

# SD Assignment 3

Goina Radu  
group: 30432

# Contents

<b>1</b>	<b>Assignment 3</b>	<b>2</b>
1.1	Assignment objective . . . . .	2
<b>2</b>	<b>Intro</b>	<b>3</b>
<b>3</b>	<b>Feature 1</b>	<b>5</b>
3.1	Requirements . . . . .	5
3.2	Implementation . . . . .	6
<b>4</b>	<b>Feature 2</b>	<b>7</b>
4.1	Requirements . . . . .	7
4.2	Implementation . . . . .	8
<b>5</b>	<b>Feature 3</b>	<b>9</b>
5.1	Requirements . . . . .	9
5.2	Implementation . . . . .	10
<b>6</b>	<b>Bonus feature 1</b>	<b>11</b>
6.1	Requirements . . . . .	11
6.2	Implementation . . . . .	11
<b>7</b>	<b>Bonus feature 2</b>	<b>12</b>
7.1	Requirements . . . . .	12
7.2	Implementation . . . . .	13
<b>8</b>	<b>Conclusion</b>	<b>16</b>

# Chapter 1

## Assignment 3

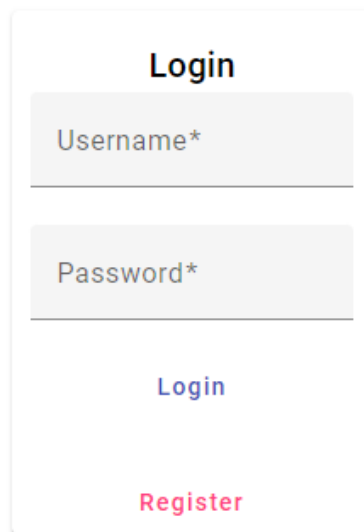
### 1.1 Assignment objective

The objective of this assignment was to connect the BE and the FE and implement all the features our application should have

# Chapter 2

## Intro

The authentication part of my website is implemented using Spring Security, and users receive authorization through JWT Bearer tokens.

A login form with a white background and rounded corners. At the top, the word "Login" is centered in bold black text. Below it are two input fields: "Username\*" and "Password\*", both with light gray backgrounds and thin black borders. At the bottom, there are two buttons: "Login" in blue text and "Register" in red text, both centered.

**Login**

Username\*

Password\*

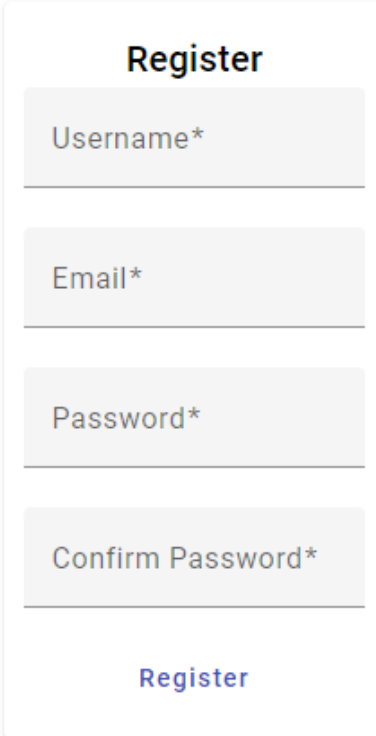
Login

Register

If the user does not have an account already he can create one by clicking the register button which will redirect him to the register page.

Once the register button is clicked the request will be sent to the backend via the `/users/addUser` endpoint.

The passwords are stored encrypted in the database.

A vertical registration form with a white background and rounded corners. At the top, the word "Register" is centered in bold black text. Below it are four input fields, each with a light gray background and a thin border. The first field is labeled "Username\*", the second "Email\*", the third "Password\*", and the fourth "Confirm Password\*". At the bottom of the form is a blue "Register" button.

