

SWE 573 2022 Fall Software Development Practice

Project Final Deliverables



Name Surname	Damla Alkan
Course Name	SWE573
Student ID	2021719000
Date	03.01.2023
Project Name	KeeP
Git Repo	damlaalkan/SWE573: Boun SWE573 Project (github.com)
Git Tag Version:	v0.9
Deployment URI	KeeP : http://54.166.23.109/

HONOR CODE

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

Name Surname: Damla Alkan

Signature:

Table of Contents

1. Overview	3
2. Software Requirements Specification	3
2.1.1. Functional Requirements	3
2.2.1. Non-Functional Requirements	6
3. Design	7
3.1. Mockups	7
3.2. UML Diagrams	9
3.2.1. UML Class Diagram.....	9
3.2.2. UML Use Case Diagram	9
4. Status of the project	10
5. Status of Deployment.....	16
6. System manual	16
7. User manual	17
8. Test Results	18
9. Functionality Video	18

1. What is KeeP ?

KeeP is a website that requires registration and after registering, you can post a post with the website link and see other people's similar posts as long as you add them as a friend. This website allows you to note the pages you like while browsing the internet and to follow the pages that others like to see them.

The **KeeP** website “keeps” all the information that interests you and that you want to remember.



2. Credentials

Test user credentials:

- Username : test
- Password : Password123.

Admin user credentials : <http://54.166.23.109/admin>

- Username : damla
- Password : Password123.

Pages (count = 8):

1. Welcome Page : <http://54.166.23.109/>
2. Login Page : <http://54.166.23.109/accounts/login/>
3. Reset Password Page : http://54.166.23.109/accounts/password_reset/
4. Reset Password Done Page : http://54.166.23.109/accounts/password_reset/done/
5. Signup Page : <http://54.166.23.109/accounts/signup/>
6. Dashboard Page : <http://54.166.23.109/accounts/dashboard/>
7. My Friends Page : http://54.166.23.109/accounts/profile_list/
8. User page : <http://54.166.23.109/accounts/profile/5> (<http://54.166.23.109/accounts/profile/%user-number%/>)

Notes for testing:

- *You may sign-up with new user and reset password with test user.*
- *You need to follow and unfollow users to see their dashboard.*

3. Software Requirements Specification

2.1.1. Functional Requirements

- The system must allow users to log into their account by entering their email and password.
- The system must allow users to log into their social media accounts including Facebook, Google, Microsoft, Twitter.
- The system must allow users to reset their password by clicking on "I forgot my password" and receiving a link to their verified email address.
- The system must allow admin to add account manually.
- The system should enable users to give user details as name, surname, age, gender and location.
- The system must allow users to create a new post which users can provide link, tag, comment, image, provider and category.
- The system must allow users to see the created post in feed page.
- The system must allow users to search posts by category, provider, date and tag.
- The system must allow users to edit posts.
- The system must allow users to follow and unfollow other accounts
- The system must allow users to see followed accounts' posts.

