EMPLOYEE MANAGEMENT

Introduction:

The purpose of this report is to provide an overview of an Employee Management System developed using Angular 15, Angular Material, and a JSON Server as the database. The system is designed to help organizations efficiently manage their employee data, including personal information, job details, and performance records.

System Architecture:

The Employee Management System is built using the following technologies:

Angular 15:

Angular is a popular JavaScript framework for building web applications. Angular 15 provides a robust and scalable platform for developing complex applications with improved performance and enhanced features.

Angular Material:

Angular Material is a UI component library that provides pre-built UI components and styles for Angular applications. It offers a set of reusable components such as forms, buttons, tables, and modals, which can be easily integrated into the application.

JSON Server:

JSON Server is a lightweight RESTful API server that allows data storage and retrieval using a JSON file as the database. It provides a simple and convenient way to simulate a backend server for testing and development purposes.

System Features:

The Employee Management System offers the following key features:

Employee CRUD Operations:

The system allows creating, reading, updating, and deleting employee records. Users can add new employees, view existing employee details, update employee information, and delete employee records as needed.

Employee Search:

The system provides a search functionality to quickly locate specific employees based on criteria such as name, department, or job title. This feature enables easy retrieval of employee information, especially in large organizations with a vast employee database.

User Authentication and Authorization:

The system includes user authentication and authorization mechanisms to ensure data security and restrict access to sensitive information. Only authorized personnel can perform administrative tasks, such as adding or updating employee data.

Responsive UI Design:

The system is built with a responsive UI design using Angular Material components, ensuring a consistent and user-friendly experience across different devices and screen sizes

Implementation Details:

To implement the Employee Management System, the following steps were taken:

Setting up Angular 15:

Angular 15 was installed using Node Package Manager (npm) and configured for the project.

Angular Material Integration:

Angular Material was integrated into the Angular project to leverage its UI components and styles. This involved installing the necessary packages and importing the required modules.

JSON Server Setup:

JSON Server was set up by creating a JSON file to serve as the database and defining RESTful API endpoints for performing CRUD operations on employee data.

Building Components:

Angular components were developed to handle employee-related operations, including listing employees, adding new employees, updating employee details, and displaying performance information.

Implementing Services:

Angular services were created to interact with the JSON Server API endpoints. These services handle data retrieval, data manipulation, and HTTP requests.

Implementing User Authentication:

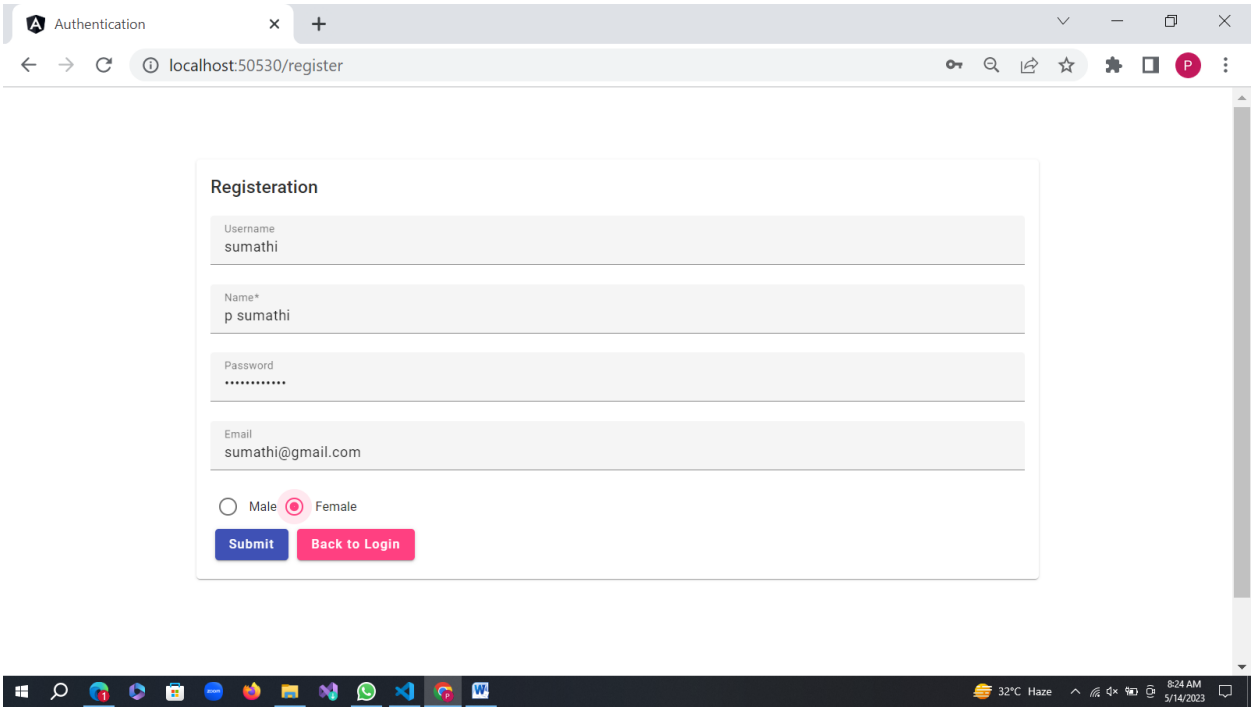
User authentication functionality was implemented using Angular's built-in authentication mechanisms or third-party libraries, depending on the specific requirements of the system.

