

SWE 573: Final Report

Mehmet Emre Berk

Lecturer: Asst. Prof. Suzan Üsküdarlı

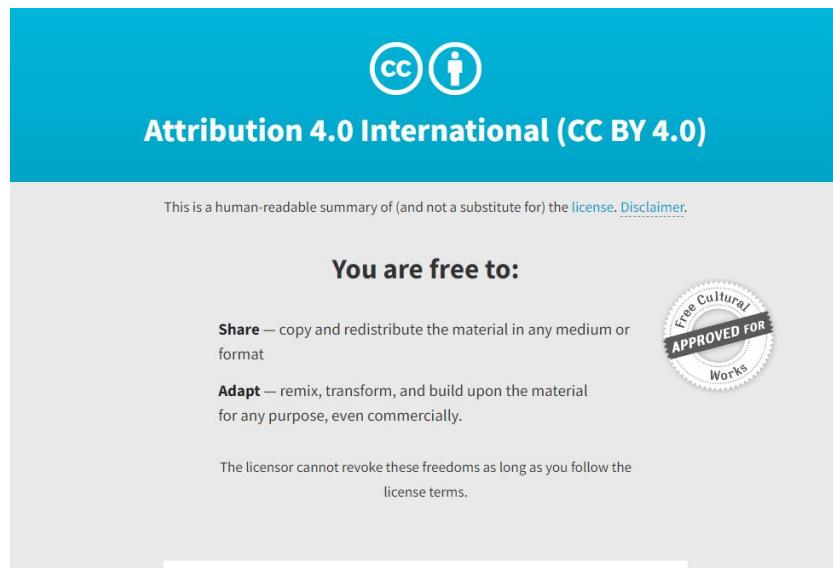
Course: Software Development Practice

Project: Social Bookmarking Web Application

Git tag version: v0.9

January 2, 2023

License



You are free to:

Share — copy and redistribute the material in any medium or format



Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:



Attribution — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable [exception or limitation](#).

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as [publicity, privacy, or moral rights](#) may limit how you use the material.

Git Repository and Deployment

* Git repository: <https://github.com/emreberk92/SWE573---SOFTWARE-DEVELOPMENT-PRACTICE.git>

* Git tag version: v0.9 (Please check v0.9.1, if you'd like to see the code with license.)

* Deployment url: <http://ec2-34-238-190-53.compute-1.amazonaws.com:8000> (Please note that deployment is not working properly due to an authorization error. Please check chapter 5 for the detailed information.)

Honor Code

Related to the submission of all the project deliverables for the Swe573 2022 Fall semester project reported in this report, I Mehmet Emre Berk declare that:

- I am a student in the Software Engineering MS program at Bogazici University and am registered for Swe573 course during the 2022 Fall semester.
- All the material that I am submitting related to my project (including but not limited to the project repository, the final project report, and supplementary documents) have been exclusively prepared by myself.
- I have prepared this material individually without the assistance of anyone else with the exception of permitted peer assistance which I have explicitly disclosed in this report.
- I hereby declare that all work in the project I am submitting is performed by myself.

Mehmet Emre Berk

TABLE OF CONTENTS

License	i
Git Repository and Deployment	ii
Honor Code	iii
1. Overview	v
2. Software Requirements Specification	vi
3. Software and Mockups Designs [Please insert into preamble] UML diagrams and Images	ix
4. Status of the project	xiv
5. Status of Deployment	xv
6. System manual	xvii
7. User manual	xviii
8. Test results	xix
REFERENCES	xxiii

1. Overview

The main objective of this course is to create Pocket alike web application, which allows to create posts, search and save articles, videos, and other types of web content for later viewing.

This web application is a great tool for saving and organizing content that is desired to read or watch later, especially if users don't have time to consume it all at once. It has ideally variety of features, including the ability to tag and organize the saved content, as well as recommendations for other content users might enjoy based on your saved items.

To create this application, the following aspects are covered:

- User scenario: to understand the needs and goals of the people who will be using this application.
- Elicitation Questions: to uncover the specific needs and requirements of the user group that the application is being developed for.
- Requirements: to identify the needs and expectations that our application is expected to fulfill.
- Mock-up design: to demonstrate the layout and design of our web application.
- Create backend and frontend using Django and HTLM/CSS
- Dockerizing the application to run and manage applications in a consistent environment regardless of the underlying infrastructure.
- Deployment in AWS to deliver the application to the end users.

2. Software Requirements Specification

In web development, requirements are the specific needs and expectations that a website is expected to fulfill. These requirements can be functional (e.g. the website must allow users to create accounts) or non-functional (e.g. the website must load within 3 seconds). Requirements are typically gathered from stakeholders such as the business owner, users, and subject matter experts, and are used to guide the design and development of the website.

Gathering requirements is an important step in the web development process because it helps to ensure that the final product meets the needs of the target audience and the business. Without a clear understanding of the requirements, it can be difficult for designers and developers to create a website that is effective and meets the goals of the project. Requirements can be gathered through techniques such as user interviews, focus groups, and elicitation questions.

