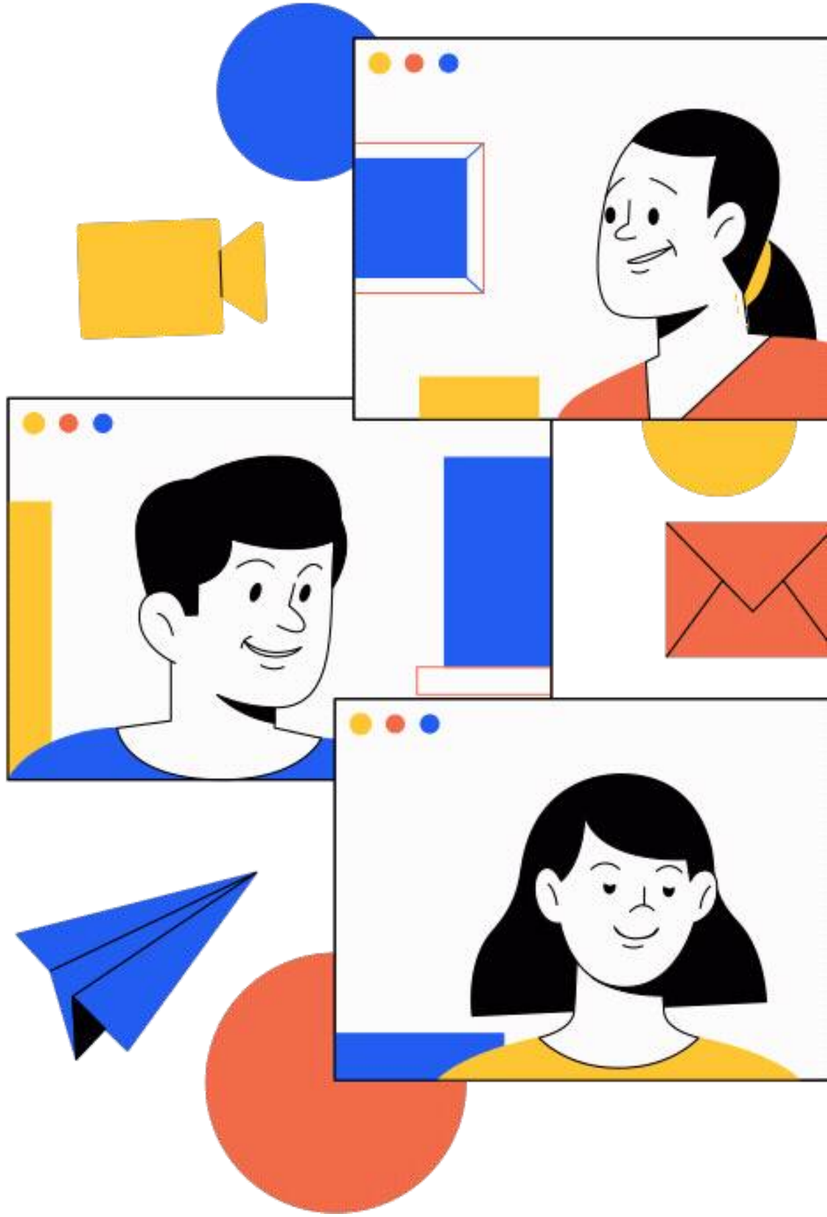
UDACITY

Week 7 - Connect React with Redux

Orchestrates some todos.



