SWE573 Software Development Practice 2022 Fall

Ceyda Düzgeç 2021719198

Advisor: Suzan Üsküdarlı

January 3, 2023

TABLE OF CONTENTS

1. Honor Code

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

Ceyda Düzgeç

2. Overview

The name of my project is Sole, comes from the meaning sun in Italian. It is a basic attempt to make a social media app but in a little bit different approach. This app is based on sharing URL's. So it acts as a database of links with a fancy user interface and social aspect. It aims to connect and interact people with similar interests to share their opinions about the external links and furthermore act on these links.

Used languages:

- Python
- CSS
- HTML

Used frameworks and technologies:

- Django
- PostgreSQL
- Docker
- AWS
- Bootstrap
- Git
- Github

Milestone 1 - 8.11.2022 - Requirements and basic infrastructure with login.

Milestone 2 - 6.12.2022 - Fundamental features, Dockerized, deployed

Milestone 3 - 27.12.2022 - Complete features, Dockerized, deployed

Final Deliverables - Report and demo video

- Git Repository: `https://github.com/ceydaduzgec/SWE573`
- Git Wiki: `https://github.com/ceydaduzgec/SWE573/wiki`
- Deployed app link: `http://ec2-54-91-110-14.compute-1.amazonaws.com/`

3. Software Requirements Specification

3.1. Glossary

- **Admin:** The user role for people who has all authorization to use admin panel to make changes on the database including deleting glimpses and users, changing user roles.
- **Discover:** The page for list of recommended glimpses for the specific user.
- **Feed:** The page for list of followed user's glimpses.
- **Glimpse:** A glimpse is a collection of data in a certain format that consists of a title, description, URL and category.
- **Guest:** A user role for people who doesn't log in to the system.
- **Space:** A collective place for a group of people to talk about certain topics and moderated by the admin of the group.
- **Space Admin:** The admin role only for a specific space.
- **Registered User:** A user role for people who loges in to the system after signing up.

3.2. Elicitation Questions

- What's the problem or need that this application will address?
- On which platform will the application run (is it a mobile/web application)?
- Is there any similar app?
- How many people are expected to use this application?
- Who can join the system?
- What will be the differences from other social media platforms?
- What will happen in case of misconduct, abuse, or disagreement?
- Is there a ban or report system? How does it work?
- Can users recover their passwords?
- Does system access to the location of the user?

- What will be the sections in the app that users can navigate?
- Does the app have a friendship/following mechanism among profiles?
- How can the users have ownership of their own data?
- How can the users interact with the glimpses?
- How many glimpse categories there will be?
- Can users create new category types?
- Is there a 3rd party app integration?
- How will the users interact with each other?
- Is there an admin role? What admin can do and can't?

3.3. User Scenarios

Suzan is an academic researcher who is passionate about collecting information from the social media. She likes to connect with her social network and interact with them, she chats with new people online. And she also likes to participate in physical meetings about her interest areas. She wants to create an online archive for herself to share with her network. But since she is using so many different social media platforms, she needs to store the information on different app's databases. She want to use only one app with her own database.

1- Suzan signs up to the Sole app.

2- She wants to find the topics she is interested.

3- She searches "salça".

4- She finds public glimpses and spaces related with "salça".

5- She joins a space named "salça lovers".

6- She shares a glimpse to "salça lovers" with a related URL to her favorite salça blog.

7- Some people who she doesn't know comments to the glimpse about how to make a better salça.

8- She connects with these people.

9- She sees a glimpse of these people about a salça festival.

10- She bookmarks this glimpse to remind her self to attend the festival.

Ceyda is a procrastinator who likes to waste time on social media when she have to work instead. She sees a lot of posts unrelated to her. She also likes to create archive for cafes, restaurant and art galleries she want to visit but always gets lost in the apps and can't find anything that she saves in the sea of post in other apps.

1- Ceyda signs up to the Sole app.

2- She starts following recommended users.

3- She scrolls the feed.

4- She sees some cool blog glimpses about Django but she doesn't want to read now.

5- She bookmarks them.

6- She eventually needs to do her project homework.

7- She remembers the glimpses about Django.

8- She logs into the app, goes to her bookmarks and read them.

İlayda is a blogger. She likes to write about stuff like books, movies, theatres, TV shows, concerts, cities, food, basically everything she experiences. But she cannot gain new followers on her blog, it is hard to promote her blog on social media apps since URL's are hard to share on the posts. She is searching for an app that she can promote her blog.

1- İlayda signs up to the Sole app.