The requirements are categorized based on their field of use:

(i) Non-functional requirements:

- 1.1 The application shall be available on macOS, Windows, iOS and Android
- 1.2 The application shall find the posts via link instead of copying that.

(ii) Login/sign-up features:

- 2.1 The user shall register in order to login the application.
- 2.2 The application shall ask the user username, e-mail address, password with verification and date of birth in order to register.
- 2.3 Where the user signs up, the application shall allow user to signup via e-mail or username, or Facebook account.
- 2.4 When user select to signup via e-mail, the application shall ask user to enter an email in the correct form.
- 2.5 When user select to signup via username, the application shall ask user to

enter a user name.

- 2.6 If the email address is not in the correct form, then the application shall display warning message.
- 2.7 If the password is not in the correct form, then the application shall show warning message to user for a new one.
- 2.8 When user clicks on sign in button, the application shall check user information.
- 2.9 If the email address is not valid, then the application shall ask for existing one.
- 2.10 If the password related to email address is not correct, then the application shall display warning message to the user.
- 2.11 The application shall ask username, email address, password with verification and date of birth to register.

(iii) Settings (Security, support and customization of the app)

- 3.1 The application shall have live support and FAQs under help section.
- 3.2 User shall have options to reach the help line by email and call under help menu.
- 3.3 The application shall have an option to choose the social media platforms to get the posts.
- 3.4 Notification menu shall be available under settings with option to have them by email or message.
- 3.5 The application shall offer privacy settings to customize the visibility of the profile as public, followers and private.
- 3.6 If user forgets his/her password, security menu shall let user to change the password.
- 3.7 When user wants to change his/her email address, security menu shall let user to change the email address.
- 3.8 When user wants to close his/her account, user shall be able to close the account in the privacy section in the settings.
- 3.9 When user wants to report or block accounts, user shall perform report and block actions in the privacy section in the settings.

(iv) Profile

- 4.1 The application shall bring profile menu after user creates an account.
- 4.2 The profile shall ask user to upload a photo, fill the bio, choose privacy settings and optionally location, and interests to illustrate the posts based on the field of interests.
- 4.3 When user doesn't choose any of the privacy options and click on set my profile button, the application ask user to choose one of the privacy options.
Homepage/Timeline
- 4.4 The application shall have search bar, which enables users to find posts by tag, keyword, link and social media platform.
- 4.5 The application shall demonstrate recommended posts in the homepage based on entered interests in the profile and search history.
- 4.6 The application shall have save, share, tagging, like, comment and report features for each post.
- 4.7 Users shall be able to follow each other mutually.
- 4.8 The application shall illustrate the posts from newest to oldest.
- 4.9 User shall be able to reach settings, profile, favourite posts and spaces from homepage.
- 4.10 User shall post a new post from homepage menu.
- 4.11 The application shall offer more recommended posts via show more recommendation button in the homepage.

(v) Spaces

- 5.1 The application shall let user to create spaces to start a discussion.
- 5.2 User shall add name, tag and description to the space after creating it.
- 5.3 The application shall give user option to make spaces visible to public or followers.
- 5.4 Where user started a discussion in spaces, the other users shall share posts and comment each other's.

3. Software and Mockups Designs – UML diagrams and Images

A mockup is a static visual representation of a user interface design. In web development, mockups are often used to demonstrate the layout and design of a website or web application, and can be created using tools such as wireframing or prototyping software.

This is the mock-up design for the dedicated app:

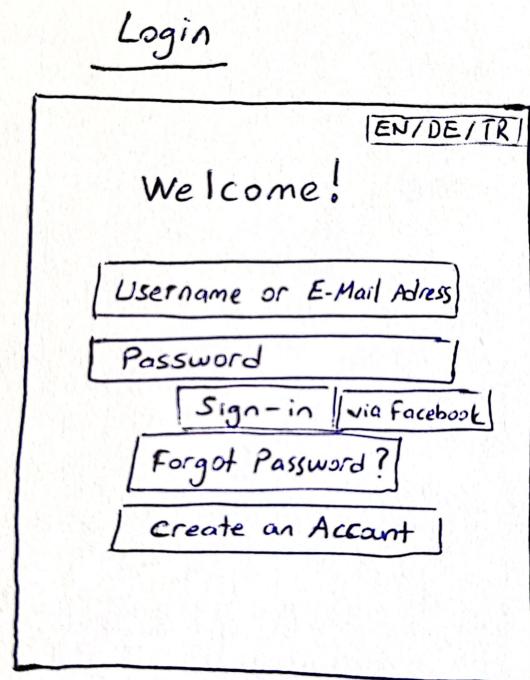


Figure 3.1. Login

Below please find the uml diagram of the general process in a summary.

Create an Account

Username

E-Mail Address

Password

Verify Password

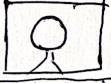
Date of Birth

Register

Figure 3.2. Create an Account

Create a Profile

(pop-ups after creating an account)

Welcome User !

Privacy Options: Private Followers Public

Upload a photo

Fill your bio!