Ahmed Fouad Lotfy

React Session Lead

Agenda

What we'll cover in this session

- Provider
- Context
- React Connect
- Application Structure
- Live Demo

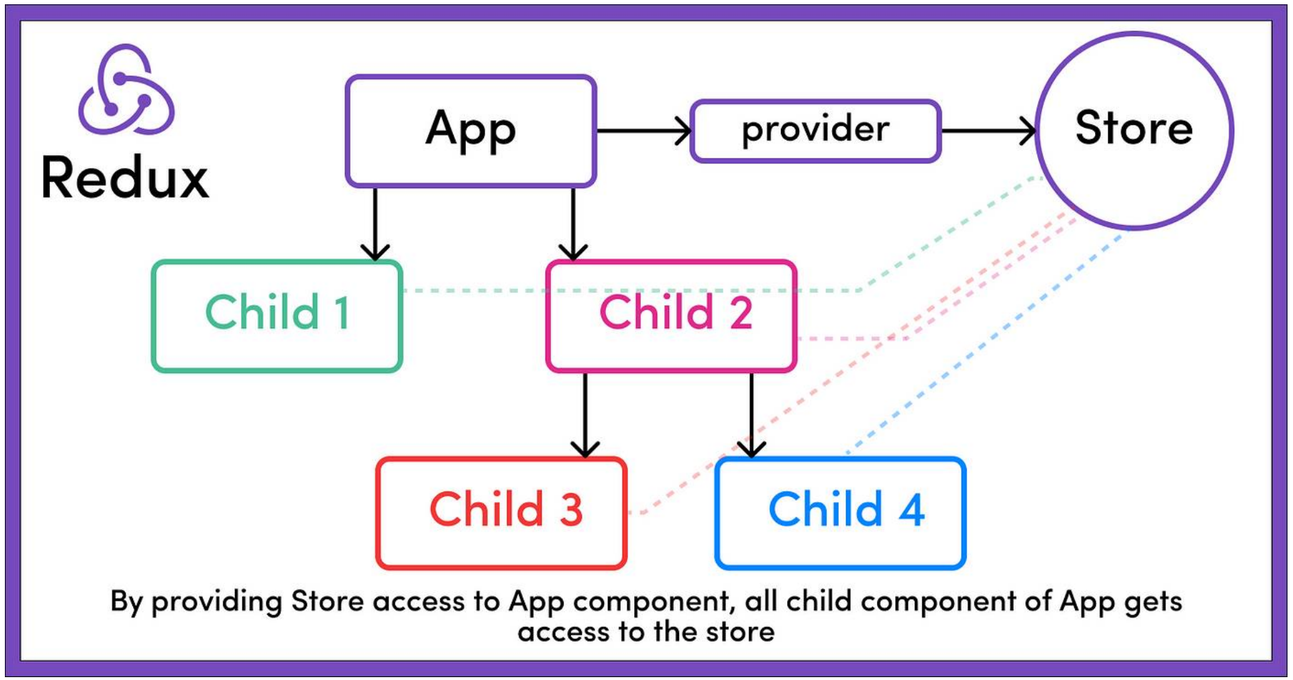
Week
Prerequisites

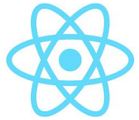
you don't have to be a master

1. HTML/CSS
2. JavaScript
3. React
4. Redux

What is the Provider?

- **The Provider:** it’s a component in ‘react-redux’ library that makes the Redux store available to all child components, ensuring they can access the store without passing props manually.

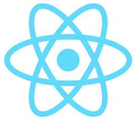




Why is the Provider Important?

- **Global Access to Store:** Allows any component in your React app to access the Redux state, making state management centralized and efficient.

- **Automatic Propagation:** No need to manually pass the Redux store to every component. Provider ensures that the Redux state is available throughout the component tree.



