**WeShare**

**Overview**

WeShare application allows registered users to share messages among them.

**Tasks involved**

1. Allows registered (pre-defined) users to login.

2. Able to post a message.

3. Able to see the messages posted by the user himself.

4. Able to see the messages posted by other users.

**Software Used**

1. Spring Boot REST Web Application

2. MySQL database

3. Spring JPA with Hibernate

4. JSP, JQuery-AJAX

**Database Structure**

Database name: weshare

Tables

1. USER

|  |  |
| --- | --- |
| Field Name | Type |
| ID | INT (Auto generated, Primary key) |
| USER\_NAME | VARCHAR |
| PASSWORD | VARCHAR |

2. MESSAGE

|  |  |
| --- | --- |
| Field Name | Type |
| ID | INT (Auto generated, Primary Key) |
| USER\_ID | INT NOT NULL (Foreign Key) |
| MESSAGE | VARCHAR |
| CREATED\_DATE | DATE |

**Project Structure**

**1. Entity**

* + User

Maps to "USER " table.

* Message

Maps to "MESSAGE " table.

Many-To-One mapping to USER table (Column USER\_ID)

**2. Dto**

* MessageDto

DTO class, contains username, message, createdDate.

**3. Repository**

* UserRepository

1. Validates user login credentials.

findByUserNameAndPassword

2. Validates valid user.

findById

* MessageRepository

1. Add a new message (save)

2. Gives a list of messages posted by the user.

findByUserIdOrderByCreatedDateDesc

3. Gives a list of messages posted by other users. findByUserIdNotOrderByCreatedDateDesc

**4. Service**

* LoginService

Validates user login credentials using UserRespository.

* MessageService

1. Add a new message using MessageRespository.

2. Gives a list of messages posted by the user using MessageRespository.

3. Gives a list of messages posted by others using MessageRespository.

**5. WeShareController (has following end points)**

* GET: http://localhost:8080/login

returns login.jsp

* POST: http://localhost:8080/login

Validates user credentials using LoginService

For valid credential, returns "welcome.jsp"

For invalid credential, shows error on the login page

* POST: http://localhost:8080/postmsg/{id}

Uses ModelMapper to convert MessageDto to Message Entity.

Add the new posted message using MessageService.add

Returns "Success" or "Failure" message

* GET: http://localhost:8080/viewmsg/{id}

Uses ModelMapper to convert MessageEntity to Message Dto.

Returns the posted messages using MessageService getAllMessagesByUser as JSON objects.

* GET: http://localhost:8080/viewothermsg/{id}

Uses ModelMapper to convert MessageEntity to Message Dto.

Returns the posted messages using MessageService getAllMessagesByOthers as JSON objects.

**6. SpringBootApplication**

WeShareApplication

**7. Login.jsp**

Accepts User ID, Password

On Submit, send "POST" request

**8. Welcome.jsp**

Displays

1. Post Message

Click on this, shows text box and button to post new messages

2. My Messages

Makes AJAX request (using JQuery) to "/viewmsg"

Parses JSON reply message and displays

3. Others Messages

Makes AJAX request (using JQuery) to "/viewothermsg"

Parses JSON reply message and displays