Choose your location (Optional)

What are your interests?
(choose max 5)

Sport	Finance	Politics	Art	Books
Travelling	History	Movies	Theatre	
Music	Football	Basketball	Crypto	

Set my Profile ✓

Figure 3.3. Create a Profile

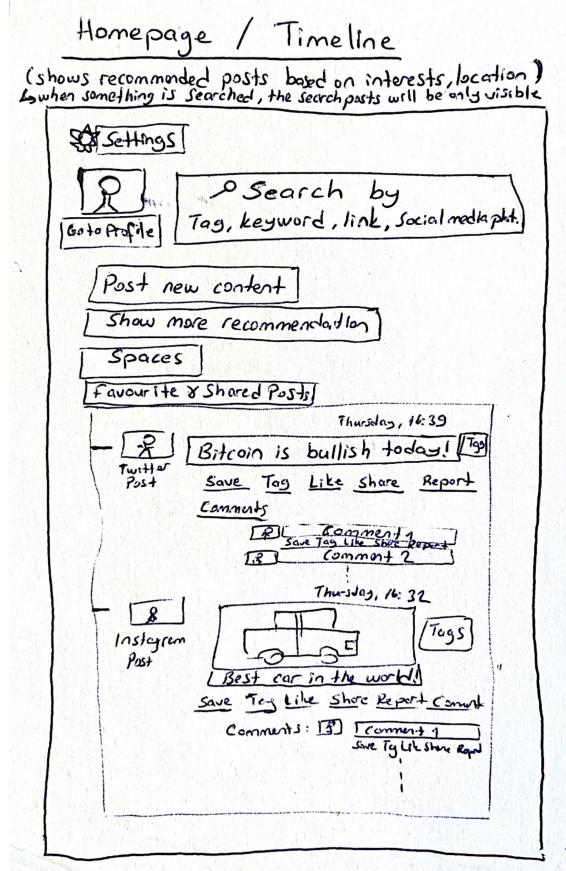


Figure 3.4. Homepage

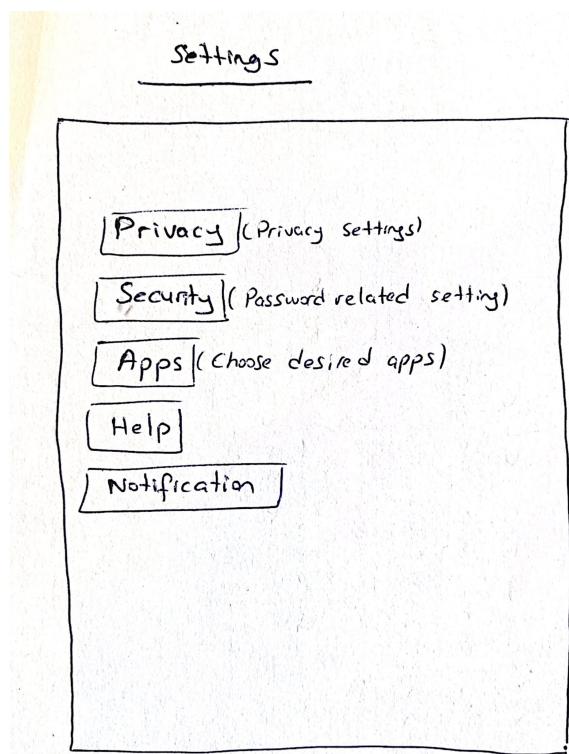


Figure 3.5. Settings

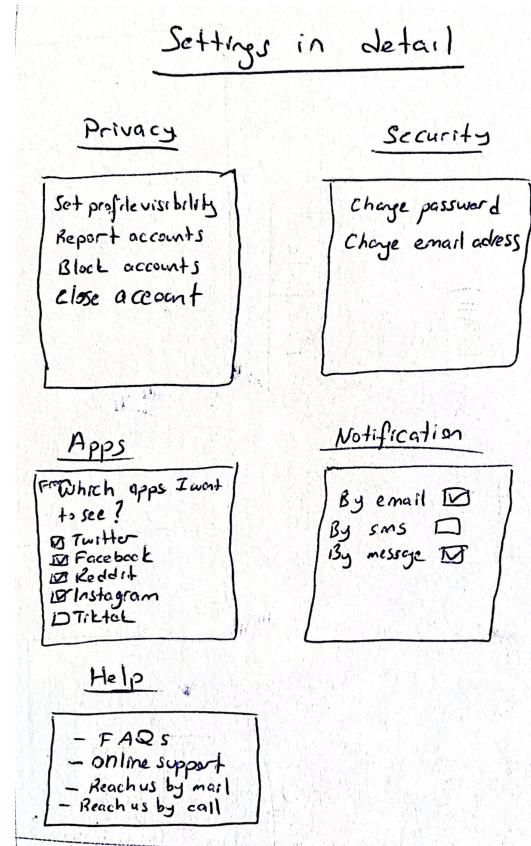


Figure 3.6. Settings in detail

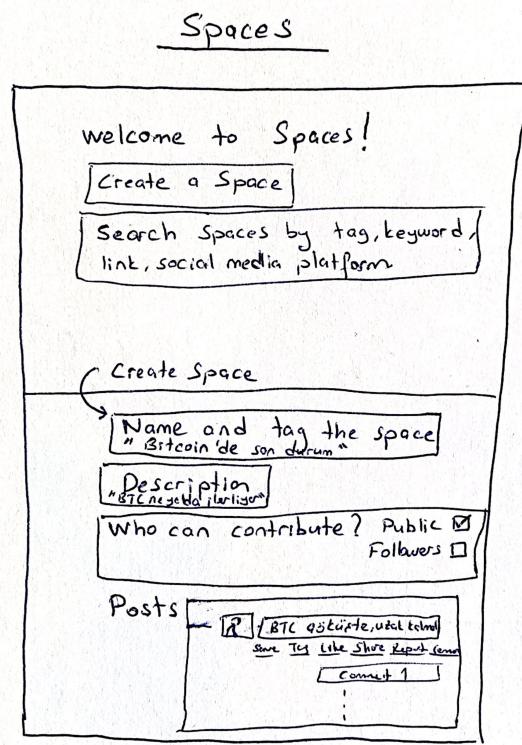


Figure 3.7. Spaces

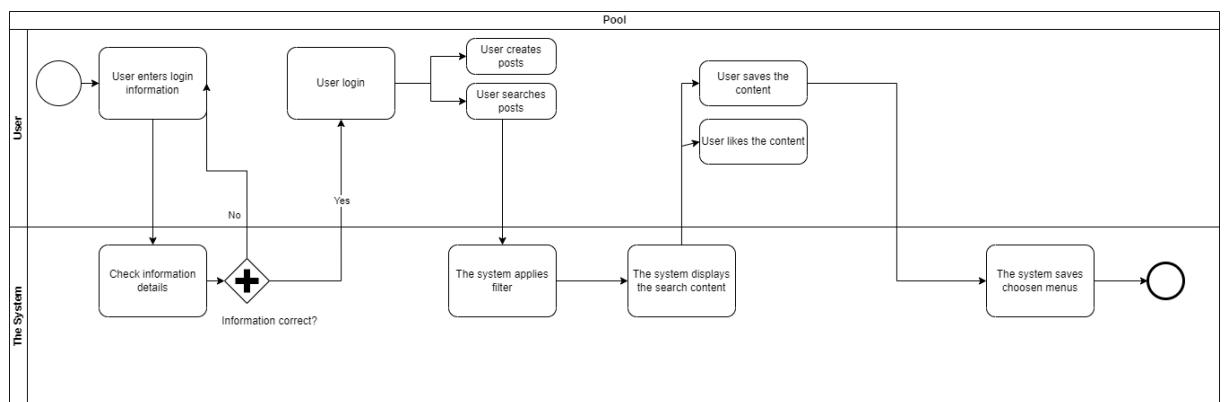


Figure 3.8. Uml diagram

4. Status of the project

The web application via login and fundamental features have been created by using Django within Visual Studio Code, which has been used for development and compile.

