# **Accounts Service Design Approach**

# Requirement Details

* Develop small app **within ReactJS** mentioned with problem statement mentioned below:
  + Create a react UI based application. Which will be showing expense and earnings of user. (Create a Login Mechanism)
  + Pie chart showing different categories on which user has spent the money.
  + A Bar Chart displaying Last 1 Year data of Earning & Expenses.
  + ​List of Recent Transactions done by user.
  + A rating Meter representing if user is having,​Good - Bad spending habits on scale of 1-10
  + **Note: You can use mocks for backend services.**

# [Solution Design](https://softwaredominos.com/home/software-design-development-articles/high-level-solution-design-documents-what-is-it-and-when-do-you-need-one/)

This application created in two parts

**Server Application:**

* Server Application created using Spring Boot with single REST API for user login.
* Login API used to login the user and get back logged in user full data or error message

**API Information:**

API URL : <http://localhost:8080/account-service/v1/user/login>

HTTP Methos : POST

Input : { "username":"User1@test.com", "password":"User1"}

Output : User all information in JSON format

**ReactJS Application:**

* Client-side UI Application created using ReactJS, ChartJS and Bootstrap CSS
* Application is running on this URL :http://localhost:3000
* Check User is logged in or not if user not logged in then show login page
* All the routs URL protected only login URL is open
* User need to enter user name and password
* User name and password has client-side validation
* After clicking on login button server login REST API will call in async
* Application receives error or success response
* In success response has logged in user all information to show in dashboard
* This information stored in Context API at client-side. When this information required, we can get using context useContext hook in all components
* Created costume hook useAuth to store login user in localStorage
* Using useAuth check user is logged in or not every time
* Created two tabs

**Dashboard**

* Pie chart showing different categories on which user has spent the money
* A Bar Chart displaying Last 1 Year data of Earning & Expenses
  + These both charts created using CharJS

**Spending money**

* List of Recent Transactions done by user.
  + This is created using nested components.
  + Pass data from parent to child using Props
* A rating Meter representing if user is having,​Good - Bad spending habits on scale of 1-10
* After clicking on logout link user will be logged out

# [Input](https://softwaredominos.com/home/software-design-development-articles/high-level-solution-design-documents-what-is-it-and-when-do-you-need-one/) data

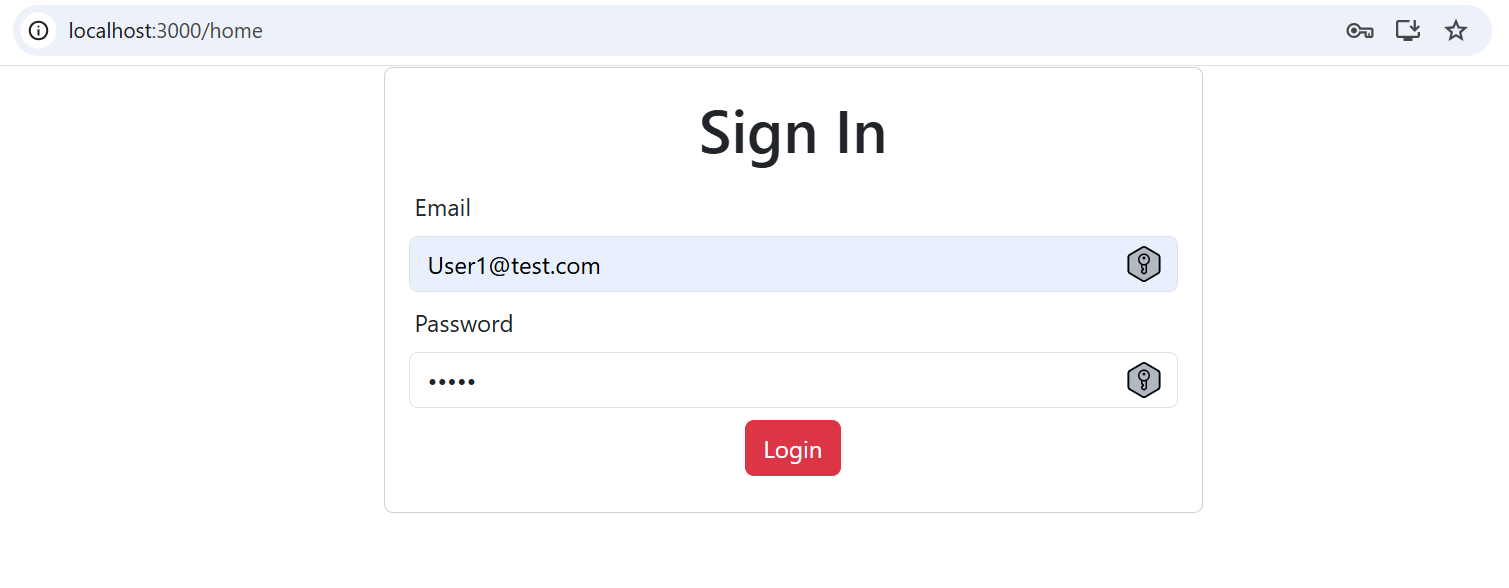
|  |  |
| --- | --- |
| **User Name** | **Passwod** |
| User1@test.com | User1 |
| User2@test.com | User2 |
| User3@test.com | User3 |
| User4@test.com | User4 |

