# MemoryBox (Social Media Information Saving & Sharing Platform)

Author: Şehmuz Cacına

Course: SWE573 – Software Development Practice

Date: 03.01.2023

Project Name: MemoryBox (*Social Media Information Saving & Sharing Platform*)

Git Repository: <https://github.com/sehmuzc/SoftwareDevelopmentPractice>

Git Tag Version: 0.9 <https://github.com/sehmuzc/SoftwareDevelopmentPractice/releases/tag/0.9>

Deployment URL: [http://54.89.208.162:8000](http://54.89.208.162:8000/)

(You can register from the top right link on the home page and start testing the deployed system with your username and password, there is no need for any special test user information.)

**HONOR CODE** Related to the submission of all the project deliverables for the Swe573 2022 Fall semester project reported in this report, I 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.

Şehmuz Cacına

Signature:

Table of Contents

[MemoryBox (Social Media Information Saving & Sharing Platform) 1](#_Toc123589292)

[Overview 2](#_Toc123589293)

[Functional Requirements 3](#_Toc123589294)

[Home Page 3](#_Toc123589295)

[Register 4](#_Toc123589296)

[Login / Logout 4](#_Toc123589297)

[Friends 4](#_Toc123589298)

[Spaces 5](#_Toc123589299)

[Profile 5](#_Toc123589300)

[Blogpost Entry, Update, Delete 5](#_Toc123589301)

[Non-Functional Requirements 5](#_Toc123589302)

[Usability & Performance 5](#_Toc123589303)

[Security 5](#_Toc123589304)

[Project Status - Requirements 6](#_Toc123589305)

[Project Status - Deployment 7](#_Toc123589306)

[Design (Use Case Diagram) 8](#_Toc123589307)

[Design (UML Class Diagram) 9](#_Toc123589308)

[Design (UML Sequence Diagram) 10](#_Toc123589309)

[Design (Mockups) 10](#_Toc123589310)

[User Manual 12](#_Toc123589311)

[System Manual 13](#_Toc123589312)

[AWS Deployment 15](#_Toc123589313)

[Test Results (Unit & User) 16](#_Toc123589314)

[User Tests 16](#_Toc123589315)

[References 18](#_Toc123589316)

[Additional Information 19](#_Toc123589317)

# Overview

MemoryBox is a social media application that has been prepared so that the users of this application will be able to save their social media content and share with other users of the system. Users will be able to easily save the links and their information that they see in other social media platforms and they will be able to share information with other users of the application. The functional requirements and the non-functional requirements of the system are detailed in the upcoming section.

# Functional Requirements

The social media application will provide the users the ability to save social media content (blogpost) that they are interested in. A blogpost is content that the users will be recording in the application and it will be consisting of below information.

* Title
* Content
* Author
* Date and Time
* Link (URL of the post from various social media applications)
* Tags

The users will also be able to view the blogposts of other users of the application. The users will also have the ability to view the profiles of other users and the blogposts of those users. The users will also have the ability to search different topics so that they can find related blogposts. The users will also be able to create spaces, which will give the users ability to share blogposts on a specific topic such as “Bosphorus University”. The application will be consisting of the below steps:

1. Home Page
2. Registration Page
3. Login Page
4. Profile (View and Update)
5. Create Blogpost
6. Search Blogposts
7. Create and View Spaces
8. Friends Page

## Home Page

1. Users shall be able to view the links to the main sections of the application when they open the home page of the application. a. Home b. Register c. Login / Logout d. Friends e. Spaces f. Profile g. Blogpost Entry
2. If the user has not completed the login process then he can only see the contents that will be visible to non-logged in users which are homepage, login page and register page.
3. If the user completes the login process successfully then he shall be able to view the main sections of the application below: a. Home b. Logout c. Friends d. Spaces e. Profile f. Blogpost Entry
4. Register and Login links on the homepage shall not be visible to logged in users.
5. Users shall be able to view the blogposts that they previously saved.
6. Users shall be able to view the blogposts of their friends.
7. Users shall be able to delete their own blogspots if they want not to see the posts anymore.

## Register

1. When a user clicks on the register link the “Register Page” shall be opened.
2. User shall only be able to register after successfully completing the below information on the Register Page: o Username o E-Mail Address o First Name o Last Name o Password
3. A user shall not be able to complete the registration process with a username that already exists in the system.
4. A user shall enter password information twice so that he does not mistakenly enter a password and complete the registration with a wrong password.

## Login / Logout

1. When a non-logged in user clicks on the Login link the “Login Page” shall be opened.
2. A user shall be able to login to the website by using the correct username and password combination that he created at the registration process.
3. A user shall not be able to login if the password does not match the password of the username he provided.
4. If the user completes the login process successfully then he shall be able to view the main sections of the application below: o Home o Logout o Friends o Spaces o Profile o Blogpost Entry
5. Logout link shall only be visible for logged in users.
6. Whenever a user clicks on the logout link the logout page shall be opened and the user shall see the information that he is logged out.
7. After the user has logged out of the system he shall not be able to view the profile page and the blogposts and he shall not be able create a blogpost entry.

## Friends

1. A user shall be able to get friends with the other users of the application.
2. A user shall be able to view all his friends in the friends section of the webpage.
3. When a user becomes a friend with an other user, he shall be able to view the blogposts of that friend user.
4. A user shall be able to view the profile page of his friends.

## Spaces

1. The users shall have the ability to search different topics so that they can find related blogposts.
2. The users will also be able to create spaces, which will give the users ability to share blogposts on a specific topic such as “Bosphorus University”.
3. The users shall be able to view all the blogposts related to a space in this section.