This application containing fundamental features works without any problem. People can register and enter the application with their credentials. They can also update their profile (profile photo, location, bio etc.), see others' profile and even follow them. Users can also follow each other, see their posts and even like them. Furthermore, it is also possible to search other users' post, as well. Moreover, with the help of Bootstrap, user interface has a better visual. Below the features of the application can be seen. However, the application has some missing requirements usually related to settings, privacy and spaces.

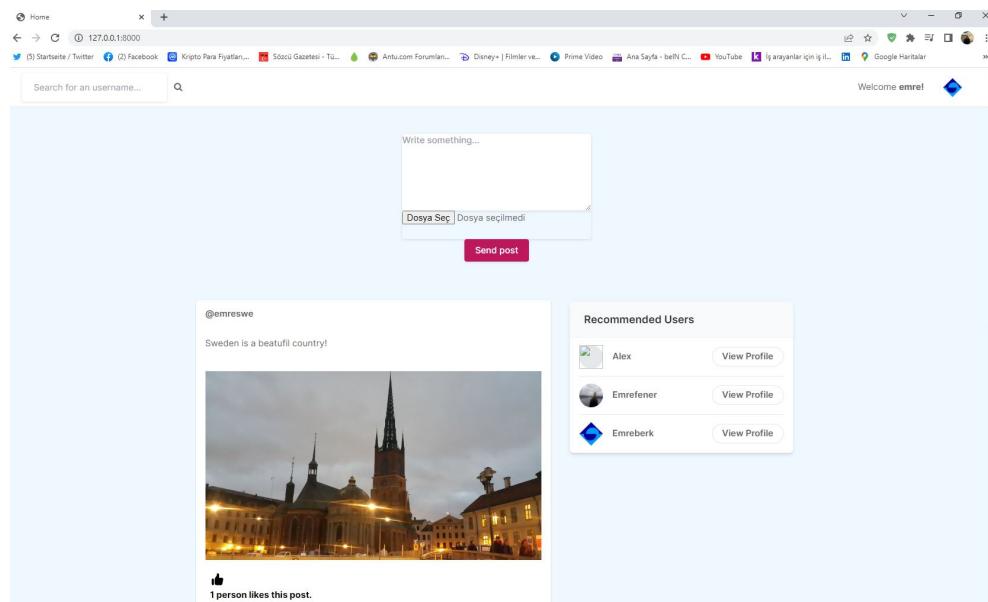


Figure 4.1. Homepage

In addition to the features, dockerization works fine, while deployment ends up with an issue. Please see the next chapter for details.

