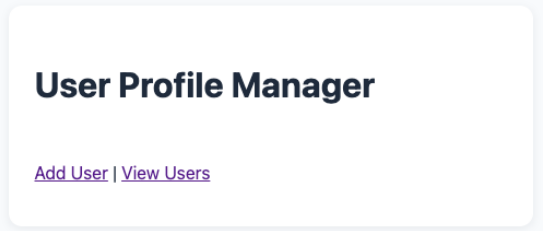

Lab 1 – Client-Side Rendering (CSR) vs. Server-Side Rendering (SSR)

In this lab exercise, you will install a simple web application (User Profile Manager) developed using client-side rendering (React app + Node.js server)* and server-side rendering (Spring Web) technologies. For this phase, the goal is to see the differences in terms of technologies for developing CSR and SSR. However, it might also be relevant to measure some web metrics (with Google Dev Tools/Lighthouse) to compare the two rendering technologies.

*(*Note: ~60% of the client-app is generated by Copilot)*

Task 1: Run the web applications

- Download the zipped files (csr-userapp_nodejs.zip and ssr-userapp.zip)

- Client-Side Rendering with React + Node.js
  - Unzip the csr-userapp_nodejs.zip. In the csr-userapp_nodejs folder, you will find the user-app (React app) and nodeserverexp(Node.js) server.
    - To run the the Node.js server,
      - From the terminal (Mac/Linux) or command line(Windows), **cd** to the nodeserverexp folder.
      - Install the dependencies by running the command: **npm install**
      - Run the command: **npm start**
      - The Node.js server will start on port 3001. You can access the server at http://localhost:3001
    - To run the user-app (React app),
      - from your terminal, **cd** to the **user-app** folder.
      - Install the dependencies by running the command: **npm install**
      - Start the app in dev mode by running the command: **npm run dev**
      - You can then access the app at http://localhost:5173
      - Use the features: Add User and View Users.

- Server-Side Rendering with Spring
  - Unzip the ssr-userapp.zip. The web app will be unzipped to ssr-userapp folder.
  - From the terminal, **cd** to **ssr-userapp**
  - Run the command: `./mvnw spring-boot:run`
  - The app will start on port 8090 and can be accessed at http://localhost:8090
  - Use the features: Add User and View Users

Task 2: Browse through the Source code

To interact with the source code for the two technologies, you may use:

- VS Code IDE for the CSR source code
- Eclipse/InteliJ/VS Code IDE for the SSR source code
- Browse through the code and compare the two technologies. What is your impression in terms of the lines of code and number of components?

Task 3: Run the Google Developer Tools to measure performance metrics (initial load time and subsequent load time)

- Access the Google Chrome Developer Tools (see the figure below)
- Click on the Network Tab
- Uncheck “Disable cache”
- Click on the Timing tab
- Interact with the web application and record the number of requests made and the load time.
    - For CSR, observe only the additional request/resource and record only the duration for the load time.