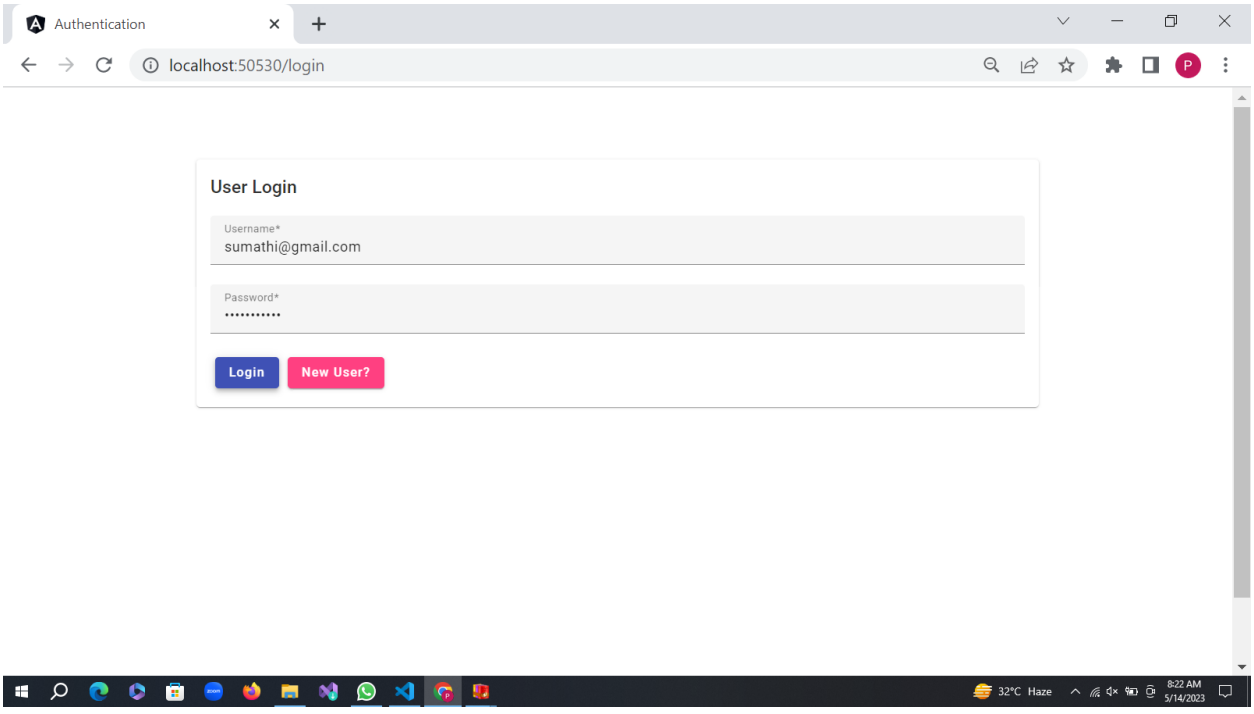
Conclusion:

The Employee Management System developed using Angular 15, Angular Material, and JSON Server provides an efficient solution for managing employee data within organizations. The system offers essential features such as CRUD operations, search functionality and user authentication. The responsive UI design ensures a seamless experience across different devices. With its scalable architecture and user-friendly interface.