In summary, the application is partially completed considering all these aspects.

5. Status of Deployment

Dockerization is a need for developers, who need to build, deploy, and manage applications in a variety of environments. In that sense, with the help of docker desktop the application is successfully dockerized. Below you can see the dockerized app:

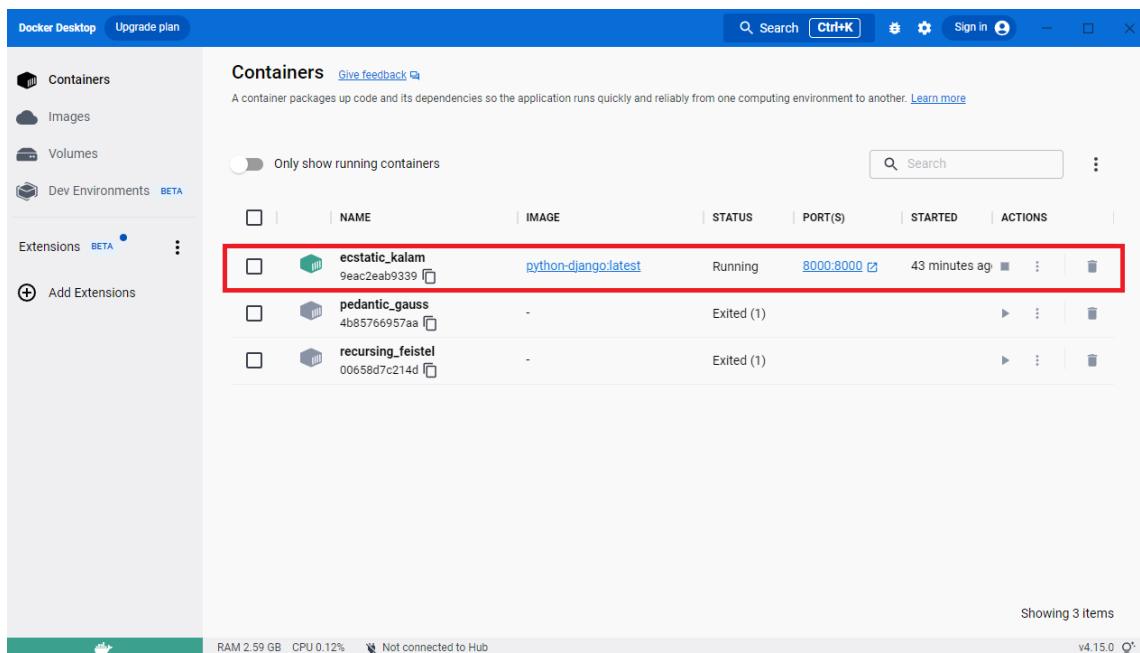


Figure 5.1. Dockerization

On the other hand, deployment has some issues. Even though, the dockerized url is running successfully in AWS, it ends up with the following error when the url is entered.