# Setup Requirements

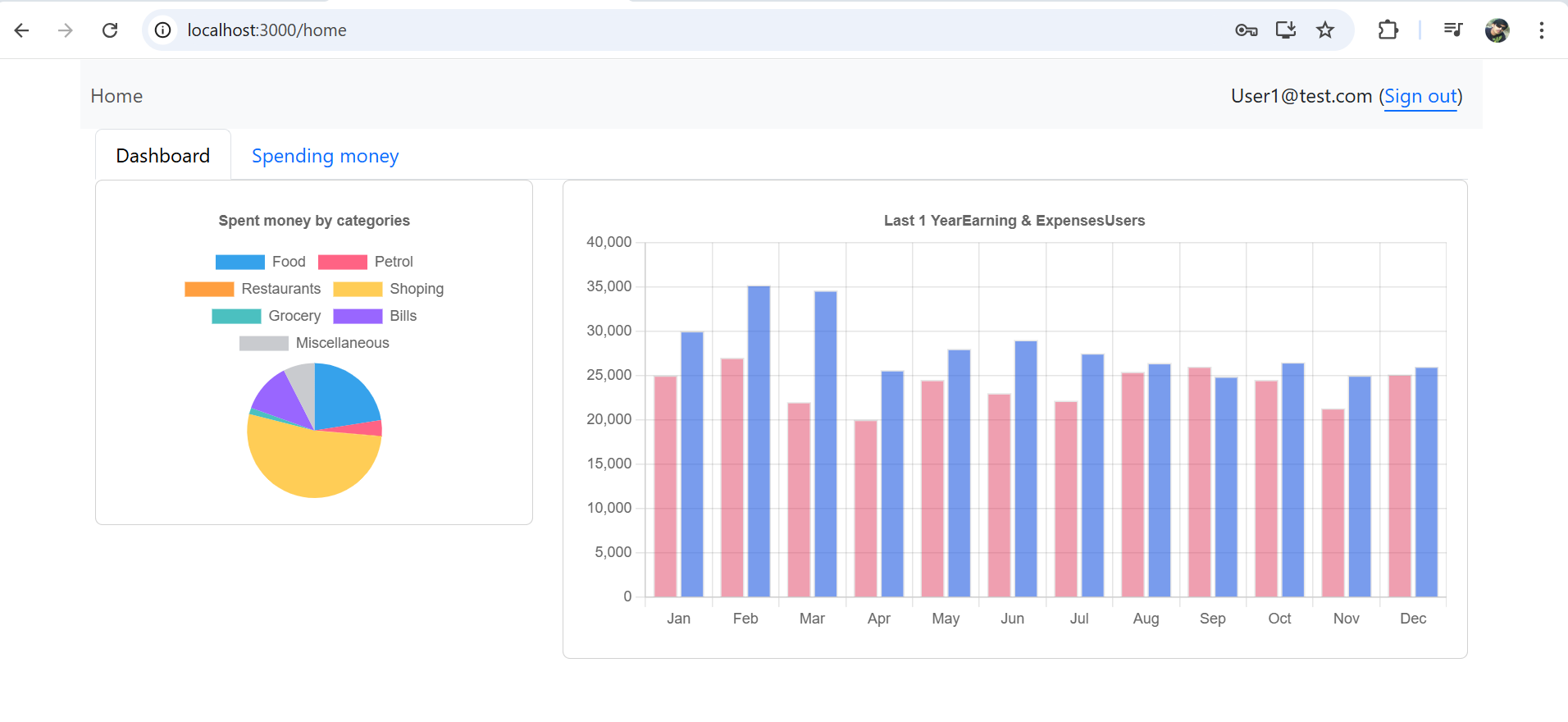
* STS, Java 21, NodeJS, VS Code, React

# Application Screens