## Profile

1. The users shall have the ability to view their profile from the My Profile section of the application.
2. The users should have the ability to add a profile picture.
3. The users shall have the ability to view their profile information such as username and e-mail address, first and last name.
4. The users shall have the ability to update their name and e-mail address information but they should not have the ability to update their username from this section.

## Blogpost Entry, Update, Delete

1. The users shall have the ability to create blogposts with entering the title, content (note about the blogpost) and URL of the related web page.
2. Title, content and the URL fields shall be mandatory for blogpost entry.
3. The date, time and author of the blogpost should be recorded in the system.
4. The users shall be able to update the title, content and URL of a blogspot.
5. The users shall be able to delete their own blogposts.

# Non-Functional Requirements

## Usability & Performance

1. All pages on the website shall be loaded within 5 seconds for at least %99 of the users.
2. The website shall have the capacity for at least 5.000 logged-in users at any given time.
3. The website must have at least 95% uptime.
4. The application must be scalable enough to support 100.000 visits at the same time while maintaining its optimum performance.

## Security

1. The password information shall never be visible at any page of the website.
2. The username and password records shall not be viewed or updated by unauthorised users.

# Project Status - Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Number** | **Requirement** | **Status** | **Detail** |
| Home-1 | Users shall be able to view the appropriate links to the main sections of the application when they open the home page of the application. a. Home b. Register c. Login / Logout d. Friends e. Spaces f. Profile g. Blogpost Entry | ✓ |  |
| Home-2 | If the user has not completed the login process then he can only see the contents that will be visible to non-logged in users which are homepage, login page and register page. | ✓ |  |
| Home-3 | If the user completes the login process successfully then he shall be able to view the main sections of the application below: a. Home b. Logout c. Friends d. Spaces e. Profile f. Blogpost Entry | ✓ | Post Space management has been done using tagged posts structure. |
| Home-4 | Register and Login links on the homepage shall not be visible to logged in users. | ✓ |  |
| Home-5 | Users shall be able to view the blogposts that they previously saved. | ✓ |  |
| Home-6 | Users shall be able to view the blogposts of their friends. | ✓ |  |
| Home-7 | Users shall be able to delete their own blogspots if they want not to see the posts anymore. | ✓ |  |
| Register-1 | When a user clicks on the register link the “Register Page” shall be opened. | ✓ |  |
| Register-2 | User shall only be able to register after successfully completing the below information on the Register Page: o Username o E-Mail Address o First Name o Last Name o Password | ✓ |  |
| Register-3 | A user shall not be able to complete the registration process with a username that already exists in the system. | ✓ |  |
| Register-4 | A user shall enter password information twice so that he does not mistakenly enter a password and complete the registration with a wrong password. | ✓ |  |
| Login / Logout-1 | When a non-logged in user clicks on the Login link the “Login Page” shall be opened. | ✓ |  |
| Login / Logout-2 | A user shall be able to login to the website by using the correct username and password combination that he created at the registration process. | ✓ |  |
| Login / Logout-3 | A user shall not be able to login if the password does not match the password of the username he provided. | ✓ |  |
| Login / Logout-4 | If the user completes the login process successfully then he shall be able to view the main sections of the application below: o Home o Logout o Friends o Spaces o Profile o Blogpost Entry | ✓ |  |
| Login / Logout-5 | Logout link shall only be visible for logged in users. | ✓ |  |
| Login / Logout-6 | Whenever a user clicks on the logout link the logout page shall be opened and the user shall see the information that he is logged out. | ✓ |  |
| Login / Logout-7 | After the user has logged out of the system he shall not be able to view the profile page and the blogposts and he shall not be able create a blogpost entry. | ✓ | The non-logged in users have been given the option to view posts of other users in the home page to better grasp the application. |
| Friends-1 | A user shall be able to get friends with the other users of the application. | ✓ |  |
| Friends-2 | A user shall be able to view all his friends in the friends section of the webpage. | ✓ | All the users are listed in Users section and they can be followed/nonfollowed. |
| Friends-3 | When a user becomes a friend with an other user, he shall be able to view the blogposts of that friend user. | ✓ |  |
| Friends-4 | A user shall be able to view the profile page of his friends. | ✓ |  |
| Spaces-1 | The users shall have the ability to search different topics so that they can find related blogposts. | ✓ |  |
| Spaces-2 | The users will also be able to create spaces, which will give the users ability to share blogposts on a specific topic such as “Bosphorus University”. | ✓ | Users can tag their blogposts with related information to achieve this functionality. |
| Spaces-3 | The users shall be able to view all the blogposts related to a space in this section. | ✓ | Users can search using tags filter. |
| Profile-1 | The users shall have the ability to view their profile from the My Profile section of the application. | ✓ |  |
| Profile-2 | The users should have the ability to add a profile picture. | ✓ |  |
| Profile-3 | The users shall have the ability to view their profile information such as username and e-mail address, first and last name. | ✓ |  |
| Profile-4 | The users shall have the ability to update their name and e-mail address information but they should not have the ability to update their username from this section. | ✓ |  |
| Blogpost-1 | The users shall have the ability to create blogposts with entering the title, content (note about the blogpost) and URL of the related web page. | ✓ | Tag feature has also been added. |
| Blogpost-2 | Title, content, and the URL fields shall be mandatory for blogpost entry. | ✓ |  |
| Blogpost-3 | The date, time and author of the blogpost should be recorded in the system. | ✓ |  |
| Blogpost-4 | The users shall be able to update the title, content and URL of a blogspot. | ✓ |  |
| Blogpost-5 | The users shall be able to delete their own blogposts. | ✓ | The users can also update their blogposts. |
| Usability Perform.-1 | All pages on the website shall be loaded within 5 seconds for at least %99 of the users. | ✓ |  |
| Usability Perform.-2 | The website shall have the capacity for at least 5.000 logged-in users at any given time. | ✓ |  |
| Usability Perform.-3 | The website must have at least 95% uptime. | ✓ |  |
| Usability Perform.-4 | The application must be scalable enough to support 100.000 visits at the same time while maintaining its optimum performance. | ✓ |  |
| Security-1 | The password information shall never be visible at any page of the website. | ✓ |  |
| Security-2 | The username and password records shall not be viewed or updated by unauthorised users. | ✓ |  |