2- She starts discovering spaces.

3- She shares her blog posts on one of the spaces.

4- Her glimpse gets comments.

5- She gets followers for her blog.

3.4. Requirements

3.4.1. Functional Requirements

3.4.1.1. User Requirements.

- Users should be able to register with an email and a password.
- Users should be able to login with the email and password they had registered.
- Users should be able to change their password using their e-mail address or old password.
- Users should be able to change their password via a link sent to their registered e-mail by the system.
- Users should be able to sign out from the system.

- Users should be able to create their own account and fill their personal information.
- Users should be able to update their accounts.
- Users should be able to report inappropriate glimpses and users.
- Users should be able to set the information in their account to be public or private or draft.
- Users are able to set their privacy for each glimpse.
- Users are able to delete their accounts.
- Users should be able to create, edit and delete glimpses.
- Users should be able to tag glimpses.
- Users should be able to select category of glimpse while creating.
- Users should be able to view the glimpses they created and liked.
- Users should have draft box to save unfinished glimpses.
- Users should be able to like or unlike glimpses.
- Users should be able to add comments to the comment section of the glimpses.
- Users should be able to edit and delete their comments.
- Users should be able to follow other users.
- Users should be able to view the glimpses of people they follow.
- Users should be able to filter the glimpses on the feed by category.
- Users should be able see recommendations page.
- Users should be provided with recommendations based on their preferences and restrictions.
- Users should be able to search for other users or glimpses.

3.4.1.2. System Requirements.

- The system should allow three kind of users to surf: admin, registered, guest.
- The system should require email confirmation on sign up.
- The system should provide users with a password recovery mechanism.
- The system should recommend glimpses to users.
- The system should be able to do search.

3.4.2. Non-Functional Requirements

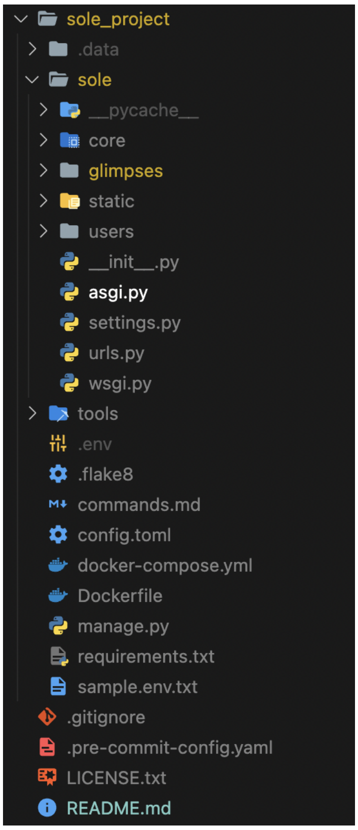
3.4.3. System Requirements

- The system should require strong password from the user.
- The system should be available in all languages.
- The system should have a web application.
- The system should use only free libraries.
- The libraries and packages shouldn't have a security vulnerability.
- The system should use encryption to store passwords.

4. Design

4.0.1. Software

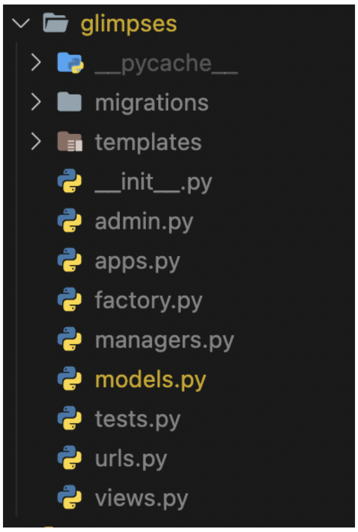
The Sole project is based on Django framework and using Bootstrap library and PostgreSQL database. The project folder setup:



```
sole_project
  .data
  sole
    __pycache__
    core
    glimpses
    static
    users
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
  tools
  .env
  .flake8
  commands.md
  config.toml
  docker-compose.yml
  Dockerfile
  manage.py
  requirements.txt
  sample.env.txt
.gitignore
.pre-commit-config.yaml
LICENSE.txt
README.md
```

Figure 4.1. Project Setup

The main folder is named *SWE573* it stores the Github actions folder, *.gitignore, .pre-commit-config.yaml, LICENSE.txt and README.md.* Pre-commit is a hook that forces user to follow the linter and code formatter rules before committing the code and this file stores the configuration of it as it can be understood from its name. This folder also stores the root folder of the Sole project which is the *sole_project* The *.data* folder stores the local database, Docker volumes and it is in *gitignore* that is why it is in grey color.



```
glimpses
  __pycache__
  migrations
  templates
  __init__.py
  admin.py
  apps.py
  factory.py
  managers.py
  models.py
  tests.py
  urls.py
  views.py
```

