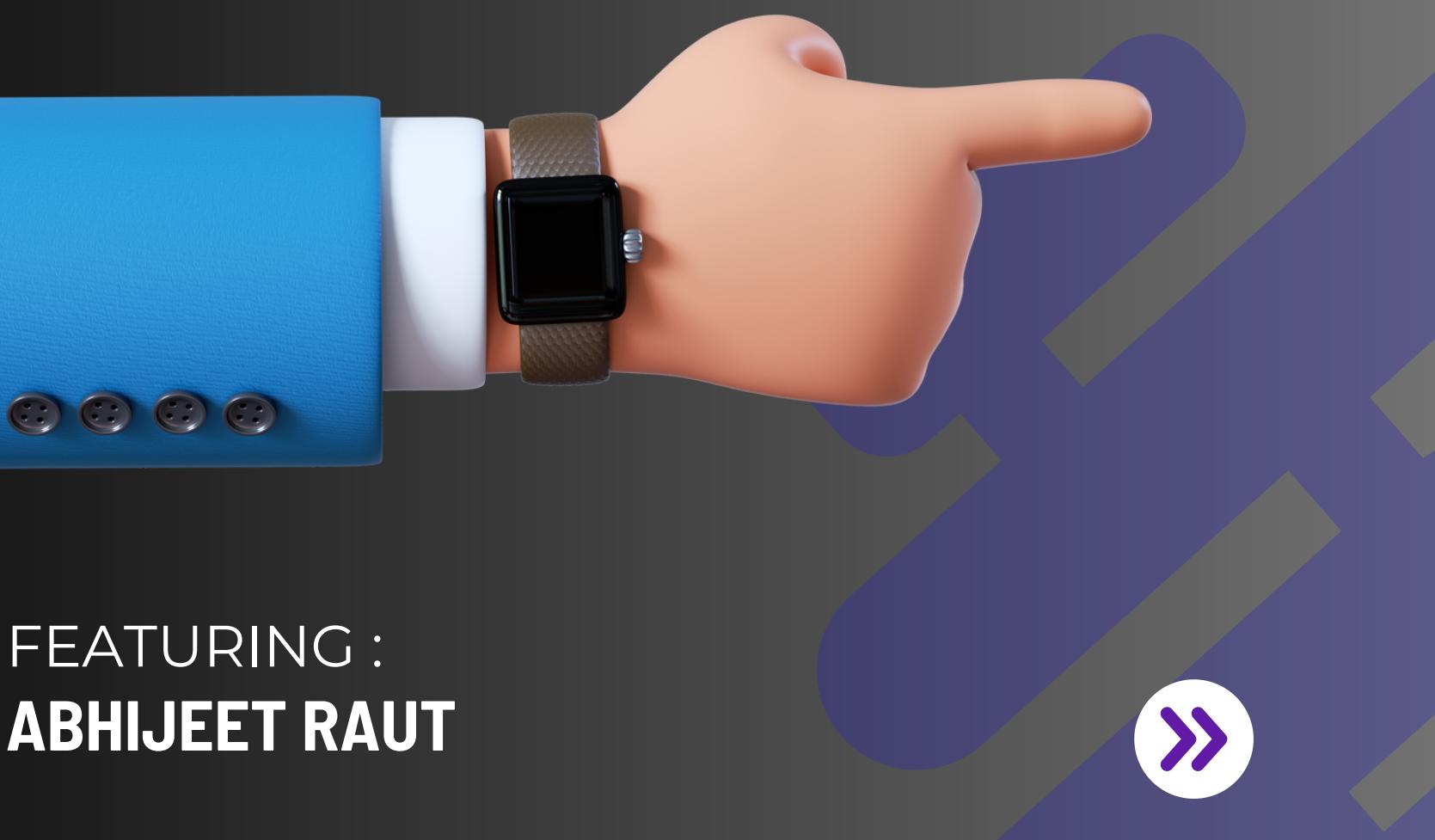


@raut-abhijeet

# MASTER REDUX WITH 5 EASY STEPS



FEATURING :  
**ABHIJEET RAUT**



01

# WHAT IS REDUX TOOLKIT?

REDUX TOOLKIT IS LIKE A MAGIC TOOLBOX FOR REDUX. IT HELPS YOU WRITE REDUX CODE FASTER AND WITH LESS FUSS.