# Project Status - Deployment

All the requirements have been successfully completed and the application has been dockerised and deployed. The deployment URL is as follows:

Deployment URL: [http://54.89.208.162:8000](http://54.89.208.162:8000/)

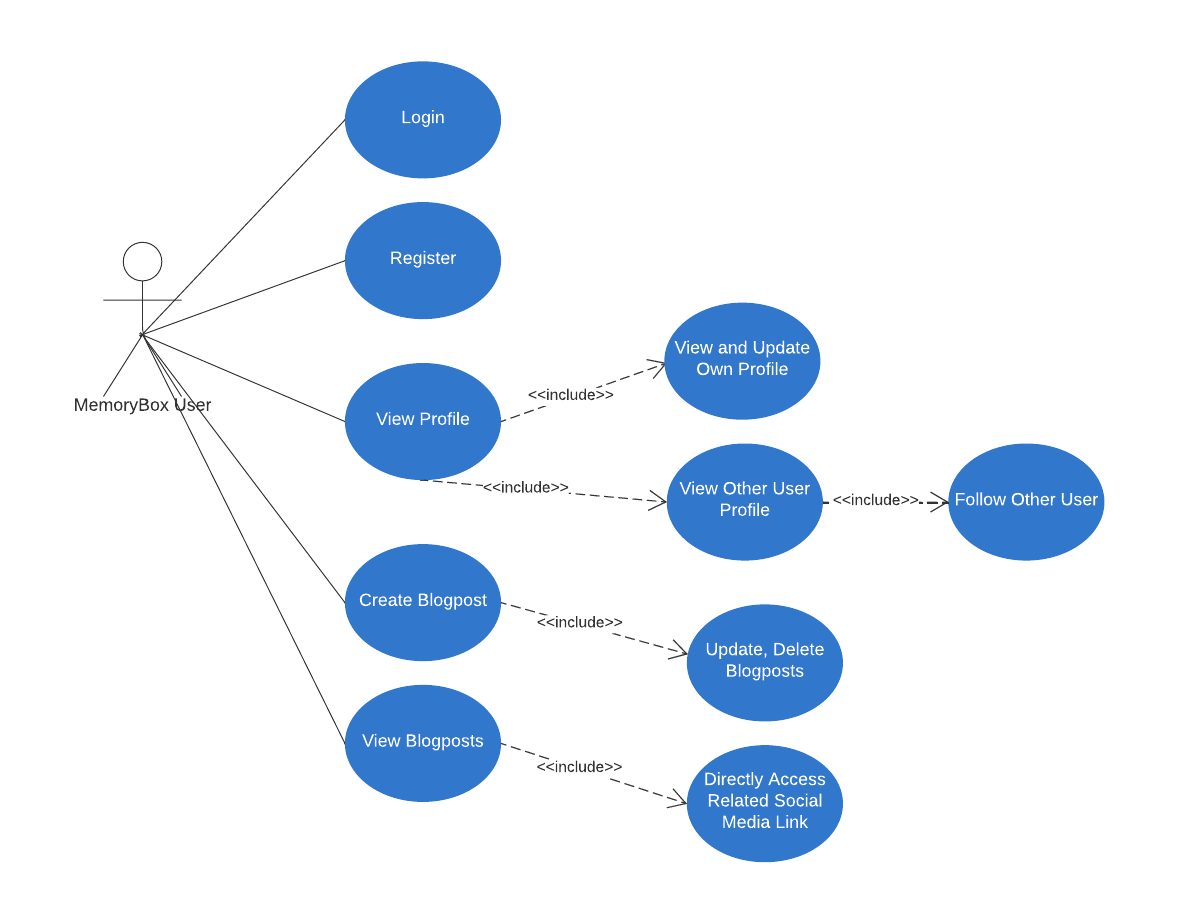
All the functionality detailed in the requirements section have been fully applied at the deployed application.

In addition to the functional and non-functional requirements below issues have also been implemented or applied.

* The blogposts have been grouped to following sections:
  + All Posts
  + My Posts: This section shows the previous posts of the logged in user.
  + Friend Posts : This section shows the posts of other users that the logged in user has followed before.
* The links of the blogposts have been represented with “Go To Link” green button to be easily reachable.
* The tags can be added by the creators of the blogposts so that other users may easily filter related content. For example a user can create a new space of “Bosphorus University” by adding the tag “Bosphorus University” to his post. If this space has been also created before, than the related post will be able to added to that space, users who filter “Bosphorus University” tags will see all the content related to this space.
* There has been a shortcut from the posts to the profile page so that you may decide to follow an other user. This will help users to follow other users if they like the shared posts of that user.
* “Posts I Saved” function has been added so that users can save the posts of other users whether they follow that user or not. User can later decide to remove the post from his saved posts section. This feature has been implemented to let users bookmark special posts and later delete these posts when their work with that posts have been done.
* “Comments” functionality has been added so that users can interact with each other by commenting on the blogposts that they are interested in.
* The web application is fully mobile compatible, the tests have been done and related bugs have been solved to achieve full mobile responsiveness.

# Design (Use Case Diagram)

Below is the use case diagram for the MemoryBox application.



