


jsx

 Copy code

```
// index.js
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App'; // Import your root component
import './index.css'; // You can import your CSS here if needed
import { Provider } from 'react-redux';
import store from './store'; // Import your Redux store


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

2. Create Redux Slice:

Create a Redux slice to manage the temperature state. Define the initial state and reducers for handling temperature conversion logic. For instance, you can create a

`temperatureSlice.js` file:

javascript

 Copy code

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

const initialState = {
  celsius: 0,
  fahrenheit: 32,
};


const temperatureSlice = createSlice({
  name: 'temperature',
  initialState,
  reducers: {
    setCelsius: (state, action) => {
      state.celsius = action.payload;
      state.fahrenheit = (action.payload * 9/5) + 32;
    },
    setFahrenheit: (state, action) => {
      state.fahrenheit = action.payload;
      state.celsius = (action.payload - 32) * 5/9;
    },
  },
});

export const { setCelsius, setFahrenheit } = temperatureSlice.actions;
export default temperatureSlice.reducer;
```

3. Configure Redux Store:

Configure the Redux store to include your temperature slice. Create a `store.js` file:

javascript


 Copy code

```
// store.js
import { configureStore } from '@reduxjs/toolkit';
import temperatureReducer from './temperatureSlice';

const store = configureStore({
  reducer: {
    temperature: temperatureReducer,
  },
});

export default store;
```

jsx

 Copy code

```
// TemperatureConverter.js
import React from 'react';
import { useDispatch } from 'react-redux';
import { setCelsius, setFahrenheit } from './temperatureSlice';

const TemperatureConverter = () => {
  const dispatch = useDispatch();

  const handleCelsiusChange = (value) => {
    dispatch(setCelsius(parseFloat(value)));
  };

  const handleFahrenheitChange = (value) => {
    dispatch(setFahrenheit(parseFloat(value)));
  };

  return (
    <div>
      <input
        type="number"
        onChange={(e) => handleCelsiusChange(e.target.value)}
      />
      Celsius =
      <input
        type="number"
        onChange={(e) => handleFahrenheitChange(e.target.value)}
      />
      Fahrenheit
    </div>
  );
};

export default TemperatureConverter;
```

2. Create Display Component:

Create a component to display the converted temperature values.

jsx

 Copy code

```
// TemperatureDisplay.js
import React from 'react';
import { useSelector } from 'react-redux';

const TemperatureDisplay = () => {
  const { celsius, fahrenheit } = useSelector((state) => state.temperature);

  return (
    <div>
      <p>Celsius: {celsius}°C</p>
      <p>Fahrenheit: {fahrenheit}°F</p>
    </div>
  );
};

export default TemperatureDisplay;
```

3. App Component:

Finally, assemble everything in your main "App" component.

jsx

 Copy code

```
// App.js
import React from 'react';
import { Provider } from 'react-redux';
import store from './store';
import TemperatureConverter from './TemperatureConverter';
import TemperatureDisplay from './TemperatureDisplay';

const App = () => {
  return (
    <Provider store={store}>
      <div>
        <h1>Temperature Conversion App</h1>
        <TemperatureConverter />
        <TemperatureDisplay />
      </div>
    </Provider>
  );
};

export default App;
```