**Register**

Username\*

Email\*

Password\*

Confirm Password\*

**Register**

# Chapter 3

## Feature 1

### 3.1 Requirements

Feature one had the following requirements

- Users should be able to ask questions and assign tags to them
- Newest questions must be displayed first
- Questions should be able to be deleted by their author
- User should be able to filter questions

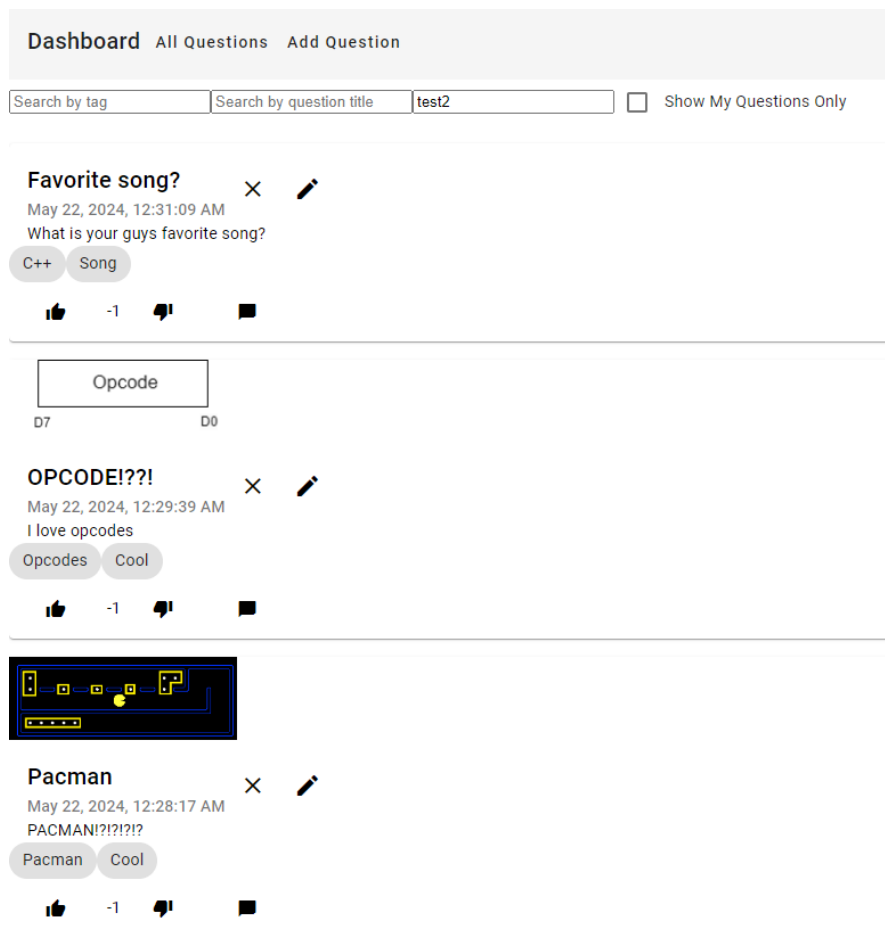
## 3.2 Implementation

Questions are retrieved from the backend by calling the `/questions/getAll` endpoint.

The images are retrieved from the backend by calling the `/images/image.path` endpoint, where `image.path` is the path for each image. This is done using the help of an angular service, the same service is responsible for fetching the images for the answers

Filtering is done in the FE

A user is able to delete ONLY their question, BE checks are implemented that prevent the deletion of other questions.



# Chapter 4

## Feature 2

### 4.1 Requirements

Feature two had the following requirements:

- Users should be able to add answers to a question
- Users should be able to delete their own answer



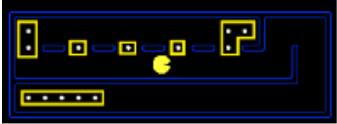
## 4.2 Implementation

Answers are retrieved from the BE by calling the `/answers/question/id` endpoint where `id` is the current question's ID.


Same method of fetching images as for the questions.

BE checks are implemented to assure that an answer can only be edited/deleted by their author.

**Dashboard** All Questions Add Question



**Pacman**  
test2 : -4.5  
May 22, 2024, 12:28:17 AM  
PACMAN!?!?!?  
Pacman Cool  
0  
Answer!



radug : 13.5  
May 21, 2024, 9:32:39 PM  
This is not a CS question  
0

# Chapter 5

## Feature 3

### 5.1 Requirements

Feature three had the following requirements:

- Users should be able to Like/Dislike any question/answer
- The vote count must be displayed for every question/answer
- answers should be sorted by vote count

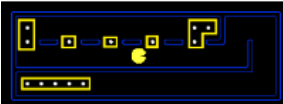
## 5.2 Implementation

The user's vote for each question is fetched via the `/user/userQuestionRating/questionId` endpoint, and for each answer via the `/user/userAnswerRating/answerId` endpoint.