# Design (UML Class Diagram)

Diagram

Description automatically generated

# Design (UML Sequence Diagram)

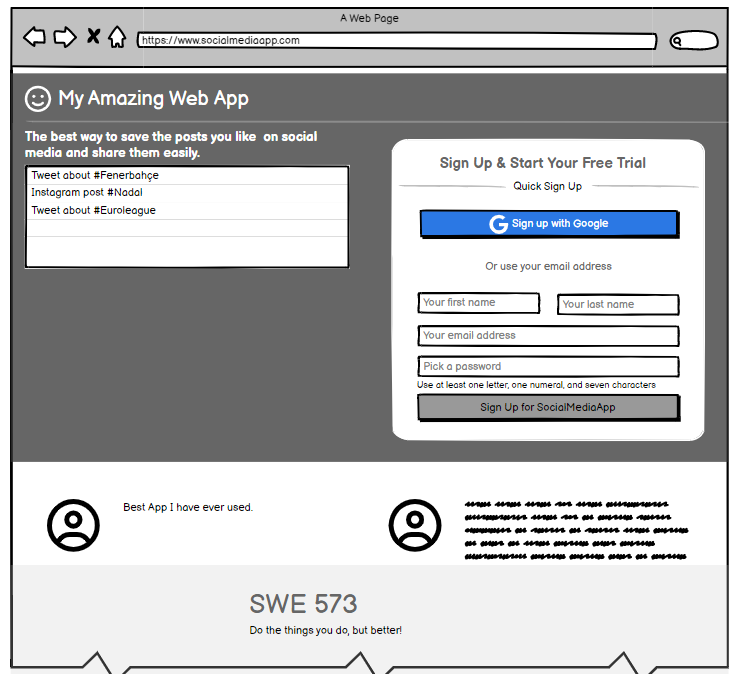
Chart, waterfall chart

Description automatically generated

# Design (Mockups)

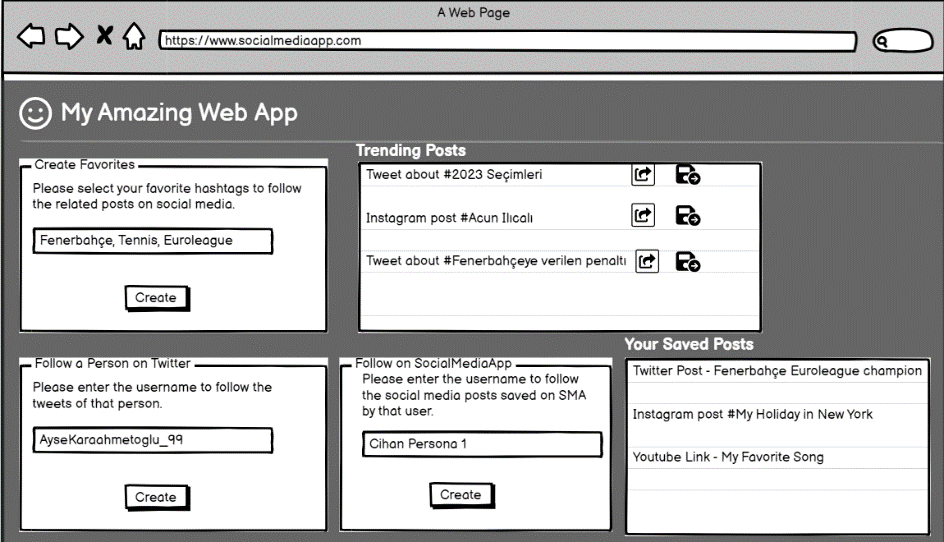
Below are the mockups that were prepared at the beginning of the project for the social media application.

* When the user first enters the web application, he will be directed to the authentication part.
* The user may sign-up using his own e-mail address by entering name and creating a password to use in the application.
* A new user will be viewing the most popular posts from different social media applications on the left of the webpage.



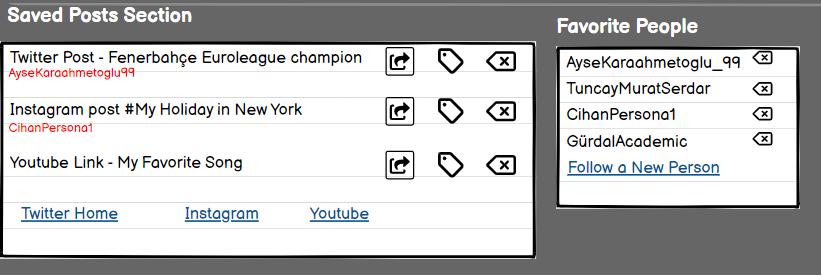
* When an authenticated user (logged in using his login credentials) comes to the webpage, he will be able to view and use the different sections below.

1. The user will be able to see the popular posts which will show the most recent posts of different users.
2. The user will be able to follow an other user so that he can later follow the related posts of that user.
3. The user will be able to view the posts he previously saved into the application in a section.



* When the user selects at least one favorite person, then he will be able to view the list of his favorite users. He will later able to delete a favorite user and no more follow the posts of that user.
* The user should be able to do the following actions on his saved posts.

1. Share: The user should be able to share a post he previously saved.
2. Tag: The user should be able to give a name to the posts he saved, so that he can later filter them according to the tag.
3. Delete: The user should be able to delete a saved post.
4. Information: The user should be able to see the content of the post and the username who posted the related content.



# User Manual

This user manual will provide an overview of the features and functionality of the MemoryBox application.

**Home page**: The home page is the default page when you open the application using the deployed application link. Home page includes the blogposts of all users and links to the main sections of the application such as Register, Login, Logout, Profile, My Posts, Friend Posts, Users, Post Entry and Search. If the user is not logged in, he will only be able to access the posts on home page, register page, and login page. If the user is logged in, he will also be able to access the My Posts, Friend Posts, Users, Post Entry and Search functionalities.