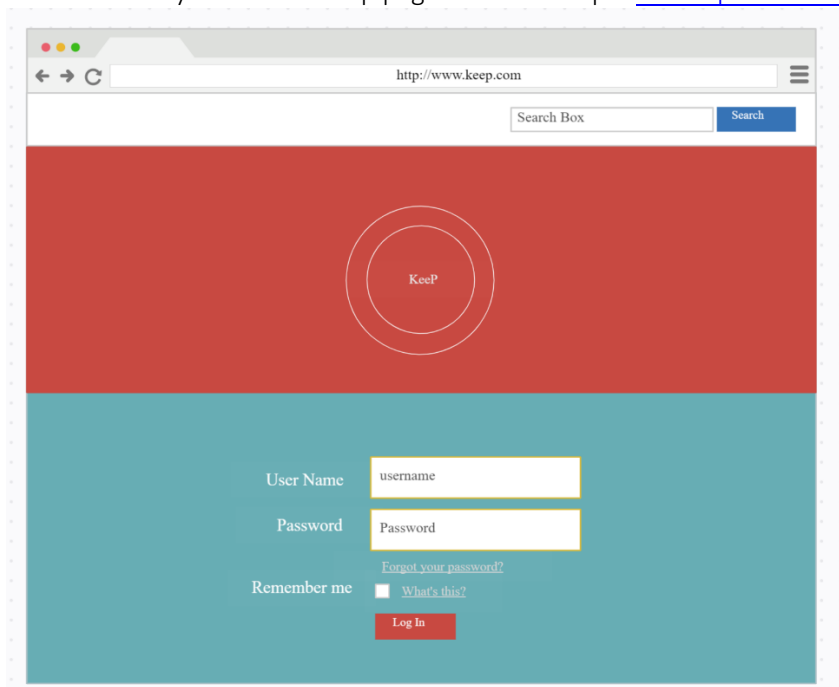
2.2.1. Non-Functional Requirements

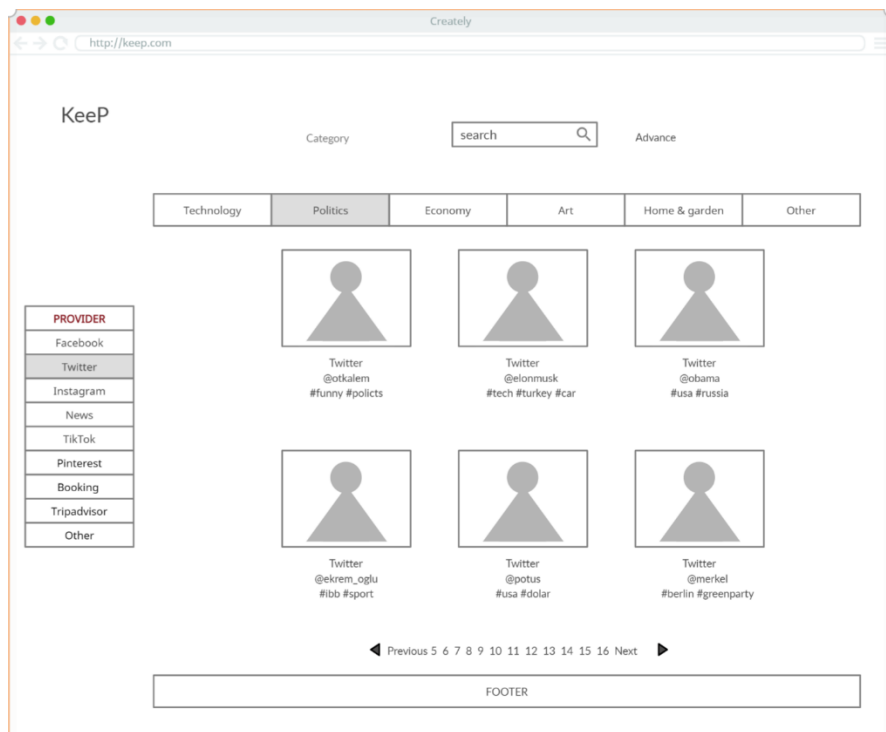
- Each page must load within maximum 2 seconds.
- Creating post must load withing max 4 seconds.
- The application must be dockerized.
- The application must be deployed to cloud.
- The application hosting must be on AWS.
- The application database must be PostgreSQL.
- The application language must be in English.
- Every attempt by a user to edit or delete an item of data shall be recorded on an audit trail.
- A website should be capable enough to handle 20 million users with affecting its performance.
- The software should be platform as a service to always be up to date.