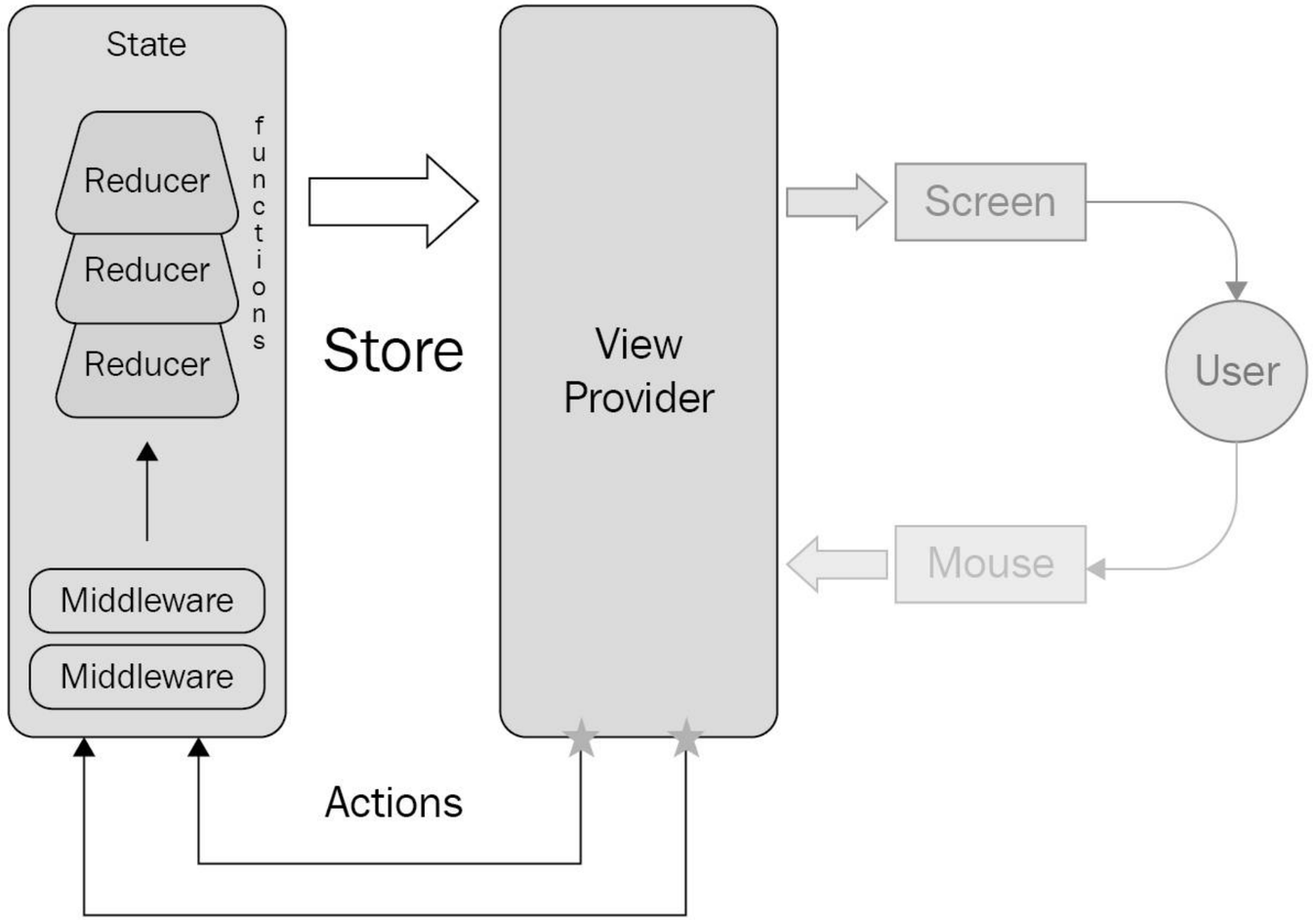
How to use Provider?

- Wrap your root React component (usually <App />) in the Provider component and pass the Redux store to it.

- As you can see the Provider is imported from ‘react-redux’ library.

