Full-stack Employee Management System (Java + Spring Boot + Angular)

A full-stack Employee Management System with role-based access using Spring Boot, JWT Authentication, and Angular.

Admins can manage employee records (Create, Read, Update, Delete), while Users can only view the employee list.

Features:

* Login & Registration
* Role-based Access (Admin/User)
* JWT Authentication
* Employee CRUD Operations (Admin only)
* Employee Listing (All roles)
* RESTful API (Spring Boot)
* Angular Frontend with Bootstrap
* Spring Security Integration
* Session Management
* Pagination & Search- Optional: Docker Support

Technologies Used:

Backend: Java 17, Spring Boot 3, Spring Data JPA, Spring Security, MySQL/PostgreSQL

Frontend: Angular 15+, Bootstrap 5, HTML/CSS, Angular Routing

Other: JWT, Lombok, Model Mapper, Docker (optional)

Project Structure:

employee-management-system/

├── backend/ (Spring Boot)

│ ├── controller/

│ ├── entity/

│ ├── repository/

│ ├── service/

│ └── security/

├── frontend/ (Angular)

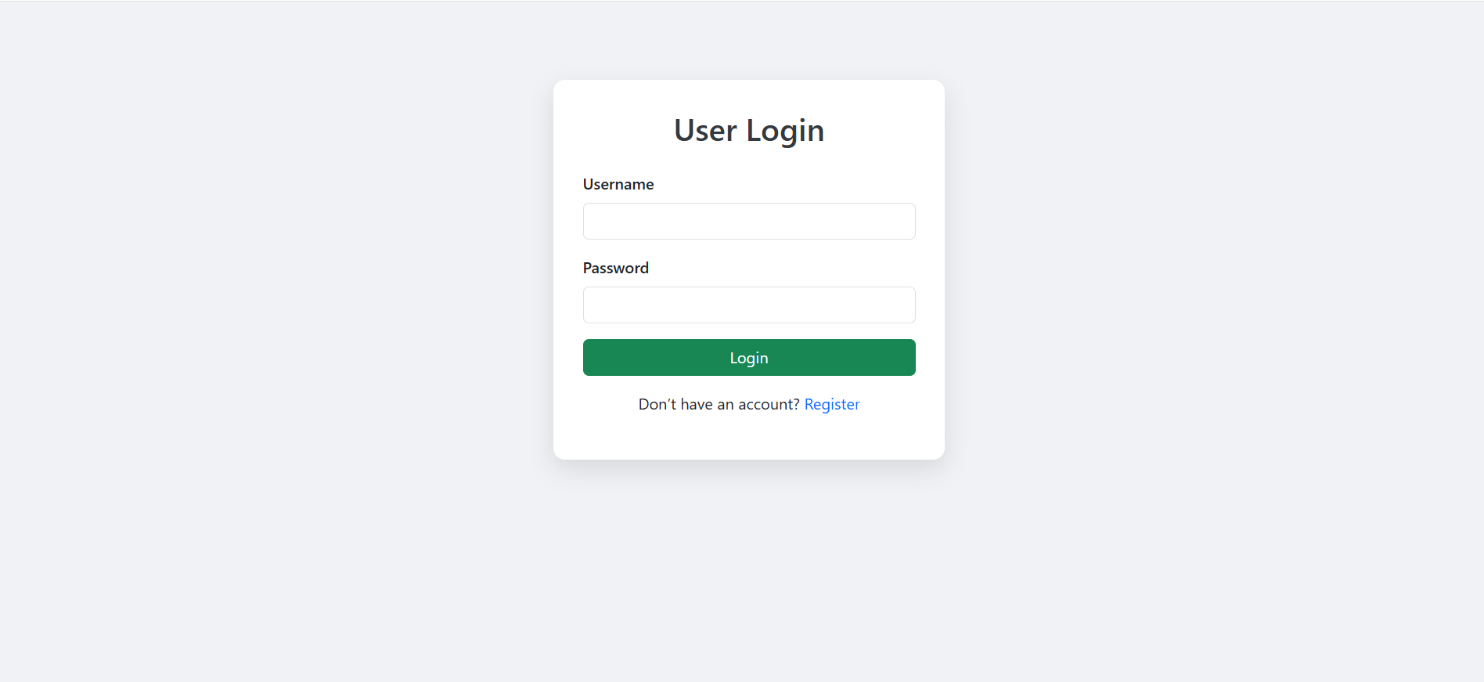
│ ├── components/

│ ├── services/

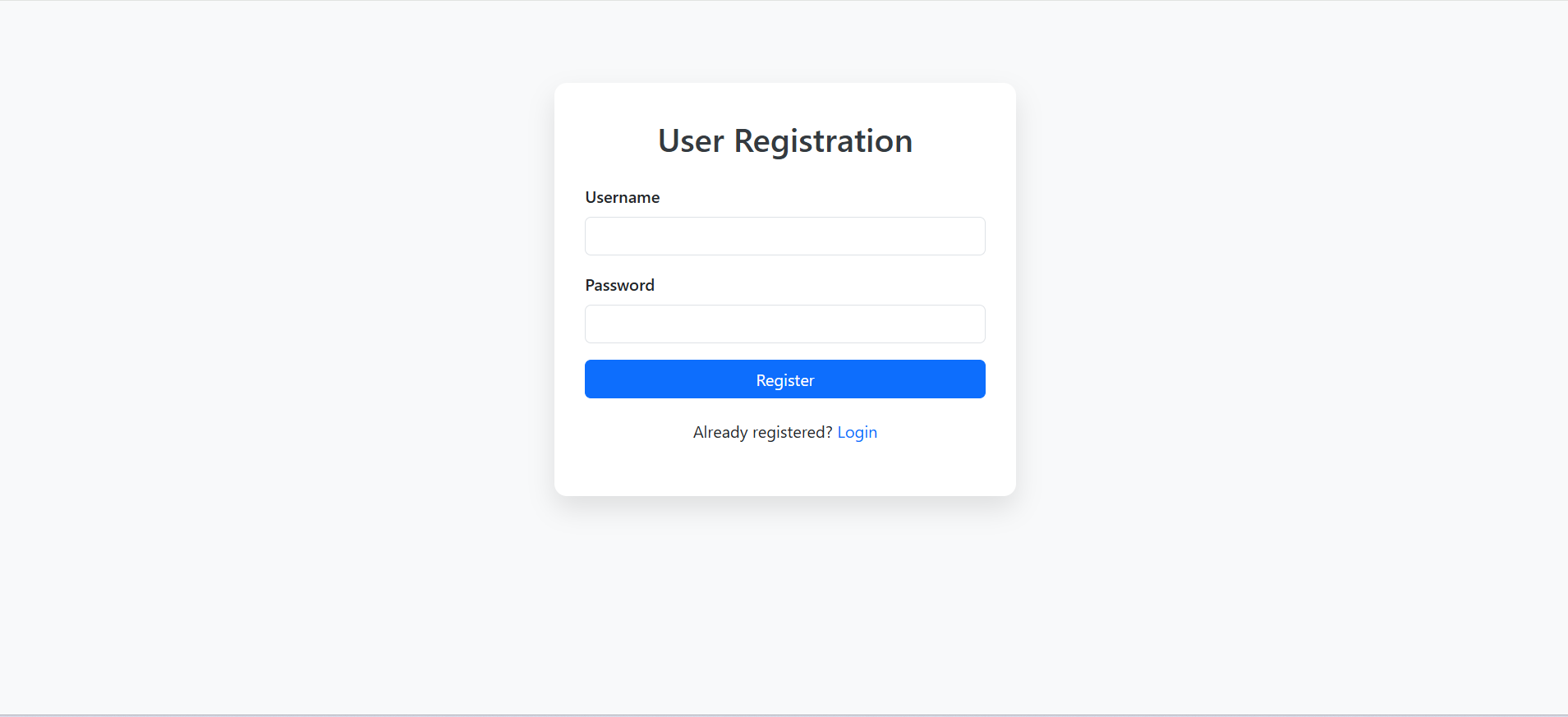
│ └── models/

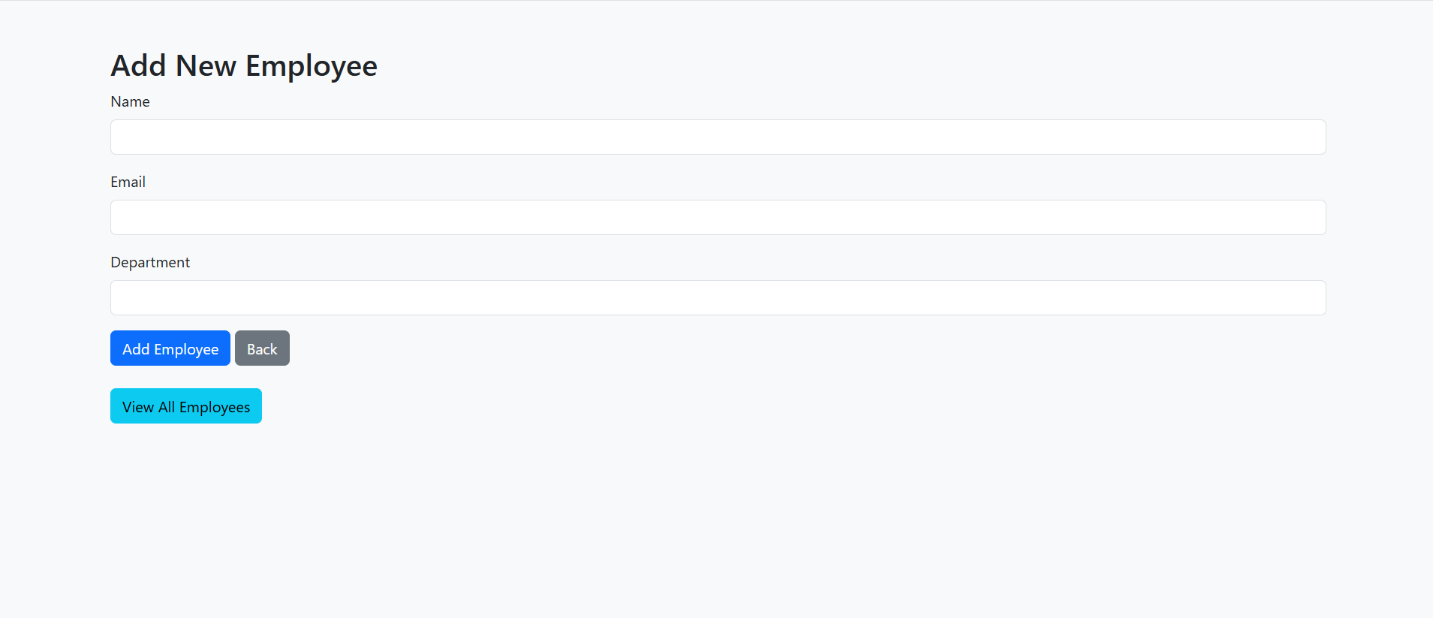
Screenshots:

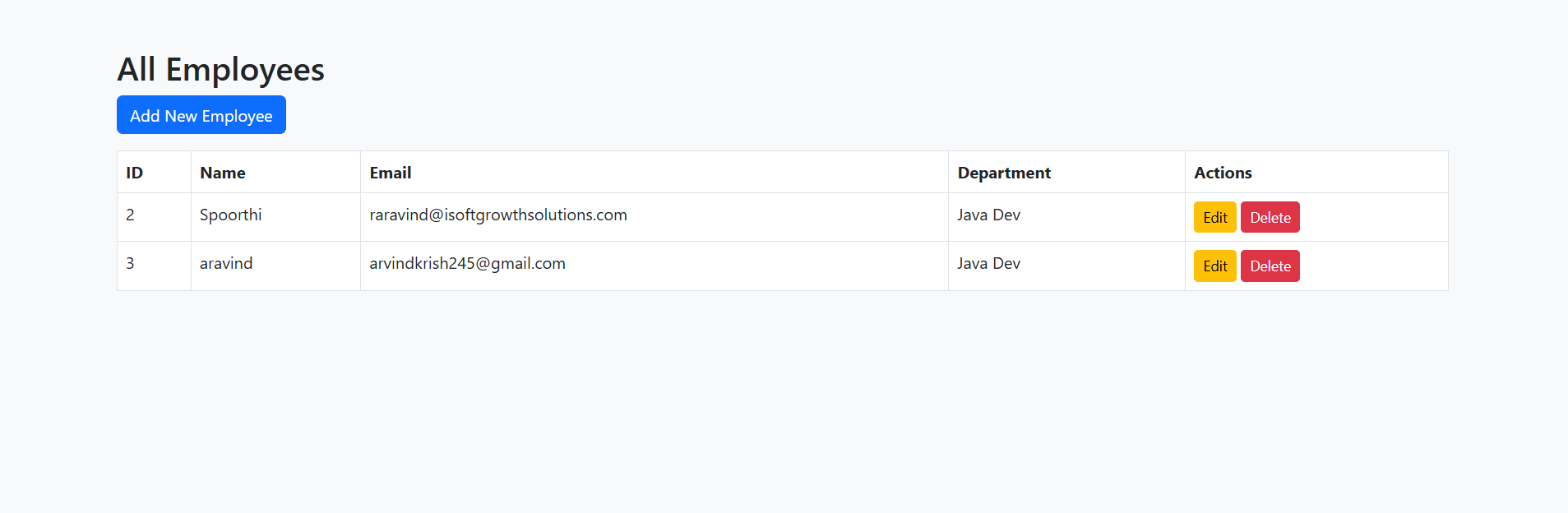
* Login Page:



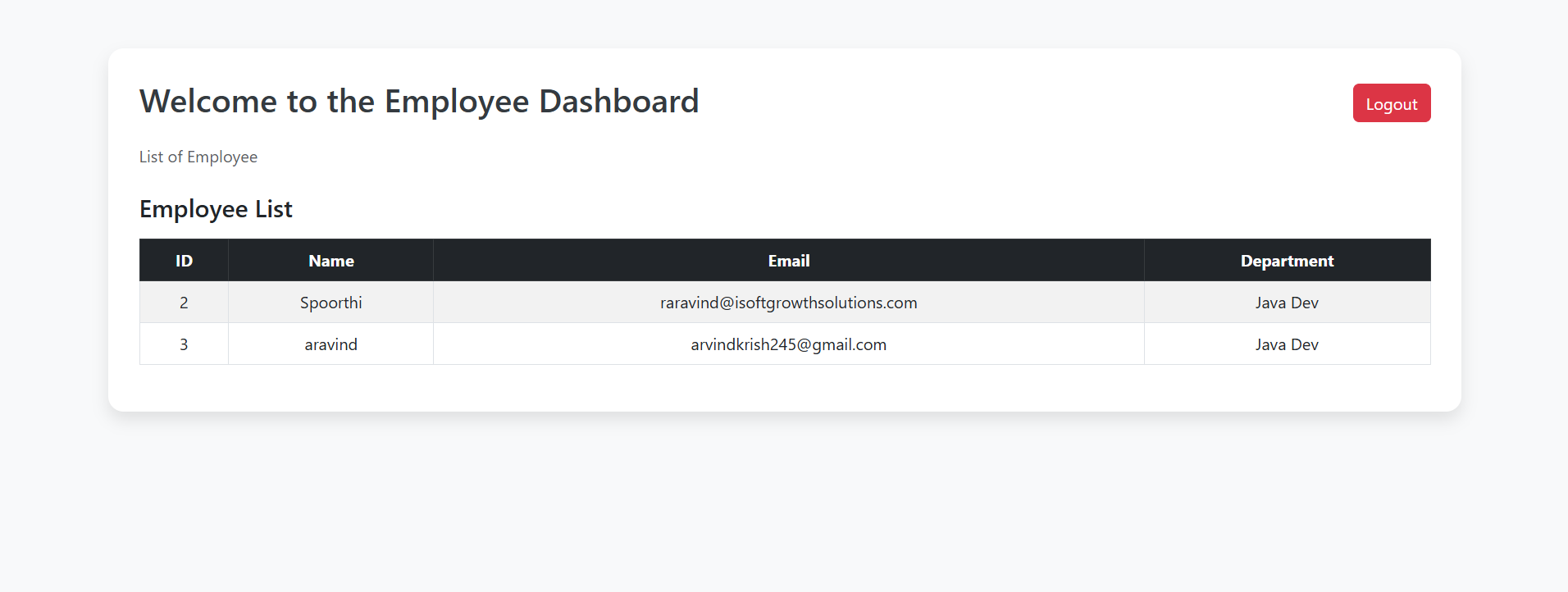
* Register Page:



* Admin Dashboard - 



* User Dashboard - View Only:



Backend Setup (Spring Boot):

1. Clone the repo:

git clone https://github.com/aravindrakonda83/Full-stack-Employee-Management-System-using-Java-Spri ng-Boot-and-Angular.git

1. Import into IntelliJ / Eclipse.
2. Configure application.properties with DB credentials.
3. Run with: ./mvnw spring-boot:run

Frontend Setup (Angular):

1. cd frontend
2. npm install
3. ng serve
4. Open http://localhost:4200

Default Users:

Username | Password | Role

admin | admin123 | ADMIN user | user123 | USER

Deployment: (Optional)

* Dockerize both frontend and backend or deploy to Heroku/Netlify

Contributing:

* Fork, branch, and pull request

License: - MIT License

Author:

Aravind Rakonda

GitHub: @aravindrakonda83