3.1. Design (Software and Mockup)

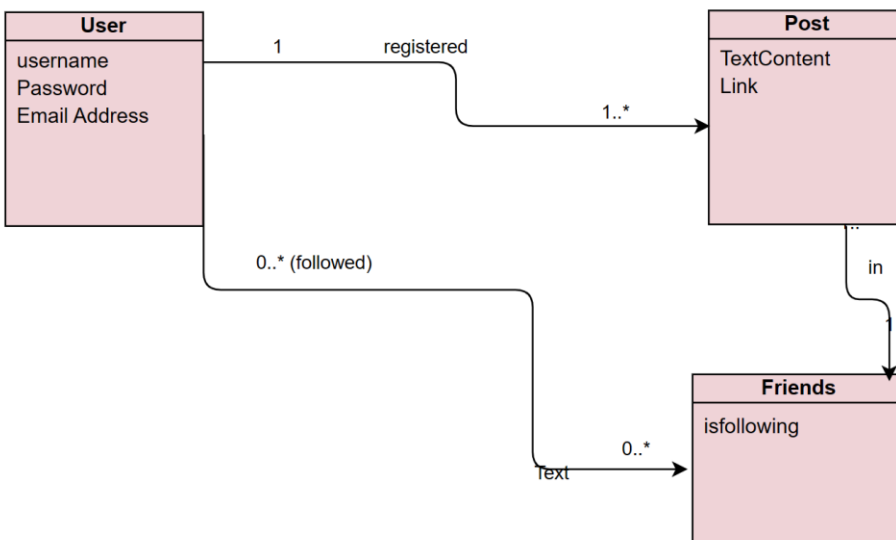
3.1.1. Mockup

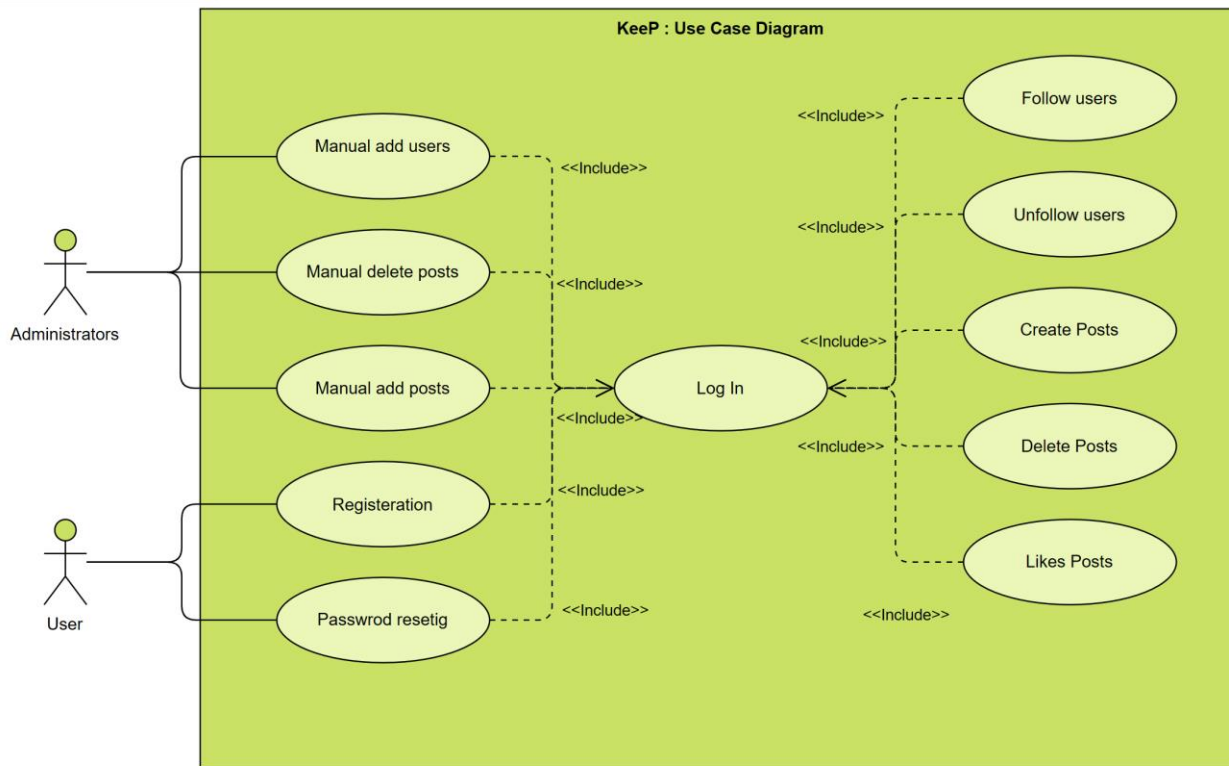
You may find the Mocup page in Github Repo: [Mockup - damlaalkan/SWE573 Wiki \(github.com\)](https://github.com/damlaalkan/SWE573-Wiki)





3.1.2. UML Diagrams





4. Status of the project

You may find each requirements , status and issue link in the below chart.