Figure 4.2. glimpse App

The *sole* folder stores the main app and it's settings. The *core* app folder stores the constants and main page templates. The *glimpse* app folder stores the admin functionalities, urls, views, templates, tests and related configurations such as managers and test mocks. The *user* app acts in the same way for the user related functionalities. It also contains *static* folder with bootstrap files. Lastly, the remaining files: *asgi, settings, wsgi* contains the settings of the app and *urls* contains the main url configuration of the app.

The *tools* folder to run Docker easier. Of course the Docker files. I also implemented *flake8, isort, black* for linting and code formatting, the *.flake8, config.toml* files are for the configuration of these. The *manage.py* is for running the Django commands and *requirements.txt* is for installing the related packages and libraries in the beginning. I also added *commands.md* for myself to put most common commands that I use for ease of access. Lastly, I also added a sample .env file for the public use purposes.

4.0.2. Mock-ups

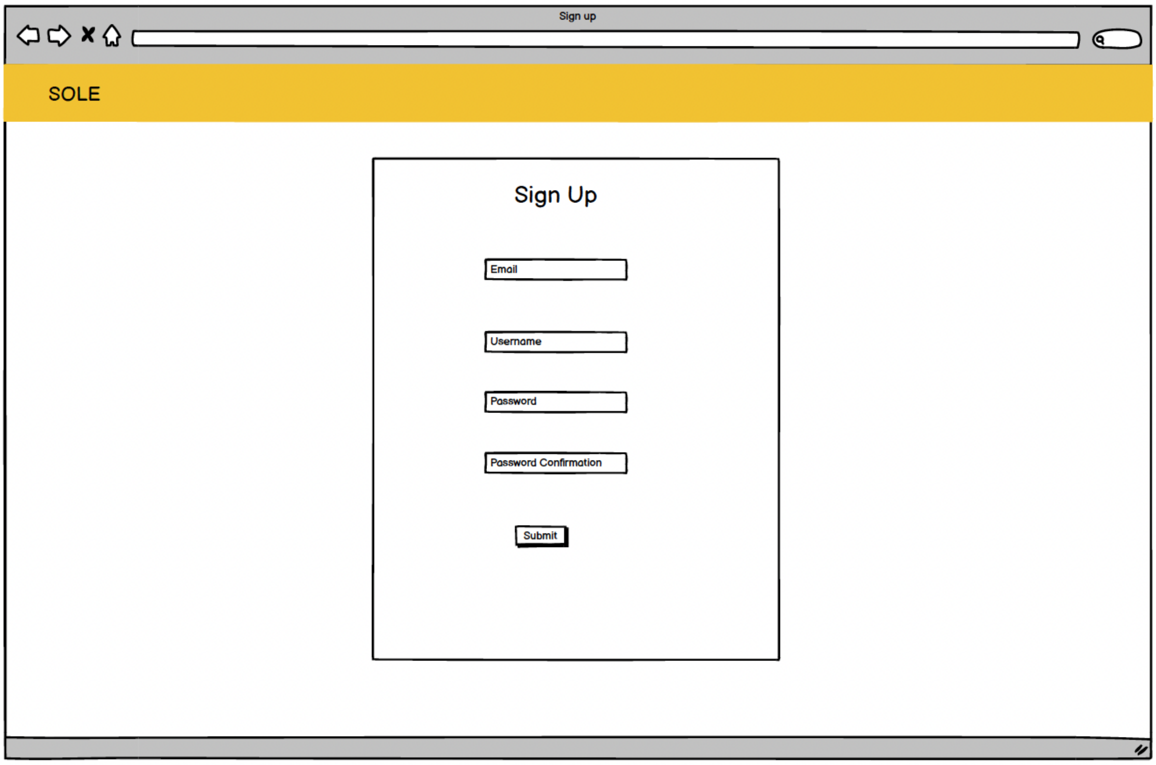


Figure 4.3. Sign up Page

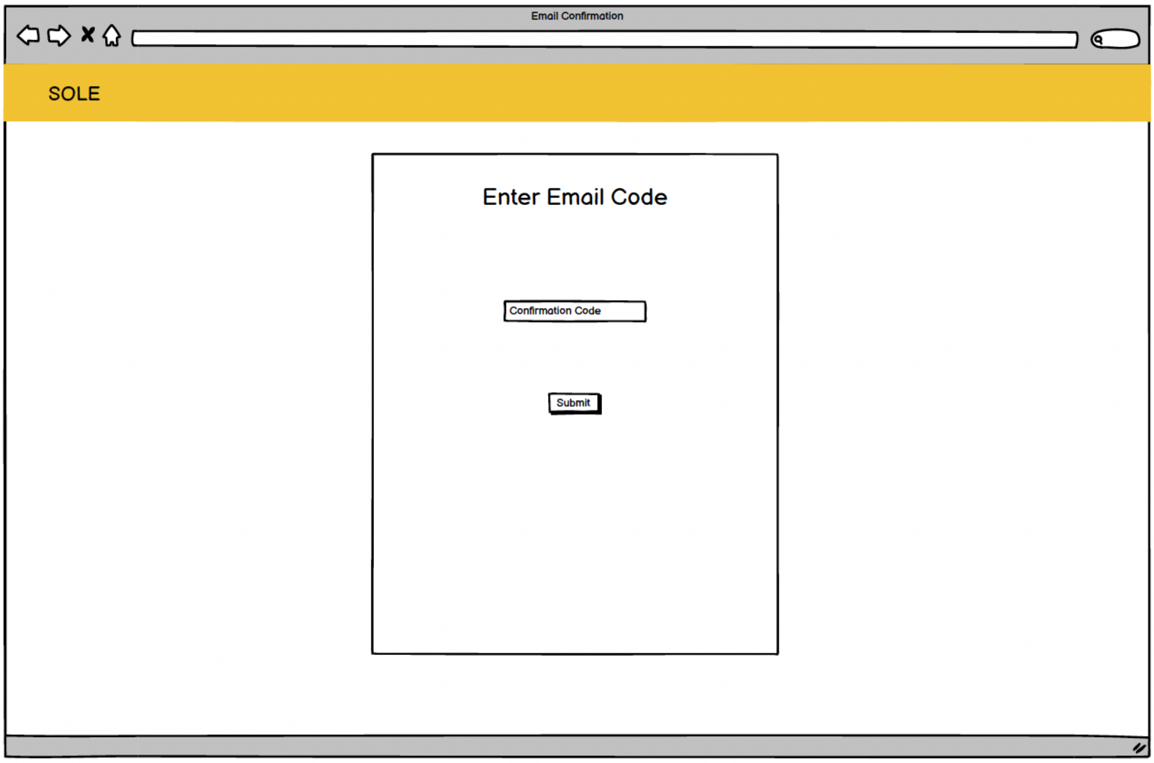


Figure 4.4. Email Confirmation Page

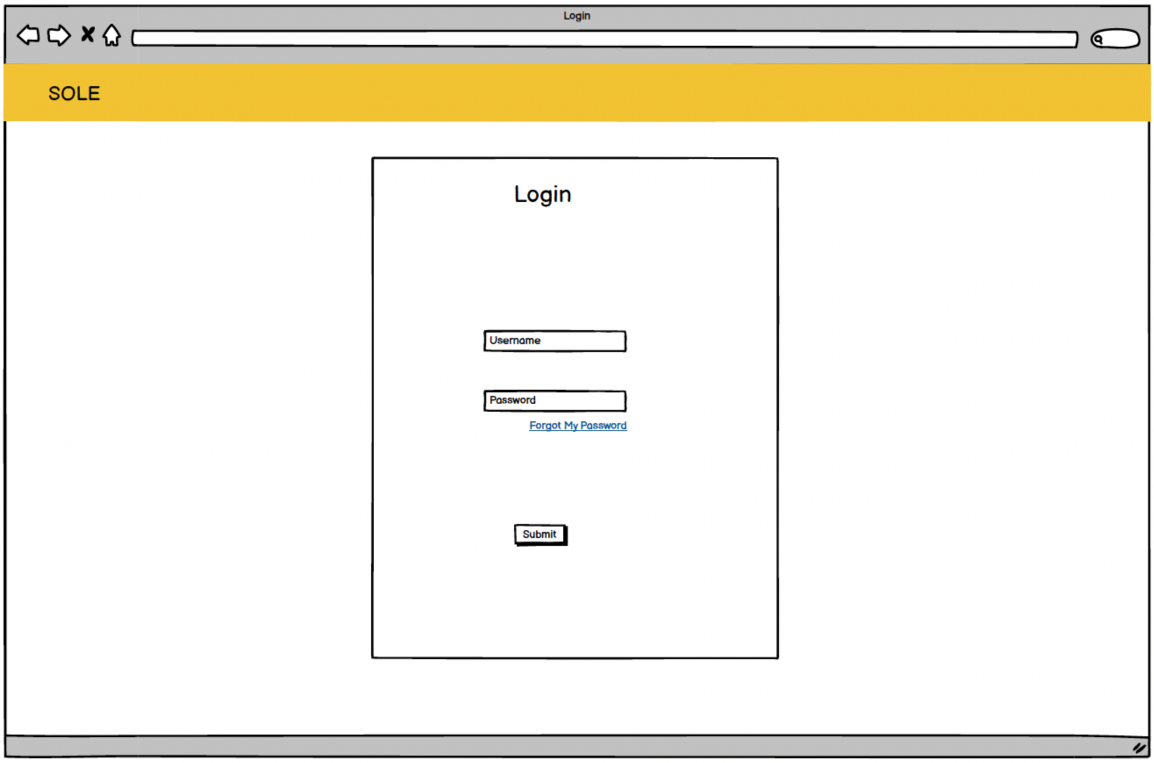


Figure 4.5. Login Page

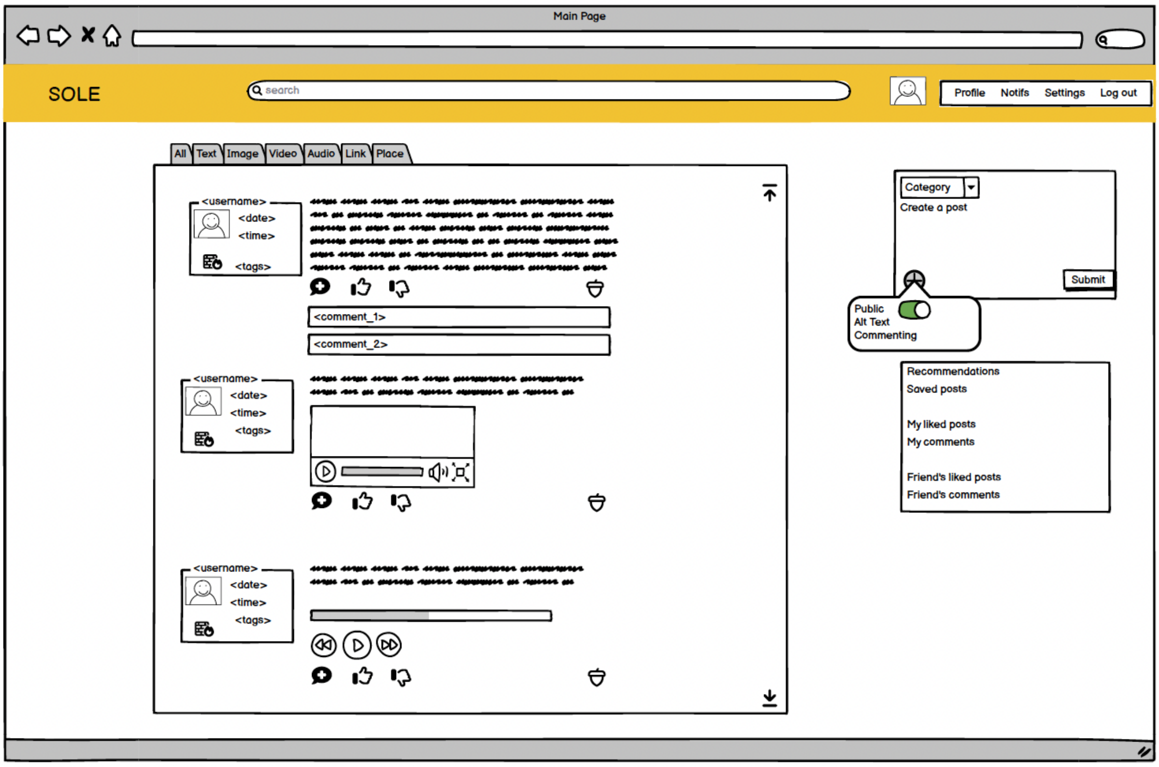


Figure 4.6. Feed Page

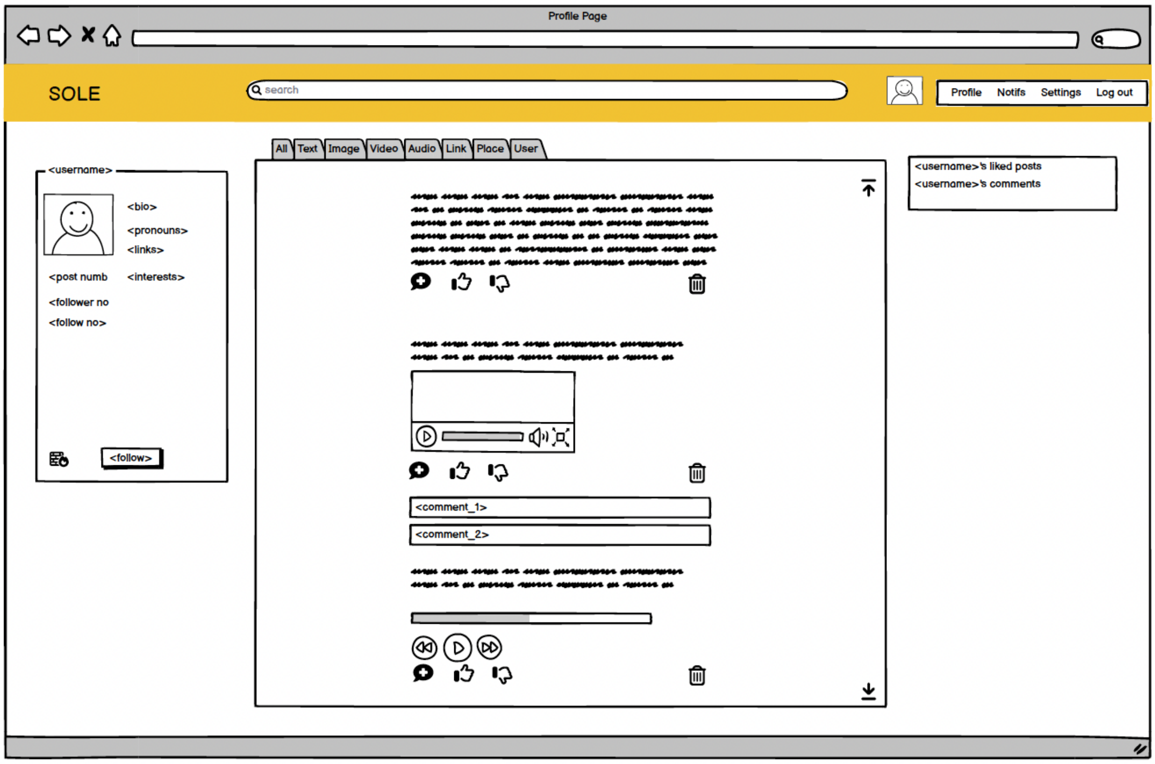


Figure 4.7. Profile Page

4.0.3. Diagrams

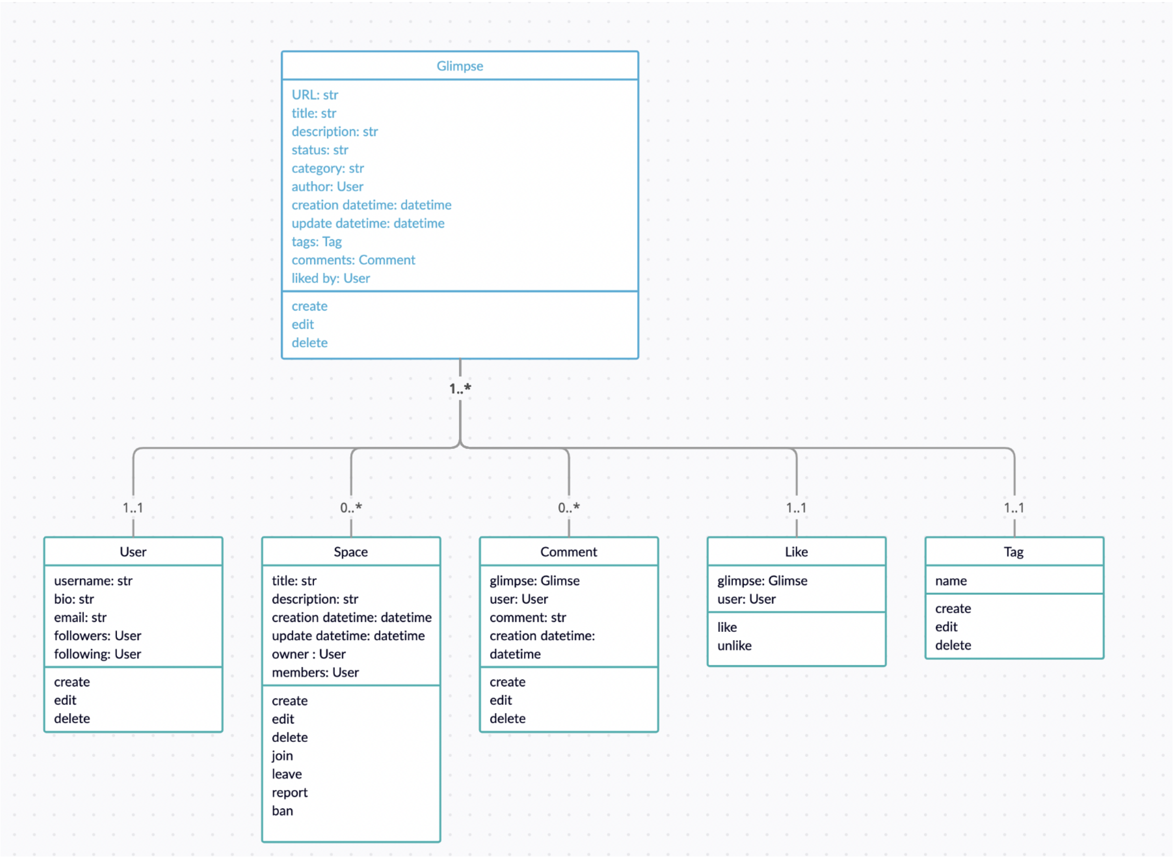


Figure 4.8. UML Diagram

5. Status of the Project

The project management board can be accessed from here:

`https://github.com/users/ceydaduzgec/projects/1`

The milestones can be accessed from here:

`https://github.com/ceydaduzgec/SWE573/milestones?state=closed`