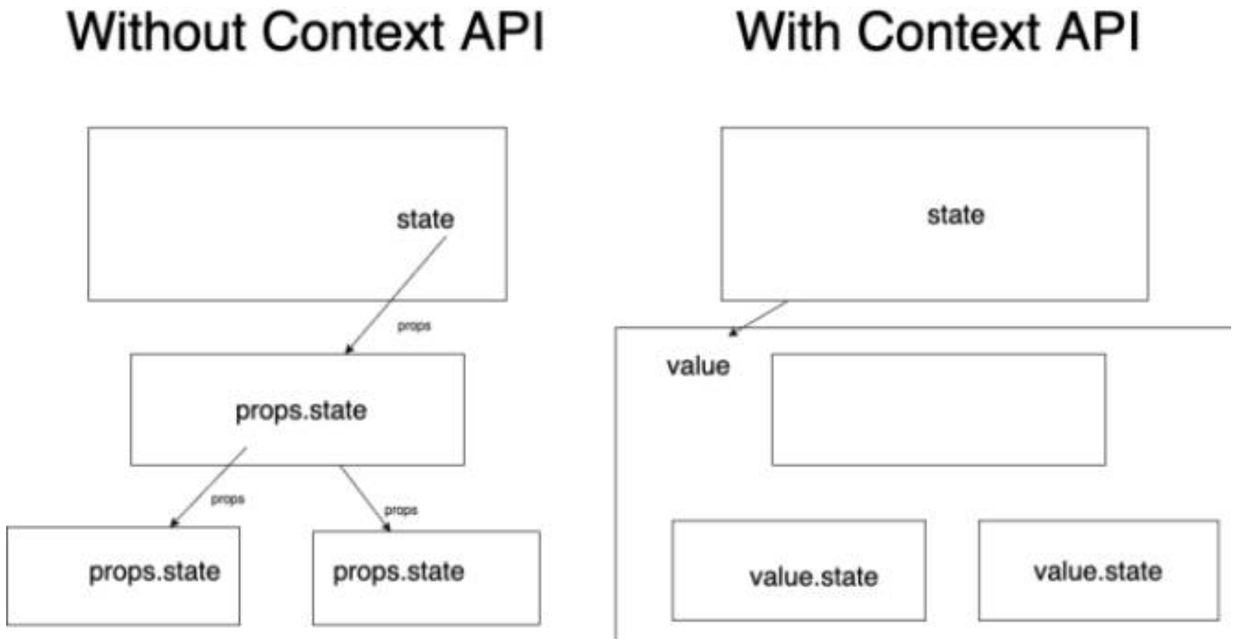
```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from 'react-redux';
import store from './store';
import { App } from './components';

ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>,
  document.getElementById('app')
);
```

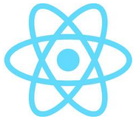
How it works?


State
Reducer
Reducer
Reducer
functions
Middleware
Middleware
Store
View
Provider
Screen
User
Mouse
Actions

What is React Context API?

- **Definition:** The Context API in React allows you to pass data through the component tree without needing to pass props manually at every level, creating a global-like state for components.





When to use Context?

- **For Non-Business Logic Data:** Context is ideal for data that isn’t crucial for business logic, such as themes, locales, or simple user settings.

- **Redux vs Context:** While Redux is typically used for managing global application state with complex logic, Context can be used for simpler, less frequent updates.

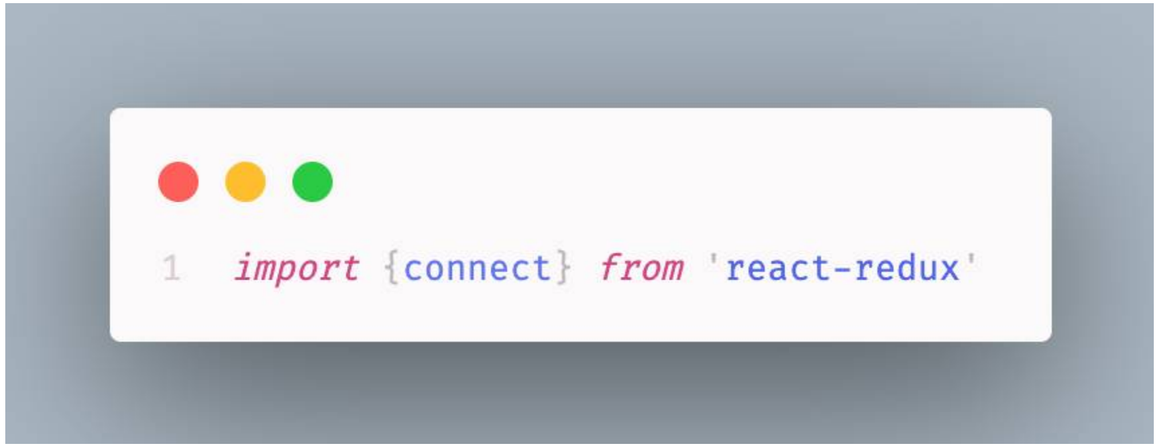
How to create and use Context?

- **Step-1:** Create a Context object.
- **Step-2:** Use the Provider to pass the data.
- **Step-3:** Consume the context using useContext or the Context.Consumer in child components.