You may find all issues here: <https://github.com/damlaalkan/SWE573/issues/>

Requirements	Status	Issues
The system must allow users to log into their account by entering their email and password.	Completed	Creating the login page · Issue #20 · damlaalkan/SWE573 (github.com)
The system must allow users to log into their social media accounts including Facebook, Google, Microsoft, Twitter.	Not Completed	Add Login Page Social Media Sign in with Google,Facebook, Google, Microsoft, Twitter. · Issue #47 · damlaalkan/SWE573 (github.com)
The system must allow users to reset their password by clicking on "I forgot my password" and receiving a link to their verified email address.	Completed	Create Signup · Issue #28 · damlaalkan/SWE573 (github.com)
The system must allow admin to add account manually.	Completed	Creating Authantication App · Issue #19 · damlaalkan/SWE573 (github.com)
The system should enable users to give user details as name, surname, age, gender and location.	Partially Completed	Creating the login page · Issue #20 · damlaalkan/SWE573 (github.com)
The system must allow users to create a new post which users can provide link, tag, comment, image, provider and category.	Partially Completed	Create Post class with link, category and body features as form · Issue #32 · damlaalkan/SWE573 (github.com)
The system must allow users to see the created post in feed page.	Completed	Design Dashboard page with Post Form · Issue #34 · damlaalkan/SWE573 (github.com)

The system must allow users to search posts by category, provider, date and tag.	Partially Completed	Create Post class with link, category and body features as form · Issue #32 · damlaalkan/SWE573 (github.com)
The system must allow users to edit posts.	Not Completed	Adding Edit feature to post · Issue #48 · damlaalkan/SWE573 (github.com)
The system must allow users to follow and unfollow other accounts	Completed	Create Profile with follow feature · Issue #29 · damlaalkan/SWE573 (github.com)
The system must allow users to see followed accounts' posts.	Completed	Design Dashboard to show followed user's posts · Issue #35 · damlaalkan/SWE573 (github.com)
Each page must load within maximum 2 seconds.	Completed	https://github.com/damlaalkan/SWE573/issue/s/26
Creating post must load withing max 4 seconds.	Completed	Migrate sqllite3 to postgresql · Issue #26 · damlaalkan/SWE573 (github.com)
The application must be dockerized.	Completed	Dockerize app and database postgresql · Issue #44 · damlaalkan/SWE573 (github.com)
The application must be deployed to cloud.	Completed	Deploy docker container app and database to cloud · Issue #45 · damlaalkan/SWE573 (github.com)
The application hosting must be on AWS.	Completed	Create container registry and container app on cloud · Issue #46 · damlaalkan/SWE573 (github.com)
The application database must be PostgreSQL.	Completed	Migrate sqllite3 to postgresql · Issue #26 · damlaalkan/SWE573 (github.com)
The application language must be in English.	Completed	Creating a new Django project · Issue #17 · damlaalkan/SWE573 (github.com)
The software should be platform as a service to always be up to date.	Partially Completed	Create container registry and container app on cloud · Issue #46 · damlaalkan/SWE573 (github.com)