- Meanings of the marks.
  - ☑ Done fully and deployed.
  - ☒ Started but couldn't finish and deployed.
  - □ Didn't start.

5.0.1. Functional Requirements

5.0.1.1. User Requirements.

- ☑ Users should be able to register with an email and a password.
- ☑ Users should be able to login with the email and password they had registered.
- ☑ Users should be able to change their password using their e-mail address or old password.
- □ Users should be able to change their password via a link sent to their registered e-mail by the system.
- ☑ Users should be able to sign out from the system.
- ☑ Users should be able to create their own account and fill their personal information.
- ☒ Users should be able to update their accounts.
- □ Users should be able to report inappropriate glimpses and users.

☒ Users should be able to set the information in their account to be public or private or draft.

☑ Users are able to set their privacy for each glimpse.

□ Users are able to delete their accounts.

☑ Users should be able to create, edit and delete glimpses.

☑ Users should be able to tag glimpses.

☑ Users should be able to select category of glimpse while creating.

☒ Users should be able to view the glimpses they created and liked.

☑ Users should have draft box to save unfinished glimpses.

☑ Users should be able to like or unlike glimpses.

□ Users should be able to add comments to the comment section of the glimpses.

□ Users should be able to edit and delete their comments.

☒ Users should be able to follow other users.

☒ Users should be able to view the glimpses of people they follow.

□ Users should be able to filter the glimpses on the feed by category.

□ Users should be able see recommendations page.

□ Users should be provided with recommendations based on their preferences and restrictions.

☑ Users should be able to search for other users or glimpses.

5.0.1.2. System Requirements.

☑ The system should allow three kind of users to surf: admin, registered, guest.

□ The system should require email confirmation on sign up.

□ The system should provide users with a password recovery mechanism.

□ The system should recommend glimpses to users.

☑ The system should be able to do search.

5.0.2. Non-Functional Requirements

5.0.3. System Requirements

☑ The system should require strong password from the user.

□ The system should be available in all languages.

☑ The system should have a web application.

☑ The system should use only free libraries.

☑ The libraries and packages shouldn’t have a security vulnerability.

☑ The system should use encryption to store passwords.