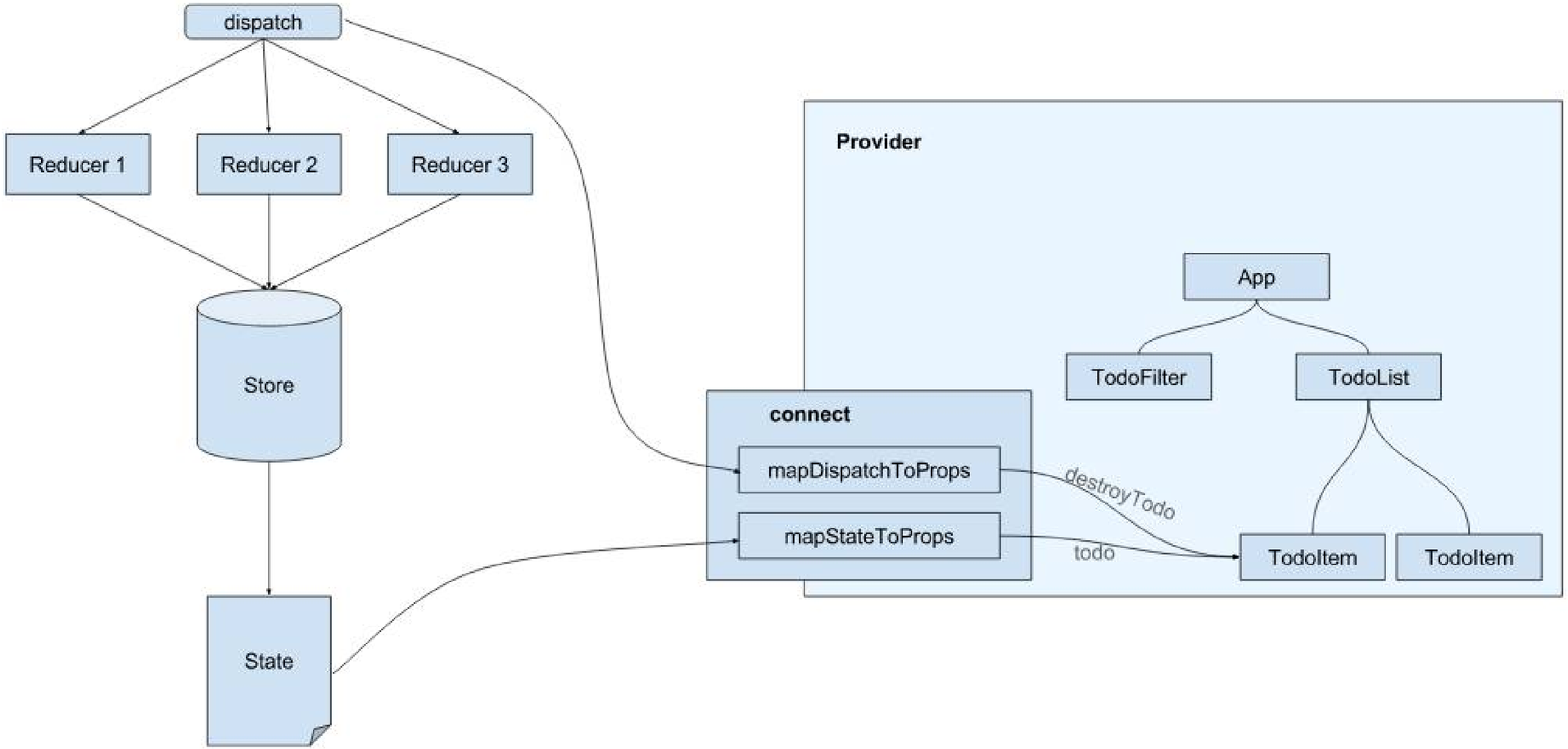
```
import { createContext } from 'react;

export const UserContext = createContext();

export const UserProvider = ({ children }) => {
  return (
    <UserContext.Provider value='Default State'>
      {children}
    </UserContext.Provider>
  )
}
```

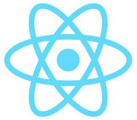
What is Connect Function?

- **Definition:** The connect function is used to connect React components to the Redux store. It allows components to subscribe to the Redux state and dispatch actions.



What is Connect Function?





How Connect Works?

- **Mapping State to Props:** You define how the Redux state should be mapped to the component's props.