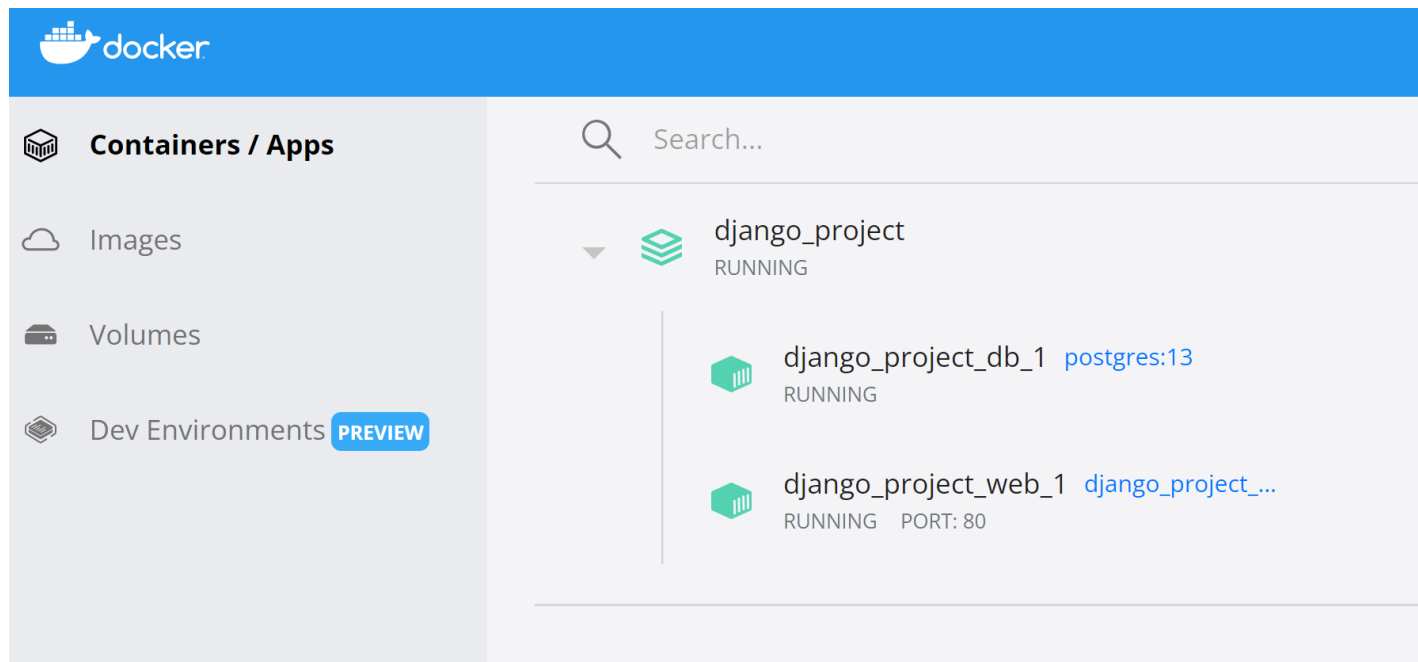
5. Status of Deployment

The application is dockerized and deployed. You may find the Dockerfile and docker-compose.yml in Github Repo. You may find snapshot of docker container, docker images and deployed links.

You may find dockerize steps here : [Dockerize Django App · damlaalkan/SWE573 Wiki \(github.com\)](#)

Followed steps for cloud deployment : [Running Ubuntu Desktop on an AWS EC2 instance | Ubuntu](#)

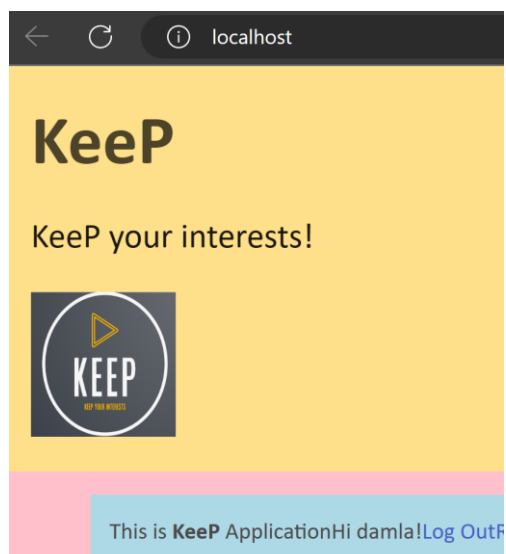
Docker Container:



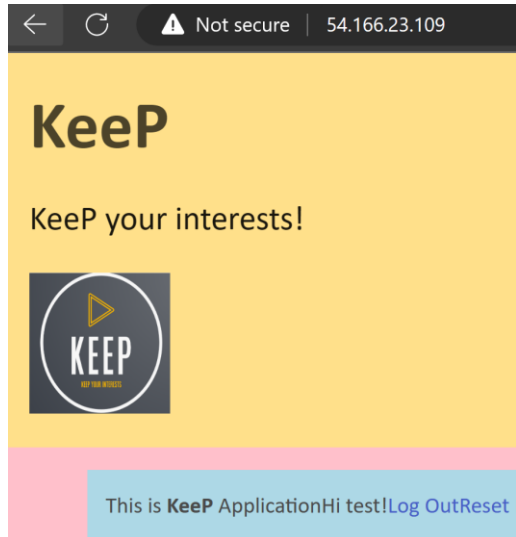
Docker Image:

django_project_web	IN USE	latest	28e0a2c8c5eb	7 days ago	169.04 MB
postgres	IN USE	13	9a26a0154be3	11 days ago	373.39 MB

Local running:



Deployed to Cloud :



6. System manual

You may find the instructions to run system manually:

1. Install Docker Desktop
 - [Docker for Mac](#)
 - [Docker for Windows](#)
 - [Docker for Linux](#)
2. Start Docker Desktop
3. [Install Python 3.10](#)
4. Open a terminal and run following commands
5. git clone <https://github.com/damlaalkan/SWE573/>
6. cd KeeP_Project
7. docker-compose up
8. Open your browser and go to *localhost*
9. Signup with a new user
10. Go to dashboard
11. Create posts and follow new users

7. Functionality Video

You may find the demonstration video in Youtube link : [SWE573-Damla-Alkan-KeeP-Application - YouTube](#)