**Register page**: To create an account on the MemoryBox application, click on the "Register" link on the top right part of the home page. You will be directed to the register page, where you will need to enter your username, email address, first name, last name, password and password confirmation information. All the fields are mandatory and after entering all the information you can easily register by clicking the "Sign Up" button to complete the process.

**Login page**: To log in to the MemoryBox application, click on the "Login" link on the home page and enter your username and password. If you do not have a username then you may click the “Register” link as a shortcut that appears next to the “Need An Account” text below the “Login” button. After successfully providing the username and password credentials you will be logged in and will be able to view the “Friend Posts” section of the application. This redirection has been done in purpose for the user to be able to view related content from related users and not all the posts of all other users.

**Logout link:** You may use this link when you want to log-out of the application, after clicking the link you will see the “You have logged out” message.

**Profile page**: To view and update your personal information, click on the "My Profile" link on the home page, this page is only visible for logged in users. The profile page includes your username, first name, last name, e-mail address, last login time and profile picture. You can update your e-mail address, first name, last name and profile picture information by changing the information you need to change and clicking “Update Profile” button on the bottom of the page.

**Create blogpost:** To create a new blogpost, click on the "Post Entry" link on the home page. You will be directed to a form page where you can enter Title, Content, Link and Tags information and create a post by clicking the “Post” button. Tag information is not mandatory, however if you need to enter more than one tag, then you need to separate the tags by using comma as the delimiter.

**Search blogposts**: To search for blogposts on specific topics, you can use the search bar on the home page. Enter the keyword you want to search for and click the "search" button to view related posts about your keyword. The search has been done on both title and content information of the blogposts.

**Tag filtering:** You can create a blogpost with the tags you want the post to be associated with. You can view the tags of the blogposts with the blue colored tags next to the Link information of a post. If you want to filter all the related posts about that tag category you can do this by just clicking on the tag. For example, you can see all the “Messi” tagged items by clicking the “Messi” link on a post that has the tag “Messi”.

**Users page**: To view other users and to manage the following/unfollowing tasks about that users, you may click on the "Users" link on the home page. You will be able to view a list of users and their username, first and last names. When you click on the Detail button of a user, you may follow or unfollow that user depending on your connection and see the e-mail and last active date of that user in addition to the username and name information.

**Friend Posts page:** To view the posts of the users that you followed earlier you can access the “Friend Posts” link from the navigation bar. The users are also redirected to this page after their login is completed successfully. If you unfollow an other user you will not be able to view the posts of that user in this page.

**My Posts page:** To view your own posts that you created earlier you can use this page by clicking the “My Posts” link on the navigation bar or “My Own Posts” link on the right part of the web page. You can also update or delete your own posts by clicking on a post title where you will be redirected to the post detail page.

**Comments:** You can add comments on the blogposts just by using the “Add Comment” button on the detail page of a blogposts. The detail pages are always accessible just by clicking on the title of a post.

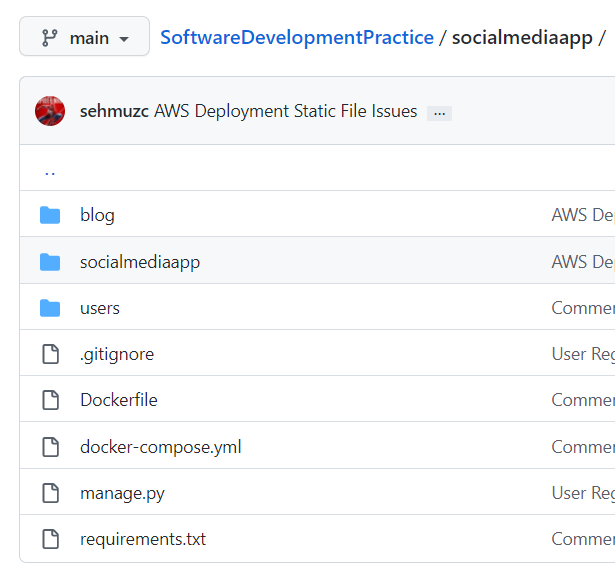
# System Manual

This system manual will provide the necessary instructions on how to dockerise the application in your local environment.

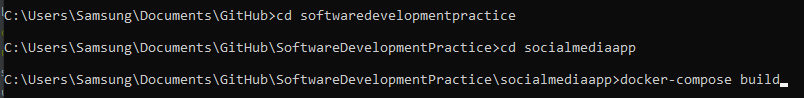
1. In order to build the application you first need to install Docker on your machine. You can use the link below to install Docker Desktop, it has been mentioned that it is the easisest and recommended way of installing Docker Compose. Docker Desktop also includes Docker Engine and CLI which are the prerequisites of Compose.

Docker Compose Link: <https://docs.docker.com/compose/install/>

1. To get started, you need to copy the folders of the application to your local environment using the files in the USB shared with the project report. Root directory of the project should be like the below screenshot.

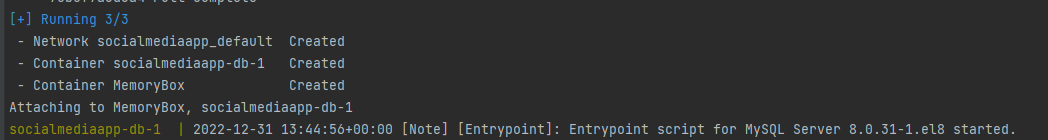


1. Navigate to the root directory using command line and run the command “docker-compose build”. This will help you build the necessary Docker images for the application. All the requirements have been detailed in requirements.txt file so you will not need to do anything special to install the related requirements except dockerising the application.