- **Mapping Dispatch to Props:** Define functions that dispatch actions and pass them as props to the component. (Not going to use it)

```
const mapStateToProps = (state) => {
    return { shows: state.shows }
}

const mapDispatchToProps = dispatch => {
    return {
      addShow: (show) => {
        dispatch(addShow(show))
      }
    };
  };

export default connect(mapStateToProps, mapDispatchToProps)(ShowsList)
```

How to create and use Context?

- **Step-1:** Create a Context object.
- **Step-2:** Use the Provider to pass the data.
- **Step-3:** Consume the context using useContext or the Context.Consumer in child components.

```
import { createContext } from 'react;

export const UserContext = createContext();

export const UserProvider = ({ children }) => {
  return (
    <UserContext.Provider value='Default State'>
      {children}
    </UserContext.Provider>
  )
}
```

But how Provider, Context, and Connect Work in Redux-React Applications Redux with React?

React && Redux

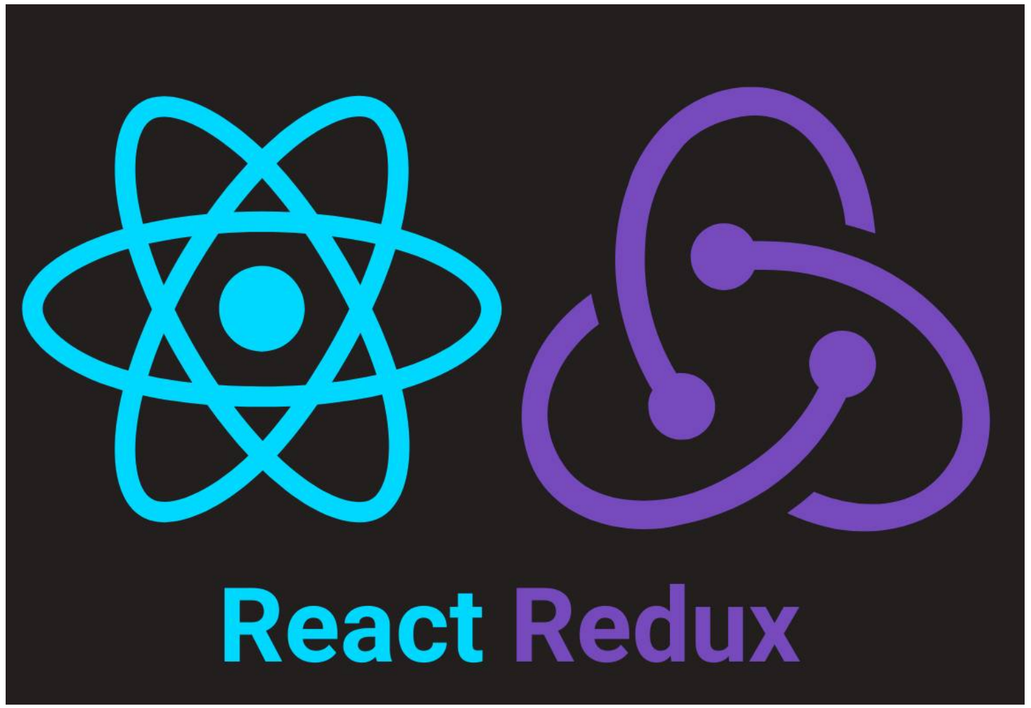
- **Provider vs. Context:**

  - Provider: *Specific to Redux, making the store available across the app.*

  - Context API: *General React feature for passing data down the tree, useful for simpler cases.*

- **Provider + Connect:** *Provider supplies the Redux store, and connect hooks components into the state and dispatch functions, enabling them to read state and send actions.*

- **Context API + Redux:** *While Context API can be used in parallel with Redux for certain tasks (like theme handling), it’s generally not used for complex business logic, which Redux handles better.*

