**Application: TO-DO-List**

Technologies Used :

* Angular 15, Typescript, HTML, CSS
* Spring Boot Framework (3.0.1), Java Version:17, H2 Database

I have worked in spring boot 2 and angular versions like 2,8, and 9. So I have used the same tech stack for creating this application.

**Backend Application:**

The spring boot application is created from scratch and configured for the H2 database in the application.properties. The application runs on port 8081. Data Transfer Object design pattern is used for the Request (TaskRequest) and Response (TaskResponse) Object. Entity Pattern creates the table structure with the fields and getter and setter methods (TaskModel.java).JAP repository is used to perform the CRUD operations

The application developed has two components in general: Log in and To Do List (Create New, Update, Delete).

1. URL: / loginRequest

Input Parameters: Username and Password in Body for security purposes

Request Type: POST

1. URL: /retrievetask

Input Parameters: Username

Request Type: GET

This application has multiple users who keep track of their to-do-list. In order to provide the records of the logged-in user only, the username is taken as input to filter the records from the database.

1. URL: /task (New Record)

Input Parameters: New task information in the body

Request Type: POST

Initially, while creating a new record, the backend receives data such as task name and description in the body. As this is a new record, no “id” will be provided and generated while inserting the record in the table.

1. URL: /task (Update)

Input Parameters: Table row data in the body

Request Type: PUT

When the user the ‘edit’ option in the front-end and gives ‘update’, the entire row of data will be sent to the backend to update the details. In the Task Request model, a parameter is used to differentiate whether the request is a ‘new record’ or an existing one.

1. URL: /task (Delete)

Input Parameters: Unique ID of the row Data as RequestParam

Request Type: DELETE

The delete option is available in the front end to delete existing records.

**Table Structure:**

Each user can save a unique task. However, there is a scenario where different users can add the same task. To achieve this, both username and task name are kept as a combined unique constraint.

Id: Primary Key and auto-generated

**Architecture:**

**Diagram

Description automatically generated**

**Test Cases:**

**Test c**ases were implemented for the positive case scenarios for all the requests.

**Front End Application:**

The front-end application is developed using version 15. The angular application has two components: Log in and ToDoList. The user will be validated in the Login component along with the password. Once the user login is successful, the user will be redirected to the to-do list page. On that page, the user will be allowed to add new tasks, update the existing tasks and delete functionality.

Future Work:

1. Spring Security using the jwt token can be implemented. Currently, the security of the application is validated using the username and password authentication, I tried implementing JWT but faced some difficulties due to the version. Implement the same in the front-end application.
2. Tried implementing basic Authorization by importing spring-security package in pom.xml. But faced some issues while trying to hit POST methods.
3. Test cases for negative and different exceptions can be written in future.
4. By default, the spring boot application used a singleton design pattern. Different design patterns can be implemented, such as request, session or prototype.