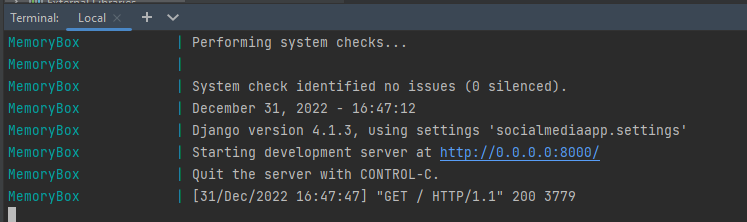


1. Run the application using the command “docker-compose up”. The containers will be up and running after the process is completed and the web application will be ready afterwards.

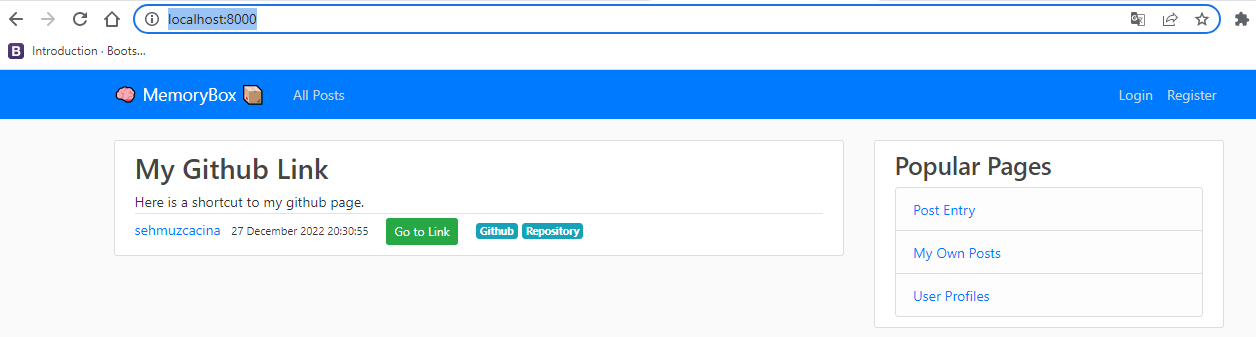




1. When you see the below screenshot in your terminal, you will be able to access the application using the “localhost:8000” link and you can access this link also by clicking the link next to the web container in Docker Desktop.



1. When you open the “localhost:8000” address from your browser you will be able to view the application.



1. All the screenshots have been taken at 31.12.2022 at local environment to verify that all the docker-compose build and up functions work as expected.

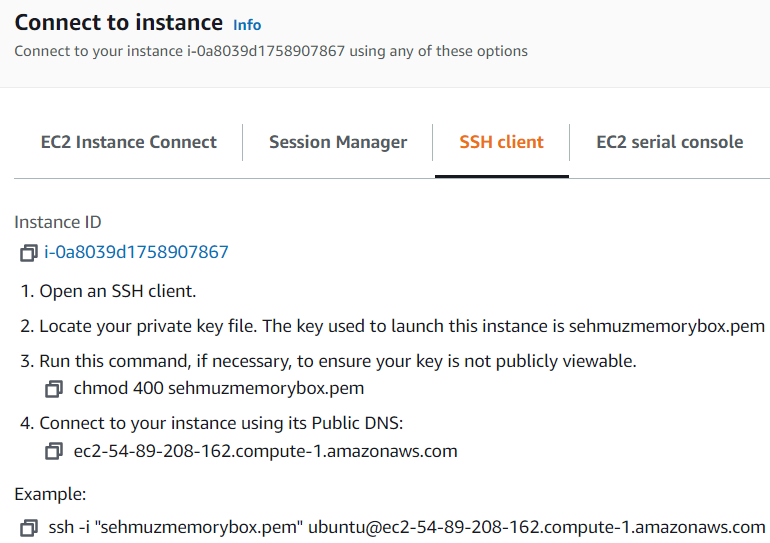
## AWS Deployment

The project has been deployed at the below URL on AWS using the dockerisation process described above on an AWS EC2 instance.

**Deployment URL:** [**http://54.89.208.162:8000**](http://54.89.208.162:8000/)

In the process of deploying the project to the AWS, I followed the below steps.

1. Open an account at AWS and create an EC2 instance.
2. Install docker on the instance using the commands below. I used Putty program to connect to the instance by SSH, the instructions are given at the connect to instance part on AWS.



sudo apt-get update

sudo apt-get install \ apt-transport-https \ ca-certificates \ curl \ gnupg-agent \ software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

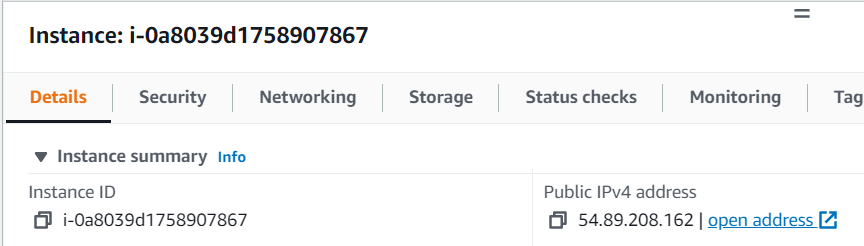
sudo apt-get install docker-ce docker-ce-cli containerd.io