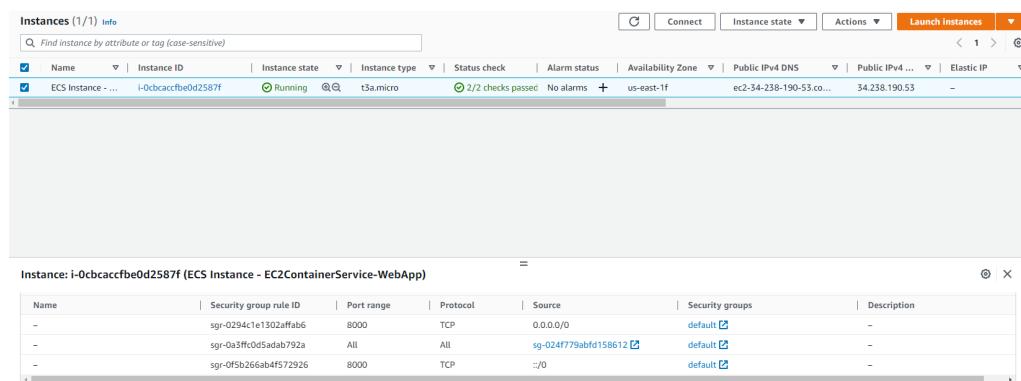


Figure 5.2. Deployment in AWS

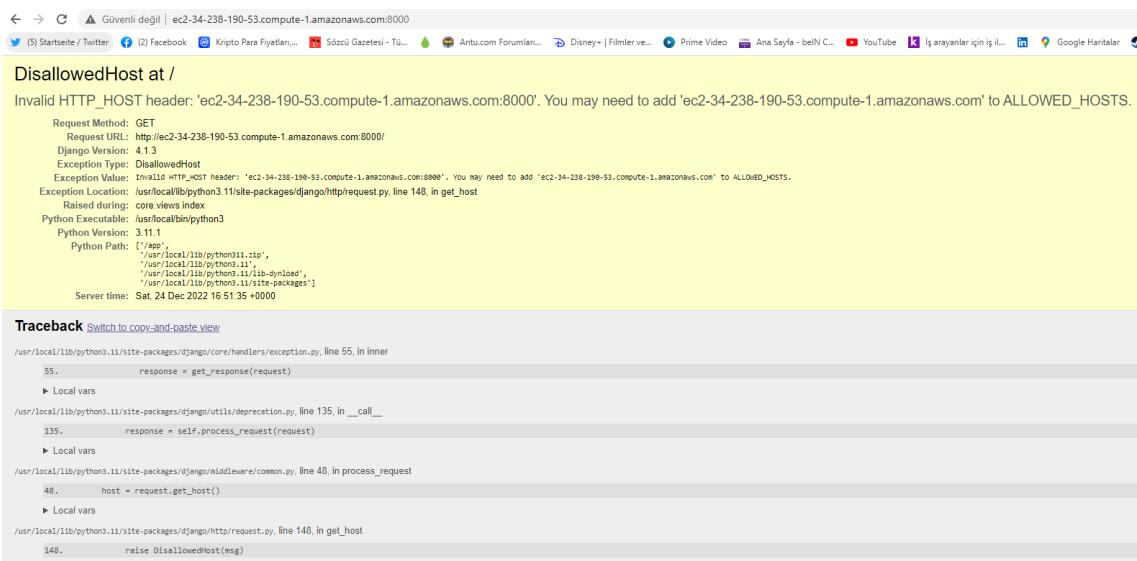


Figure 5.3. Error in deployment

6. System manual

Python and pip must be installed to be able to execute an application created in Django. Moreover, to create a container and image with the dockerized application, docker desktop must be installed with the following requirements:

- Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit: Home or Pro 21H1 (build 19043) or higher, or Enterprise or Education 20H2 (build 19042) or higher.
- Enable the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
- The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
 - 64-bit processor with Second Level Address Translation (SLAT)
 - 4GB system RAM
 - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.
- Download and install the Linux kernel update package.

The other requirements to run this application using Django is as follows:

- asgiref: 3.5.2
- Django: 4.1.3
- Pillow: 9.3.0
- sqlparse: 0.4.3
- tzdata: 2022.6

7. User manual

Please follow these steps to be able to run the code in Github and run the server.

- (i) Download and install the latest versions of python, pip and Django. See chapter 6 for the details.
- (ii) Download and install the latest versions of Docker desktop.
- (iii) Download the code in the Github repository with tag v0.9 given in the first page.
- (iv) Move the downloaded files in a folder
- (v) Extract the venv.rar file to integrate the environmental files with the app. Optionally, this can be created from scratch by user, as well.
- (vi) Launch command prompt and define the location of the folder with all codes using cd command.
- (vii) Execute this command to run the application in localhost. (<http://127.0.0.1:8000/>)

`python manage.py runserver`

- (viii) As an alternative to item vi and vii, it is also possible to run the server with docker. In doing so, run the following commands, respectively.

`docker build -tag python-django .`

`docker run -publish 8000:8000 python-django`

- (ix) Run the container with start button, and type the url: <http://127.0.0.1:8000/>
- (x) Now the application should work fine.

8. Test results

User tests have been performed to test the functionalities such as login, send and search post, following other users' within the application.

(i) Registration tests

1.1 It is not possible to create an account with existing user as expected.

The image shows a registration form titled "Registration". At the top, a red error message reads "This username is already registered!". Below the message are four input fields: "Username", "E-mail Adress", "Password", and "Confirm Password". A large pink "Register" button is at the bottom. Below the button, a link says "Do you already have an account? Login".

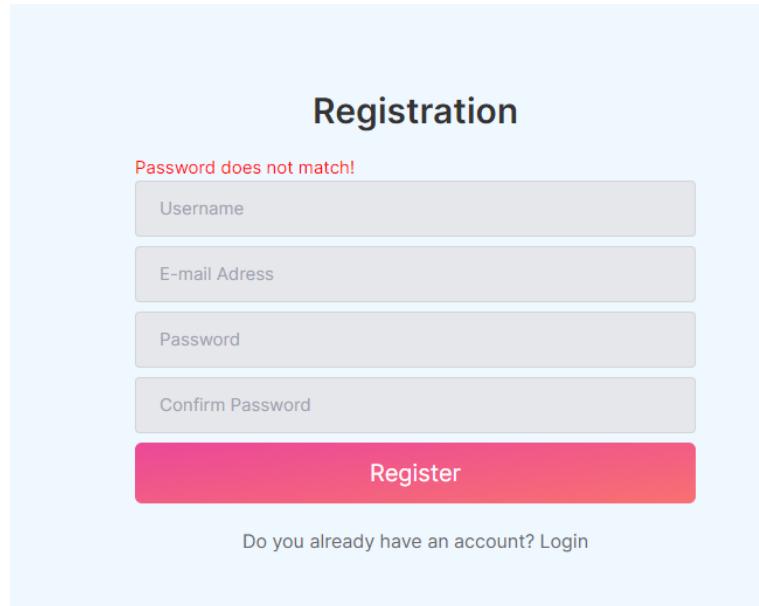
Figure 8.1.