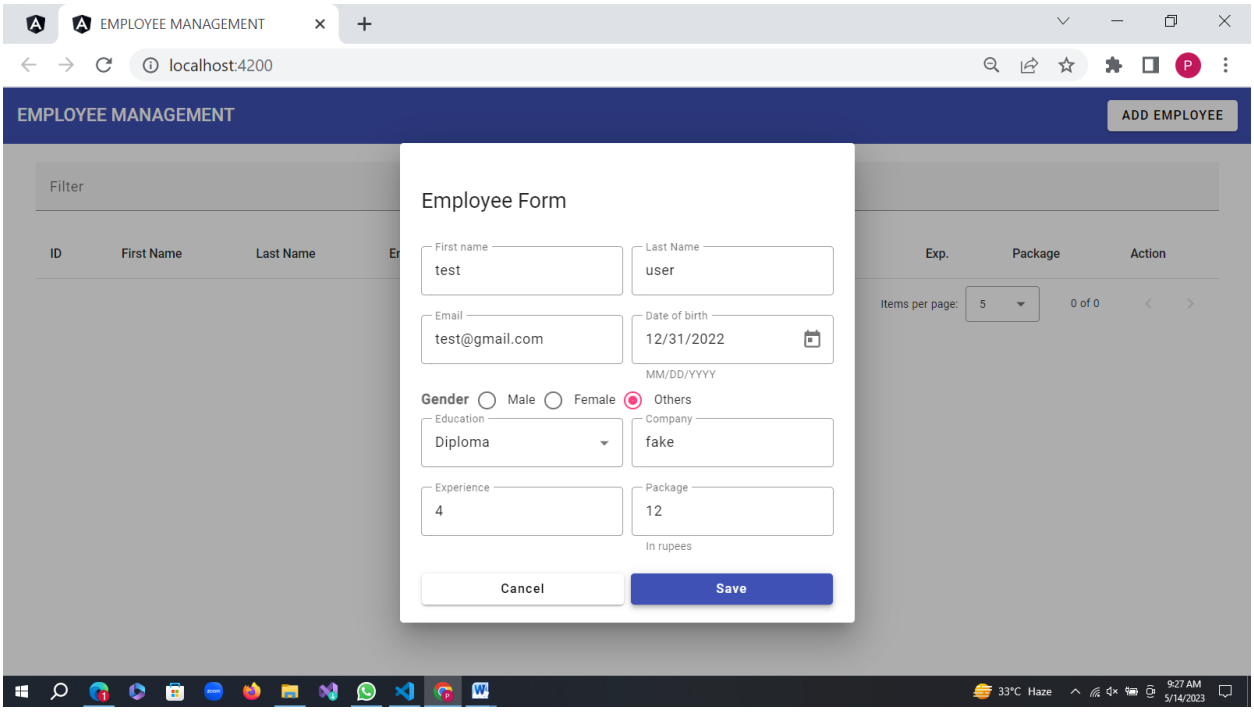
OUTPUT:

USER REGISTRATION AND LOGIN

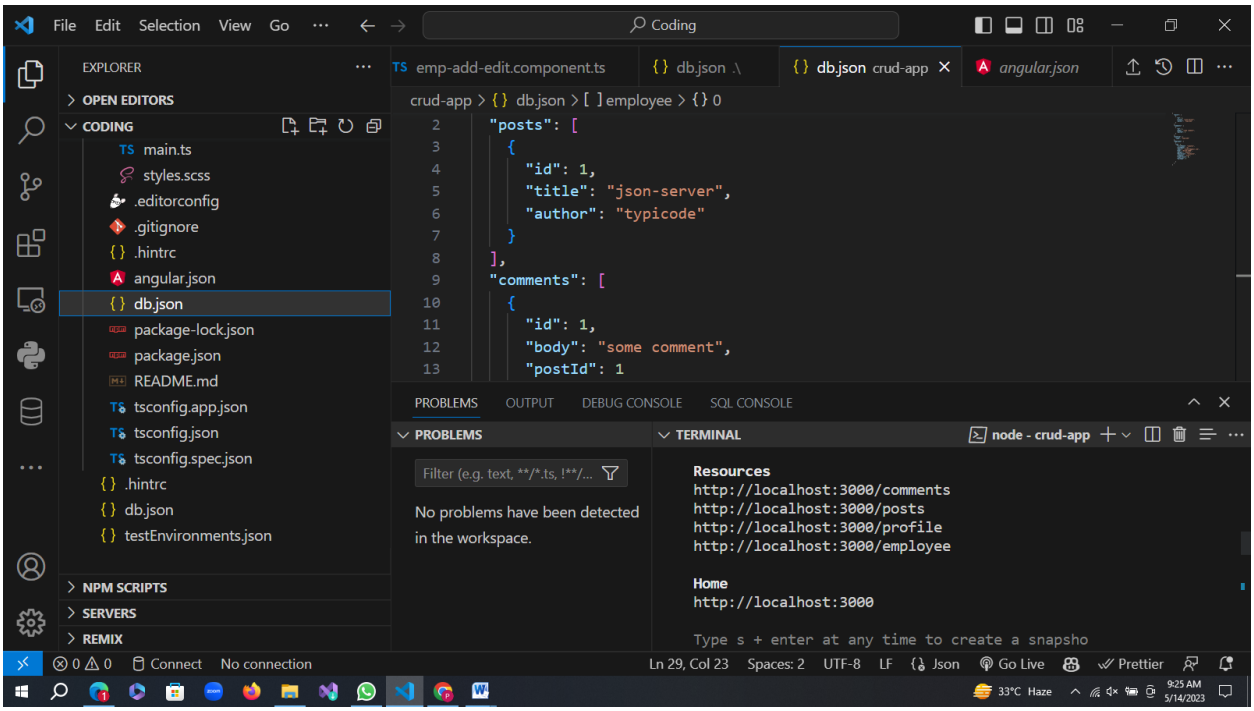




CRUD OPERATIONS

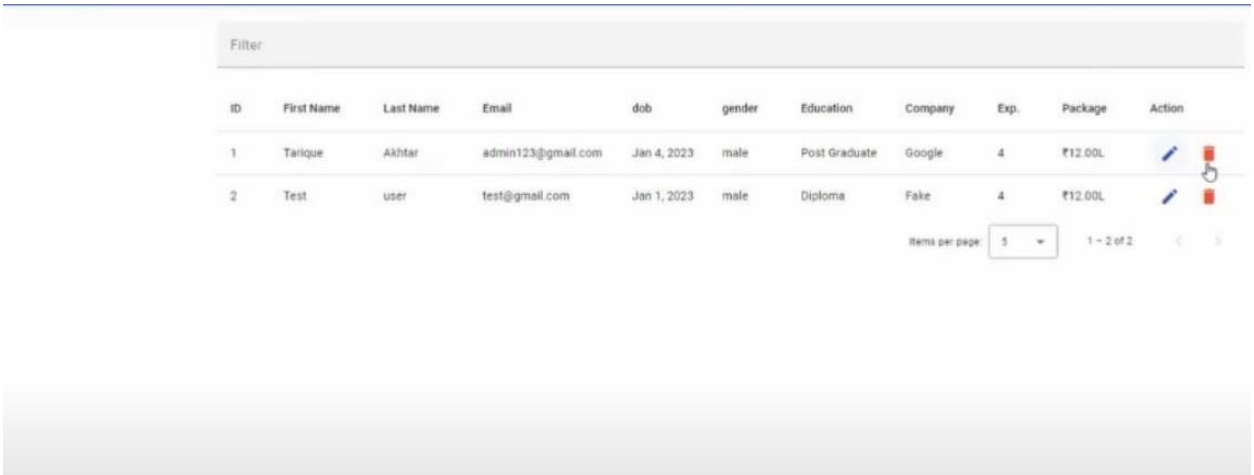


We can Update, Create, Delete, Add employees and the data will be stored in the Jason.db.



We can see the details of Employee, Posts, Profiles and Comments with the help of localhost.

In localhost employee details will be found and the data will be stored in db.json.



We can delete, add and update the details of the employee.