apt-cache madison docker-ce

sudo apt-get install docker-ce docker-ce-cli containerd.io

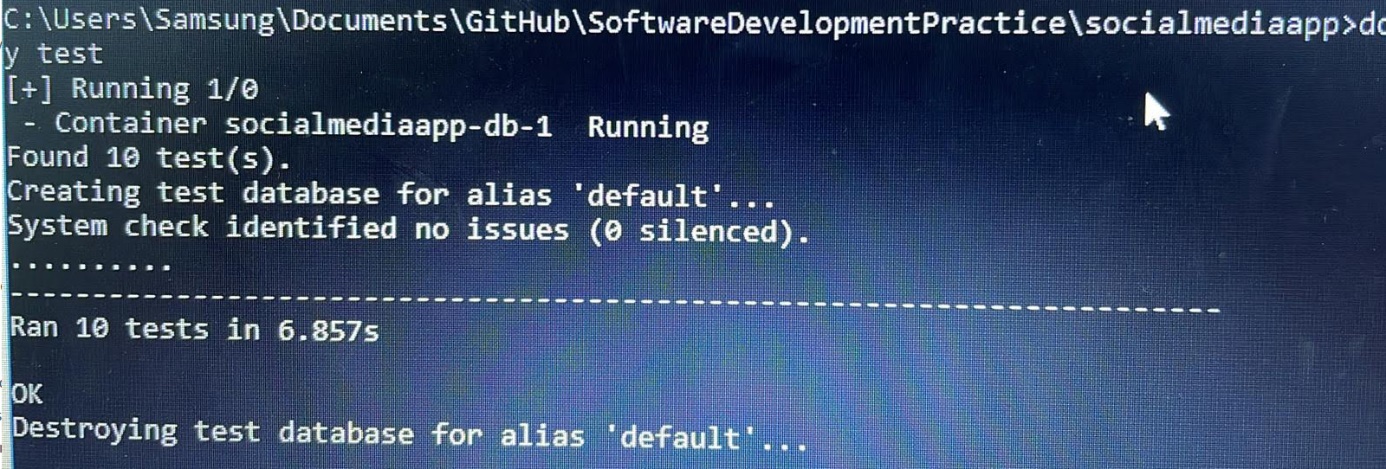
sudo apt install docker.io

1. After installing Docker on the instance, I completely moved my project folder to the instance using Filezilla program and using docker-compose build and up functions were enough to complete the project building process.
2. You can access your website from the "Public IPv4 address" that is given in the instance details on AWS.



# Test Results (Unit & User)

All the unit test have been successfully run as below. Unit tests cover the functionality that the home page links are successfully opened which would check that those pages are successfully opened and there are no critical errors. Test cases also cover that the posts can be created with correct information and user creation and authentication functions properly functions. Below is a screenshot that shows all the tests have been successfully executed.