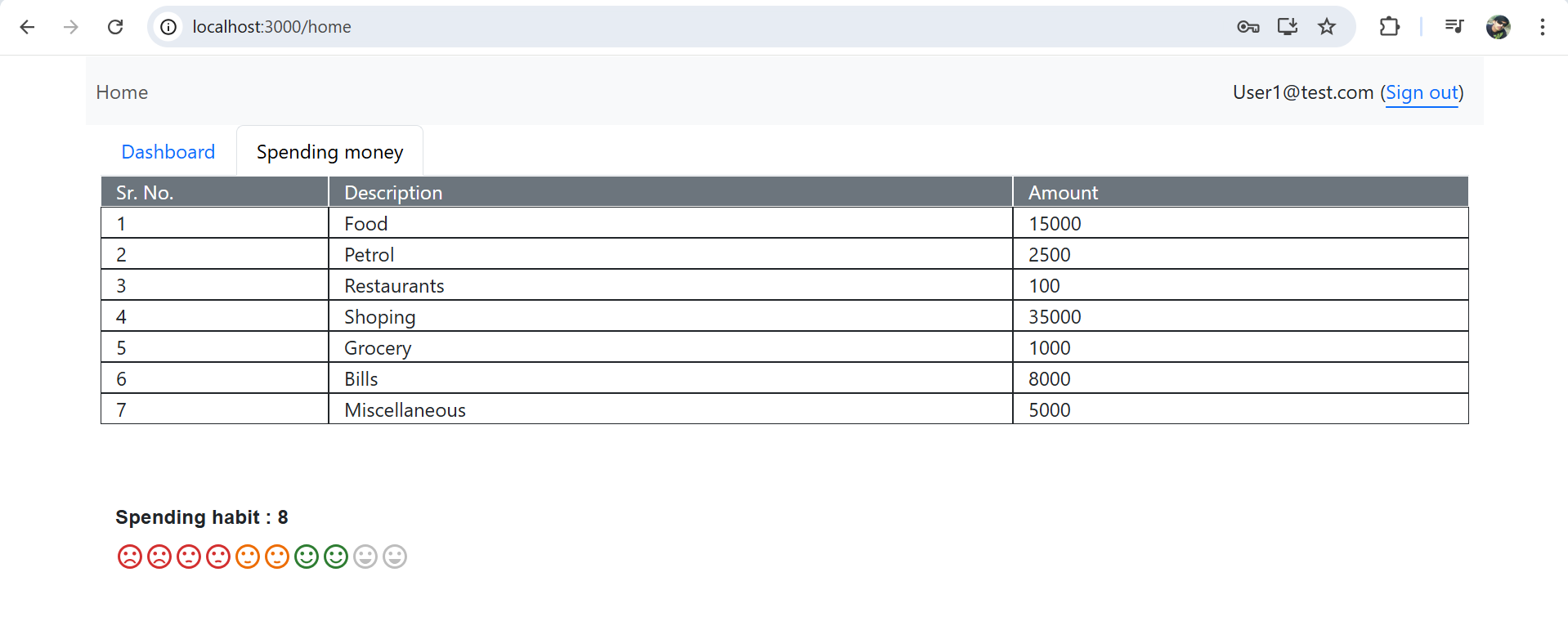
* Login Page



* Home page – Dashboard tab



* Home page – Spending Money tab



* After Log out screen

