**Neflix clone:**

Animation Implementation Using Framer Motion

Using the Framer Motion library, I implemented smooth animation on sliders, modal windows, logos, and more. Implement API communication capabilities using React-Query

Data was obtained by communicating 22 API keys using React-Query.

When searching, I received the user's search keyword as a parameter and entered it in the API key.

Use React-router-dom to deliver once received API values to the detail page

Using the useLocation and useMatch of React-router-dom, when the URL changes, data of the existing API value is sent so that the user can see the screen first with the existing data, and the detailed information API is communicated and displayed on the screen.

Use similar design components as reuse components in different situations

There is a function to recommend a similar movie or TV on the detail screen in a movie or TV. If you press a similar movie, the user will display it on the screen with the same layout

**Catch mind**

"Catch Mind" is a real-time drawing game built using Vanilla JS, SCSS, ExpressJS, Socket.IO, and Gulp. The project utilizes the latest version of ECMAScript, ES6, and NodeJS as the server-side runtime environment. Socket.IO library is used to enable real-time communication between clients and the server.

The project is based on the following theoretical concepts: Socket.IO, Events, Emit Events, Broadcast Events, and Canvas API. Socket.IO library provides the backbone for the real-time communication between the clients and server, while Events, Emit Events, and Broadcast Events allow for the creation of dynamic interactions between the players. Canvas API is used to enable live painting functionality, allowing users to draw on a shared canvas in real-time.

"Catch Mind" comes with several features, including live painting, real-time chat, and real-time notifications. The live painting feature allows users to draw on a shared canvas in real-time, while the real-time chat feature enables players to communicate with one another during the game. The real-time notification feature informs users of important game events, such as when a new player joins the game, a player leaves the game, or when the game ends. The scoreboard is another feature of "Catch Mind", which shows the player's scores and ranks during the game. Additionally, the project supports user identification, allowing players to create and customize their profiles

**Drag-n-Drop Trello**Build and implement input forms using React Hook Form

Using the React Hook Form, I was able to get less code than the amount used to receive the input value of the existing input form.

Manage application status using Recoil's atoms

The saved message list was managed globally through Recoil.

Leverage Beautiful DnD to implement drag-and-drop

Using the Beautiful DnD library, we implemented a drag-and-drop function for each board, and the colors of the board and messages change during drag.

**Momentum**Save User's Name -> Your name has been saved even when the site is reloaded using local storage.

Today's to-do input and output functions -> When the user enters the to-do task, it is saved on local storage, and the import is also linked to save even if the computer is turned off and off.

Today's weather function -> We used the site of https://openweathermap.org/ and used JavaScript's Navigator.geolocation.getCurrentPosition function to obtain API, LAT, and LON to provide the user's city, weather, and temperature.

Quote and wallpaper random function -> I used the Math.Random function to make it come out differently every time I reload.

Current Date and Time Function -> The current date and time are linked to the site using the new Date() function of JavaScript.

I made a stopwatch using the stopwatch function -> Interval and split and put a zero before the number in a way of ${hours<10?0${hours} : hours}. It was also saved to local storage so that the time wrap remained when reloaded, and the ability to delete the time wrap history was included. And when the start button is pressed several times, time passes quickly, so it is deactivated when the start button is pressed, stopped, and reactivated when the reset button is pressed.

**Crypto Tracker**

1.Coin Information, Dark Mode  
Use React Query to take data from cryptocurrency APIs and show real-time coin information.  
Using Styled Components and Recoil, we implemented light and dark mode switching.  
2. Data visualization, infinite scrolling  
Using Apex Charts, it visualizes price changes in cryptocurrencies and shows them in charts.  
When scrolling through infinite scrolling, you can check the information of all coins.

**Chat Application**

"Chat Application" is a Java program designed to enable real-time communication between multiple clients and a server. The program is built using JavaFx, which provides a robust and user-friendly graphical user interface (GUI) for the application. The program also utilizes socket programming to establish a connection between the clients and the server.

The project includes several key features, including low coupling and high cohesion, which ensure that the application is modular, flexible, and easy to maintain. The program is designed to handle exceptions, which allows the user to know if the inputs are incorrect.

The project leverages several Java concepts and techniques, including creating classes, methods, and using arrays. Java I/O is used to enable input and output operations in the program, while inheritance and polymorphism allow for the creation of dynamic and flexible code. Exception handling ensures that the program can handle errors and exceptions gracefully.

JavaFx is used to provide a rich and interactive user interface for the chat application, while lambda expressions and functional interfaces allow for the creation of more concise and efficient code. The program also uses the collection framework, which provides a comprehensive set of data structures and algorithms for working with collections of objects.

Java threads are used to enable concurrency and parallelism in the program, which ensures that the application can handle multiple clients simultaneously. Socket programming and Java networking are used to establish a connection between the clients and the server, enabling real-time communication between multiple users.

Overall, "Chat Application" is a fully functional and user-friendly Java program that enables real-time communication between multiple clients and a server. It leverages several key Java concepts and techniques, and provides a comprehensive set of features for handling exceptions, managing input and output operations, and enabling real-time communication between multiple users.