The overall vote count also fetched from the BE.

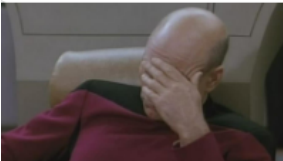
The answers are sorted by vote count in the FE.

**Dashboard** All Questions Add Question



**Pacman**  
**test2 : -4.5**  
May 22, 2024, 12:28:17 AM  
PACMAN!?!?!?!?  
Pacman Cool  
👍 -1 Answer!

**radug : 11**  
May 21, 2024, 10:46:37 PM  
I have more votes so Im first!  
👍 1 👎



**radug : 11**  
May 21, 2024, 9:32:39 PM  
This is not a CS question  
👍 -1 👎

# Chapter 6

## Bonus feature 1

### 6.1 Requirements

Bonus feature one requires that a score be computed for each user according the the following rules

Users gain points when

- Their question is voted up (2.5 points per vote)
- Their answer is voted up (5 points per vote)

Users lose points when

- Their question is voted down (-1.5 points per vote)
- Their answer is voted down (-2.5 points per vote)
- They down vote an answer of another user (-1.5 point)

### 6.2 Implementation

The overall score is received from the BE via the `getUserScore/userId` endpoint. The computation is done in an SQL query. Implementation can be seen in the image above

# Chapter 7

## Bonus feature 2

### 7.1 Requirements

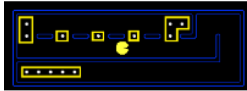
The second bonus feature has the following requirements:



- Add admins, who are users with special privileges
- An admin should be able to edit/remove any question/answer
- An admin should be able to ban/unban users
- Banned users should be able to see that they are banned when trying to log in.
- Banned users should not be able to do anything on the website
- Banned users must receive an email when they are banned



## 7.2 Implementation



Admins ignore the BE constraints for updating/removing post so they can update/remove any post.

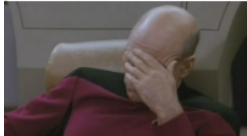
[Dashboard](#) [All Questions](#) [Add Question](#) [User Management](#)





**Pacman**  
[test2 : -4.5](#)  
May 22, 2024, 12:28:17 AM  
PACMAN!?!?!?  
[Pacman](#) [Cool](#)  
  -1 [Answer!](#)









[radug : 11](#)  
May 21, 2024, 10:46:37 PM  
I have more votes so Im first!  
 1  [×](#) [✎](#)

[test2 : -4.5](#)  
May 21, 2024, 11:10:41 PM  
Im a different user  
 0  [×](#) [✎](#)



[radug : 11](#)  
May 21, 2024, 9:32:39 PM  
This is not a CS question  
  -1 [×](#) [✎](#)

Moreover admins have access to the user management page, Where they can ban/unban users.

Dashboard		All Questions	Add Question	User Management
test_admin  				
User Id: 4				
User Join Date: 2012-03-16T16:03:24.000+00:00				
Banned: 0				
Email: test@example.com				
test2  				
User Id: 18				
User Join Date: 2024-05-18T22:14:54.000+00:00				
Banned: 0				
Email: test2@example.com				
test3  				
User Id: 19				
User Join Date: 2024-05-21T19:32:18.000+00:00				
Banned: 1				
Email: test3@example.com				
radug  				
User Id: 23				
User Join Date: 2024-05-21T20:25:37.000+00:00				
Banned: 0				

Once a user is banned they will receive an email

- Totally legit stackoverflow BAN



You got BANNED from totally legit stackoverflow

And if they try to log in they will receive the following message.

A screenshot of a web login interface. At the top, a red banner with white text reads 'Your account has been banned.'. Below this is a white box with a grey border. Inside the box, the word 'Login' is centered at the top. Below it are two input fields: 'Username\*' containing the text 'radug' and 'Password\*' containing four dots. At the bottom of the box are two links: 'Login' in blue and 'Register' in red.

Being banned will prevent the user from receiving the authorization token, therefore they are unable to retrieve/post anything from the BE.



## Chapter 8

### Conclusion

To conclude, I successfully created a StackOverflow-like website that implements all the required and bonus features.