**REDUX DEMO**

**Commands used for installation :**

cd redux-demo ge hogu

npm install redux react-redux

npm install @reduxjs/toolkit react-redux

npm install redux-thunk

npm install redux-observable rxjs

npm install redux-undo

npm install immutable

**Steps to create a new projects and install packages :**

File -> New Project -> use the above commands -> add the codes.

**App.js**

import React, { useEffect } from 'react';

import { useSelector, useDispatch } from 'react-redux';

import { fetchPosts } from './features/postsSlice';

function App() {

const dispatch = useDispatch();

const { items, status, error } = useSelector((state) => state.posts);

useEffect(() => {

if (status === 'idle') {

dispatch(fetchPosts());

}

}, [status, dispatch]);

return (

<div>

<h1>Redux Async Data Fetching Example</h1>

{status === 'loading' && <p>Loading...</p>}

{status === 'failed' && <p>Error: {error}</p>}

{status === 'succeeded' &&

items.slice(0, 5).map((post) => (

<div key={post.id}>

<h3>{post.title}</h3>

<p>{post.body}</p>

</div>

))}

</div>

);

}

export default App;

**index.js**

// index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import { Provider } from 'react-redux';

import { store } from './store';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<Provider store={store}>

<App />

</Provider>

);

**App.test.js**

import { render, screen } from '@testing-library/react';

import App from './App';

test('renders learn react link', () => {

render(<App />);

const linkElement = screen.getByText(/learn react/i);

expect(linkElement).toBeInTheDocument();

});

**store.js**

// index.js

import React from 'react';

import ReactDOM from 'react-dom/client';

import { Provider } from 'react-redux';

import { store } from './store';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<Provider store={store}>

<App />

</Provider>

);

**postsSlice.js**

// features/postsSlice.js

import { createSlice, createAsyncThunk } from '@reduxjs/toolkit';

// Step 1: Create async thunk

export const fetchPosts = createAsyncThunk('posts/fetchPosts', async () => {

const response = await fetch('https://jsonplaceholder.typicode.com/posts');

return response.json();

});

// Step 2: Create slice

const postsSlice = createSlice({

name: 'posts',

initialState: {

items: [],

status: 'idle', // idle | loading | succeeded | failed

error: null,

},

reducers: {},

extraReducers: (builder) => {

builder

.addCase(fetchPosts.pending, (state) => {

state.status = 'loading';

})

.addCase(fetchPosts.fulfilled, (state, action) => {

state.status = 'succeeded';

state.items = action.payload;

})

.addCase(fetchPosts.rejected, (state, action) => {

state.status = 'failed';

state.error = action.error.message;

});

},

});

export default postsSlice.reducer;