## User Tests

All the users tests below have been successfully completed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Number** | **Requirement** | **Status** | **Detail** |
| Home-1 | Users shall be able to view the appropriate links to the main sections of the application when they open the home page of the application. a. Home b. Register c. Login / Logout d. Friends e. Spaces f. Profile g. Blogpost Entry | ✓ |  |
| Home-2 | If the user has not completed the login process then he can only see the contents that will be visible to non-logged in users which are homepage, login page and register page. | ✓ |  |
| Home-3 | If the user completes the login process successfully then he shall be able to view the main sections of the application such as All Posts, My Posts, My Profile and Users. | ✓ |  |
| Home-4 | Register and Login links on the homepage shall not be visible to logged in users. | ✓ |  |
| Home-5 | Users shall be able to view the blogposts that they previously saved. | ✓ |  |
| Home-6 | Users shall be able to view the blogposts of their friends. | ✓ |  |
| Home-7 | Users shall be able to delete their own blogspots if they want not to see the posts anymore. | ✓ |  |
| Home-8 | If a non logged-in user clicks the Post Entry, My Own Posts, User Profiles links, then he shall be directed to the login page. | ✓ |  |
| Register-1 | When a user clicks on the register link the “Register Page” shall be opened. | ✓ |  |
| Register-2 | User shall only be able to register after successfully completing the below information on the Register Page: o Username o E-Mail Address o First Name o Last Name o Password | ✓ |  |
| Register-3 | A user shall not be able to complete the registration process with a username that already exists in the system. | ✓ |  |
| Register-4 | A user shall enter password information twice so that he does not mistakenly enter a password and complete the registration with a wrong password. | ✓ |  |
| Register-5 | If the user enters different password information, he should see the error message: “**The two password fields didn’t match.”** | ✓ |  |
| Register-6 | The user shall not be able to continue registration with an e-mail address information that does not contain “@” character. | ✓ |  |
| Register-7 | The username that is taken as input at registration shall be at least 3 characters. | ✓ |  |
| Login / Logout-1 | When a non-logged in user clicks on the Login link the “Login Page” shall be opened. | ✓ |  |
| Login / Logout-2 | A user shall be able to login to the website by using the correct username and password combination that he created at the registration process. | ✓ |  |
| Login / Logout-3 | If the username is correct and the password does not match, the user shall get “Please enter a correct username and password. Note that both fields may be case-sensitive.” | ✓ |  |
| Login / Logout-4 | If the username entered does not exist, the user shall get the “Please enter a correct username and password. Note that both fields may be case-sensitive.” error message. | ✓ |  |
| Login / Logout-5 | If the user completes the login process successfully then he shall be able to view the main sections of the application. | ✓ |  |
| Login / Logout-6 | Logout link shall only be visible for logged in users. | ✓ |  |
| Login / Logout-7 | Whenever a user clicks on the logout link the logout page shall be opened and the user shall see the information that he is logged out. | ✓ |  |
| Login / Logout-8 | After the user has logged out of the system he shall not be able to view the profile page and he shall not be able create a blogpost entry. | ✓ |  |
| Users-1 | A user shall be able to follow other users of the application. | ✓ |  |
| Users-2 | A user shall be able to view all users in the Users section. | ✓ |  |
| Users-3 | When a user follows an other user, he shall be able to view the blogposts of that user in Friend Posts section. | ✓ |  |
| Users-4 | A user shall be able to view the profile page of other users and should be able to unfollow those users. | ✓ |  |
| Users-5 | A user shall be able to view last active date and e-mail address of other users in user detail pages. | ✓ |  |
| Search-1 | The users shall have the ability to search title and content of blogposts from the Search bar at the navigation menu. | ✓ |  |
| Search-2 | If the user searches for a keyword that does not exist in any title or content of a blogpost, then he shall get the error message: “Your search returned no results!” | ✓ |  |
| Search-3 | If the user searches for a keyword that only exists in Title section of a post, he shall be able to view that post. | ✓ |  |
| Search-4 | If the user searches for a keyword that only exists in Content section of a post, he shall be able to view that post. | ✓ |  |
| Tags-1 | The users shall be able to create spaces, which will give the users ability to share blogposts on a specific topic such as “Bosphorus University” using the tags capability of a blogpost. | ✓ |  |
| Tags-2 | The users shall be able to view all the blogposts related to tag. | ✓ |  |
| Profile-1 | The users shall have the ability to view their profile from the My Profile section of the application. | ✓ |  |
| Profile-2 | The users should have the ability to add a profile picture. | ✓ |  |
| Profile-3 | The users shall have the ability to view their profile information such as username and e-mail address, first and last name. | ✓ |  |
| Profile-4 | The users shall have the ability to update their name and e-mail address information but they should not have the ability to update their username from this section. | ✓ |  |
| Blogpost-1 | The users shall have the ability to create blogposts with entering the title, content (note about the blogpost) and URL of the related web page. | ✓ |  |
| Blogpost-2 | Title, content, and the URL fields shall be mandatory for blogpost entry. | ✓ |  |
| Blogpost-3 | The date, time and author of the blogpost should be recorded in the system. | ✓ |  |
| Blogpost-4 | The users shall be able to update the title, content, tags and URL of a blogspot. | ✓ |  |
| Blogpost-5 | The users shall be able to delete their own blogposts. | ✓ |  |
| Blogpost-6 | When a user deletes his own blogspot, no other user shall be able to see that post. | ✓ |  |
| General-1 | The blogposts shall have been correctly grouped into following sections: All Posts: All posts shall exist here. My Posts: This section shall show the previous posts of the logged in user. Friend Posts : This section shall show the posts of other users that the logged in user has followed before. | ✓ |  |
| General-2 | The links of the blogposts shall be represented with “Go To Link” green button and this button should direct the user to the related web page. | ✓ |  |
| Save-1 | Users shall be able to save the posts of other users. | ✓ |  |
| Save-2 | Users shall not be able to save their own posts. Save button shall not be visible. | ✓ |  |
| Save-3 | Users shall be able to see the blogposts that they saved in the “Posts I Saved” section. | ✓ |  |
| Comment-1 | Users shall be able to add comments to the blogposts. | ✓ |  |
| Comment-2 | The commenting user and his profile photo, his comment and comment timestamp shall be visible under the detail page of the related blogpost. | ✓ |  |
| Usability Perform.-1 | All pages on the website shall be loaded within 5 seconds for at least %99 of the users. | ✓ |  |
| Usability Perform.-2 | The website shall have the capacity for at least 5.000 logged-in users at any given time. | ✓ | Current number of users did not have any issues. |
| Usability Perform.-3 | The website must have at least 95% uptime. | ✓ | It is %100 up for the last month. |
| Usability Perform.-4 | The application must be scalable enough to support 100.000 visits at the same time while maintaining its optimum performance. | ✓ |  |
| Usability Perform.-5 | The application shall be mobile compatible and the pages shall be opened according to the mobile phone screen size. | ✓ | Mobile phone tests have been done using two different IOS and Android phones. |
| Security-1 | The password information shall never be visible at any page of the website. | ✓ |  |
| Security-2 | The username and password records shall not be viewed or updated by unauthorised users. | ✓ |  |

# References

[Python Django Tutorial: Full-Featured Web App Part 1 - Getting Started – 31.08.2018 – Corey Schafer](https://www.youtube.com/watch?v=UmljXZIypDc&t=641s)

(This tutorial has been of great help to start at the beginning of the project to get the basics of creating a web application with Django.)

# Additional Information

These are the additional notes on the technical side that may be helpful for similar future projects.

1. Using Bootstrap for frontend development would be very helpful as it would speed up the front-end development part and the application would look much better.
2. Before publishing your code to Github, it would be a wise practice to use a .gitignore file to prevent unwanted content to be delivered on Github.
3. When dockerising the application with mySQL and running the containers the web part was trying to start before the database was ready. Even the DB seems to be started in Docker Desktop, it was not fully ready for use. To avoid this problem healthcheck mechanism has been used, the following part was added to docker-compose.yml file.

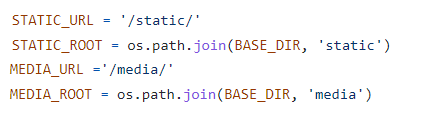


1. When dockerising the application with mySQL and having mySQL at your local environment, you may face the error message:” *Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:3306 -> 0.0.0.0:0: listen tcp 0.0.0.0:3306: bind: Normal olarak her yuva adresi (iletişim kuralı/ağ adresi/bağlantı noktası) için yalnızca bir kullanıma izin veriliyor.”* To fix the error you may update the port number with 3307 in the docker-compose.yml file.



1. Debug Mode Update – When the debug mode is changed to False, the CSS and the images will not be working properly, to fix the issue you may do the below updates on settings.py and urls.py files.

Settings.py



urls.py



1. Before cloud deployment, make sure that the Python and mySQL versions you use are using are compatible with the cloud service provider. If you select AWS as the cloud service provider, you may read the above steps described in the “AWS Deployment” section of the report.
2. It is a good practice to use .env files for storing secret information, you may use the below code after installing dotenv as follows.

*pip install python-dotenv*