1.2 It is not possible to create an account with registered mail as expected.

The image shows a registration form titled "Registration". At the top, a red error message reads "This e-mail is already registered!". Below the message are four input fields: "Username", "E-mail Adress", "Password", and "Confirm Password". A large pink "Register" button is at the bottom. Below the button, a link says "Do you already have an account? Login".

Figure 8.2.

1.3 Password check has been done during the registration. The application doesn't allow non-matching passwords as expected.

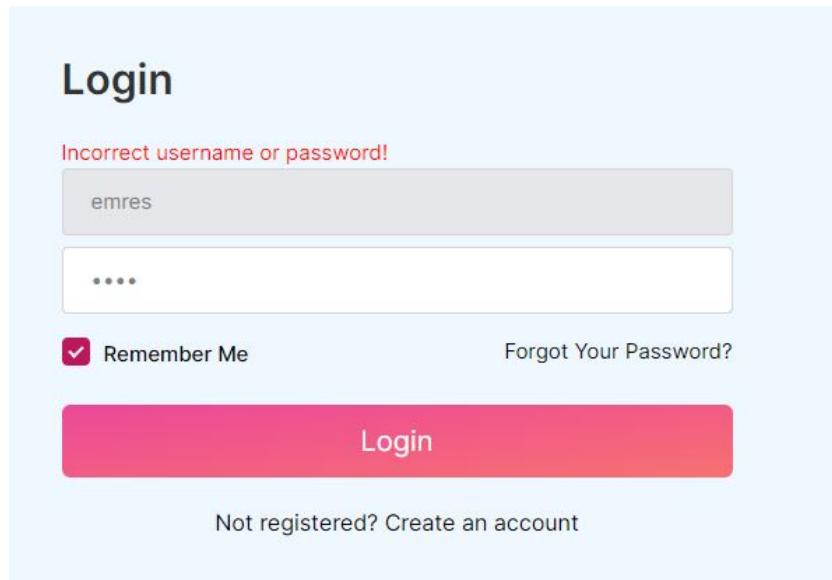


The image shows a registration form titled "Registration". It includes four input fields: "Username", "E-mail Adress", "Password", and "Confirm Password". A red error message "Password does not match!" is displayed above the "Confirm Password" field. Below the input fields is a pink "Register" button. At the bottom of the form, there is a link "Do you already have an account? Login".

Figure 8.3.

(ii) Login test

2.1 In case of wrong username and/or password, it gives error to the user.



The image shows a login form titled "Login". It features two input fields: "Username" containing "emres" and "Password" containing "****". A red error message "Incorrect username or password!" is displayed above the password field. Below the input fields is a pink "Login" button. To the left of the button is a checkbox labeled "Remember Me" with a checked mark. To the right of the button is a link "Forgot Your Password?". At the bottom of the form, there is a link "Not registered? Create an account".

Figure 8.4.

(iii) Testing send and visualize post features as well as liking them

3.1 A post created by user "emreswe", which can also be seen and liked by user "emre".

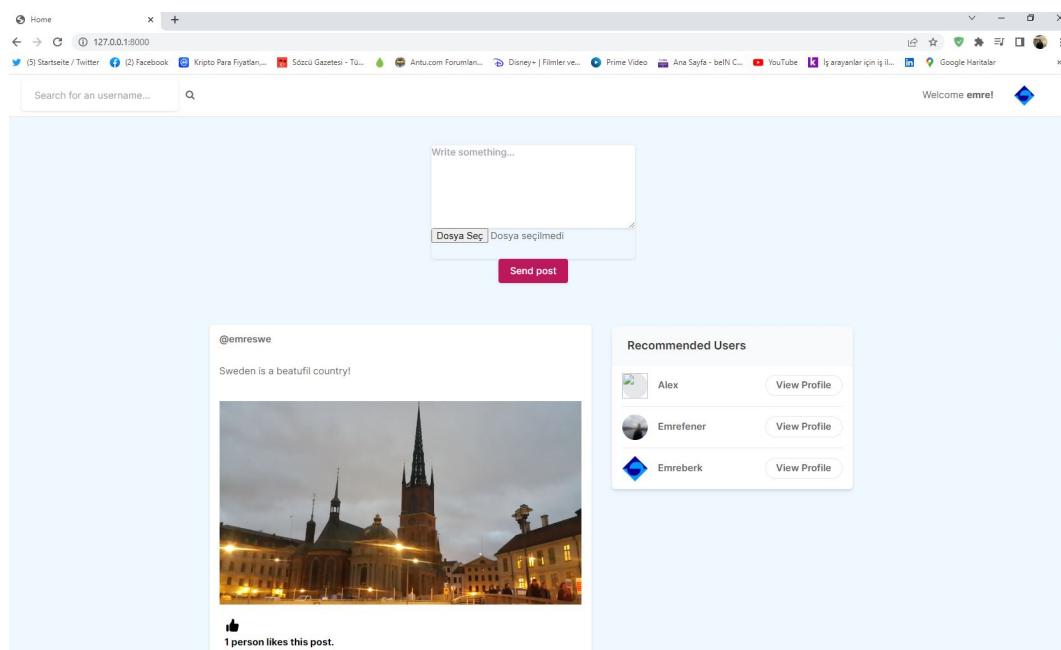


Figure 8.5.