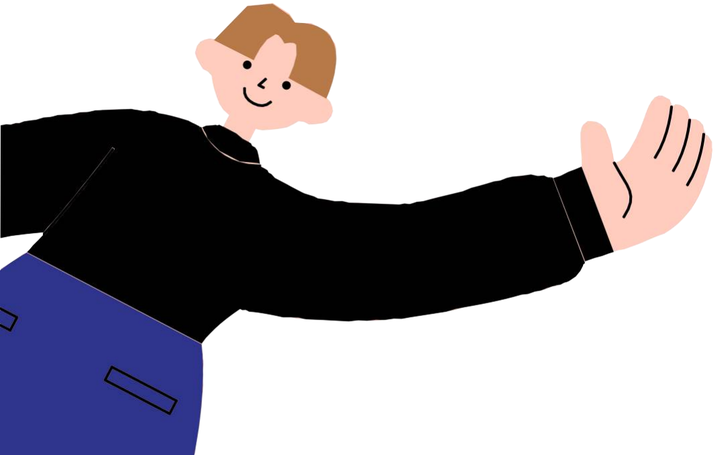


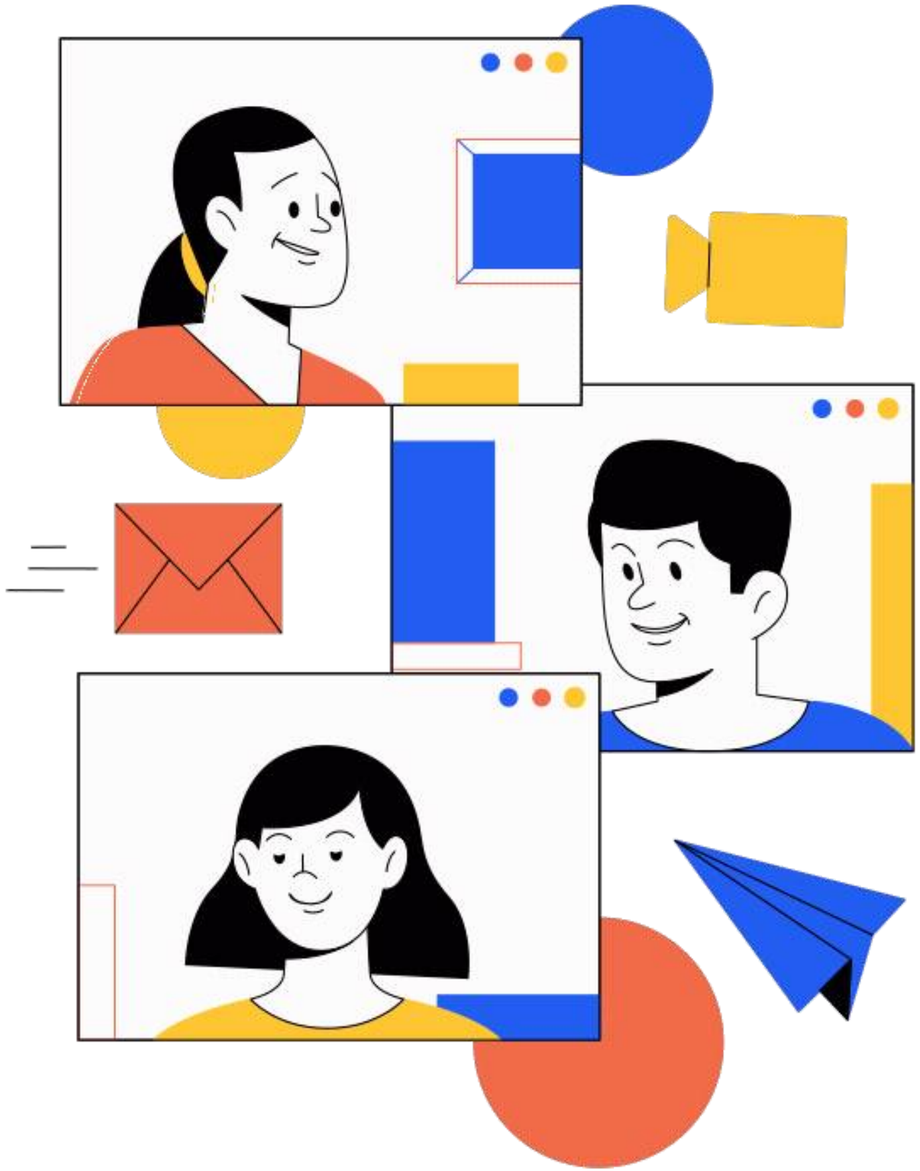
Its Demo Time

Next Session:

Testing with Jest

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**