02



## WHY USE REDUX TOOLKIT?

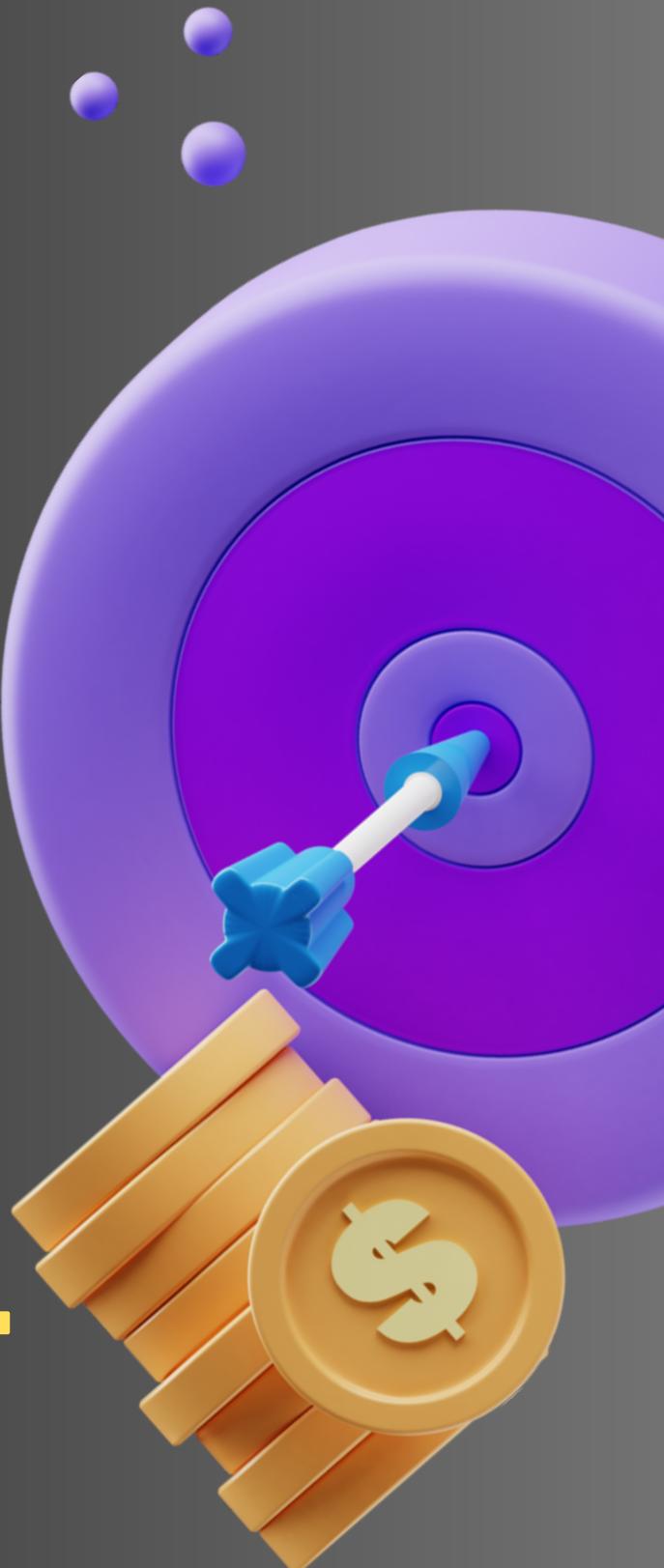
REDUX TOOLKIT IS USED FOR MANAGING  
COMPLEX STATE IN REACT APPS.

WHETHER HANDLING USER  
AUTHENTICATION, CACHING API RESPONSES,  
OR SYNCING GLOBAL APP STATE, REDUX  
TOOLKIT PROVIDES A RELIABLE SOLUTION  
FOR MANAGING APP-WIDE DATA WITH EASE.