3.1 A post and bio created by the user can be seen in the profile menu. It is also validated to follow other users from this menu.

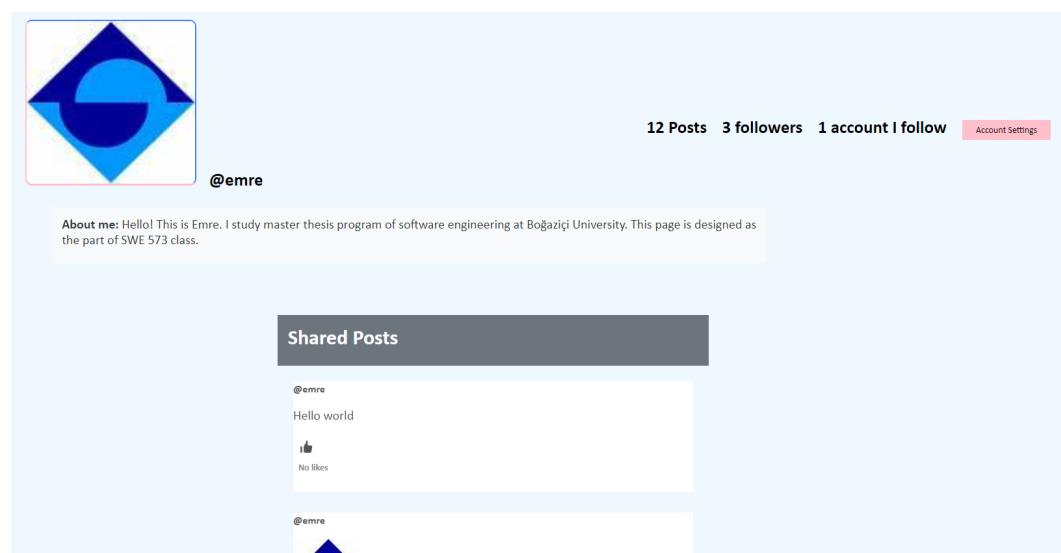


Figure 8.6.

(iv) Search other posts

4.1 A post created by user "emreswe", which can also be seen and liked by user "emre".

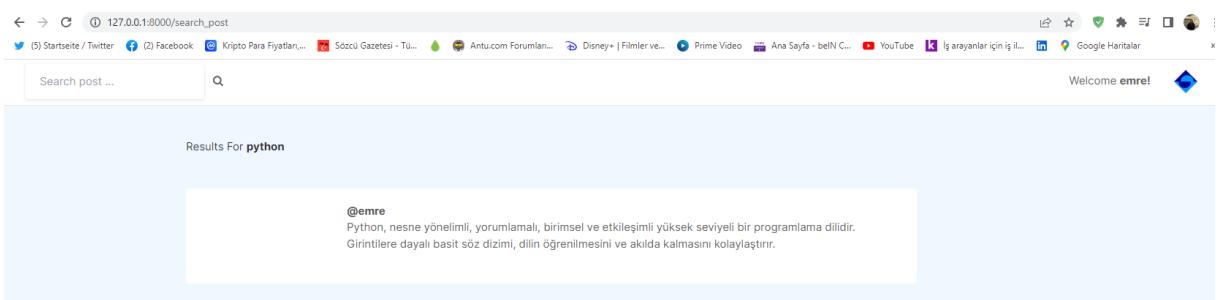


Figure 8.7.

(v) Verifying created posts, users, likes and comments from admin panel

5.1 From the created models in Django, all the users, comments, likes and etc. can be verified. Below, you can find the list of users.

Select profile to change	
Action:	<input type="button" value="Go"/> 0 of 5 selected
<input type="checkbox"/> PROFILE	
<input type="checkbox"/> emrefener	
<input type="checkbox"/> emreswe	
<input type="checkbox"/> emre	
<input type="checkbox"/> emreberk	
<input type="checkbox"/> alex	
5 profiles	

Figure 8.8.

REFERENCES

- (i) <https://www.wikipedia.org/>
- (ii) <https://www.youtube.com/watch?v=W5Ov0H7Eo4> (*For dockerization*)
- (iii) <https://ordinarycoders.com/django-custom-user-profile>
- (iv) https://www.w3schools.com/django/django_add_record.php
- (v) <https://getbootstrap.com/docs/4.0/components/forms/>
- (vi) <https://pylessons.com/django-user-profile>
- (vii) <https://ngangasn.com/how-to-use-bootstrap-5-with-django-the-right-way/>
- (viii) <https://www.youtube.com/watch?v=rHux0gMZ3Eg> (Django Tutorial 1)
- (ix) <https://www.youtube.com/watch?v=F5mRW0jo-U4> (Django Tutorial 2)
- (x) <https://www.youtube.com/watch?v=0mCZdemSsbs> (Bootstrap tutorial)
- (xi) <https://www.youtube.com/watch?v=L5My8Tk-zs> (AWS tutorial 1)
- (xii) <https://www.youtube.com/watch?v=zs3tyVgiBQQ> (AWS tutorial 2)