@raut-abhijeet

# 03

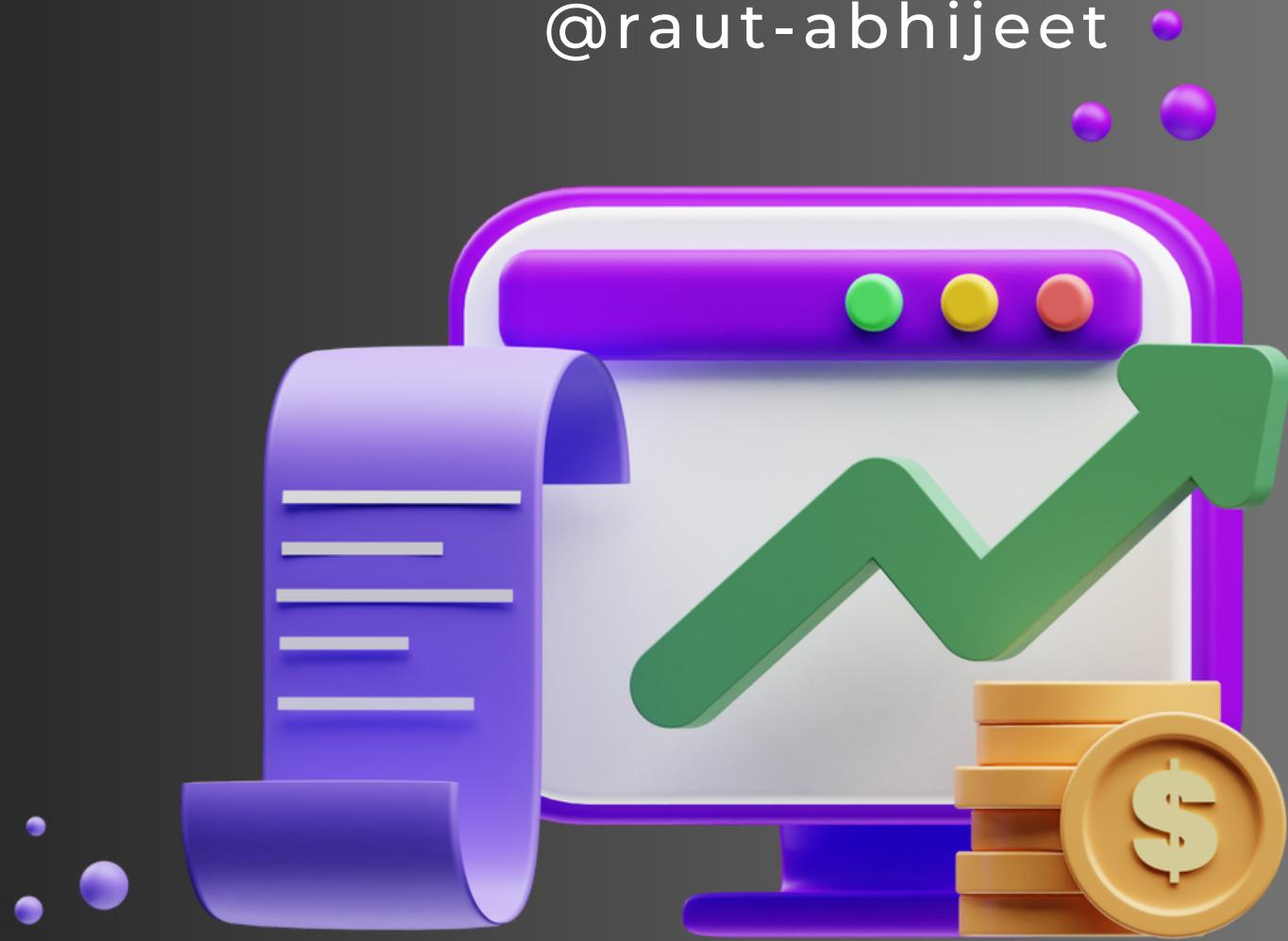
## INSTALLING REDUX TOOLKIT



```
npm install @reduxjs/toolkit
```



# 04



## CREATE A STORE

LET'S SET UP THE REDUX STORE. IT'S LIKE A BIG BOX WHERE YOUR APP'S DATA LIVES. REDUX TOOLKIT MAKES IT EASY

```
import { configureStore } from '@reduxjs/toolkit';
import counterReducer from './counterSlice';

const store = configureStore({
  reducer: {
    counter: counterReducer,
  },
});
```

# 05



## CREATE A SLICE

SLICES ARE LIKE SLICES OF CAKE IN REDUX TOOLKIT! THEY'RE SMALL PIECES OF YOUR APP'S DATA AND THE ACTIONS THAT CHANGE

```
import { createSlice } from '@reduxjs/toolkit';

const counterSlice = createSlice({
  name: 'counter',
  initialState: 0,
  reducers: {
    increment: state => state + 1,
    decrement: state => state - 1,
  },
});

export const { increment, decrement } = counterSlice.actions;
export default counterSlice.reducer;
```

# 06

## WRAP WITH PROVIDER



TO USE REDUX TOOLKIT, YOU NEED TO PROVIDE THE REDUX STORE TO YOUR REACT APP. WRAP YOUR APP IN A <PROVIDER>

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

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

# 07

## DISPATCH ACTIONS



ACTIONS ARE LIKE MESSENGERS THAT TELL REDUX WHAT TO DO. WITH REDUX TOOLKIT, SENDING ACTIONS IS SUPER SIMPLE

```
import { useDispatch } from 'react-redux';
import { increment, decrement } from './counterSlice';

function CounterComponent() {
  const dispatch = useDispatch();

  return (
    <div>
      <button onClick={() => dispatch(increment())}>Add 1</button>
      <button onClick={() => dispatch(decrement())}>Subtract 1</button>
    </div>
  );
}
```

# 08



## USE-SELECTOR HOOK

NEED TO SEE WHAT'S IN YOUR REDUX STORE? NO PROBLEM! USE USESELECTOR IS THERE FOR YOU

```
import { useSelector } from 'react-redux';

function CounterDisplay() {
  const counter = useSelector(state => state.counter);

  return <div>Count: {counter}</div>;
}
```

@raut-abhijeet

DON'T FORGET  
TO LIKE, SHARE  
AND REPOST IF  
YOU LIKE THIS  
POST



FEATURING :  
**ABHIJEET RAUT**