6. Manuals

6.1. System Manual

Prerequisites

- Download Docker

Installation

1. Clone the repository.

```
git clone https://github.com/ceydaduzgec/SWE573.git
```

2. Open project with your favorite code editor and a new terminal.

```
cd sole_project
```

3. Run Docker infastructure.

```
source tools/run_development.sh
```

4. Rename the ”sample.env.txt” to ”.env” file under sole_project and put related values.

5. Migrate the database.

```
python manage.py migrate
```

6. (Optional) Create a super user for admin panel.

python manage.py createsuperuser

7. Run the server.

python manage.py runserver 0:8000

8. Go to *http://127.0.0.1/* or *http://127.0.0.1:80/* from your browser.

More information can be found in the project repository and if any error occurs an issue can be created from here: https://github.com/ceydaduzgec/SWE573/issues/new/choose

6.2. User Manual

Access to website: `http://ec2-54-91-110-14.compute-1.amazonaws.com/`

User can sign up to the system by clicking the Sign Up buttton on the top right corner of the system.[Figure 6.1]

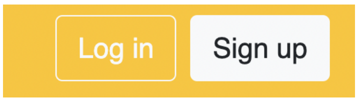
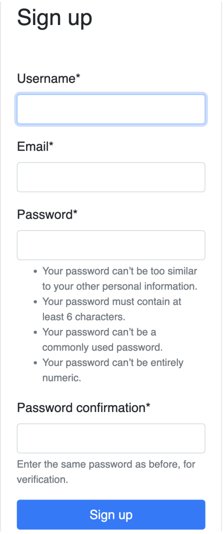


Figure 6.1. Sign up - Log in buttons

A form will open in the screen. In this form user should fill out the required username, email, password confirmation fields. The username should be only lowercase English letters, period and underscore characters with minimum length of three characters. The password should be a strong one, also password and confirmation should match with each other. [Figure 6.2]

After signing up user will be welcomed by the public feed.[Figure 6.3] The sidebar is a navigator for the user. The sidebar contains, *Share a glimpse, My glimpses, My Drafts, Settings, Log out* sections. Also the navigator bar on the top contains the app name, username and search bar.



Sign up

Username*

Email*

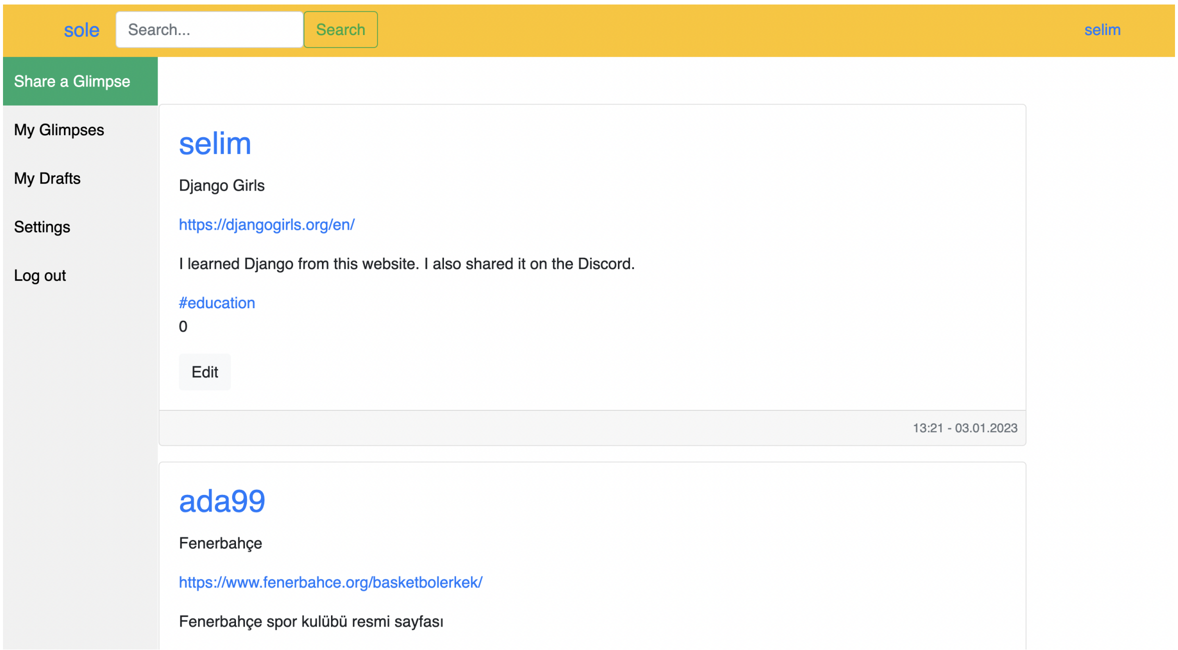
Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 6 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

Figure 6.2. Sign up page



selim

Django Girls

https://djangogirls.org/en/

I learned Django from this website. I also shared it on the Discord.

#education
0

13:21 - 03.01.2023

ada99

Fenerbahçe

https://www.fenerbahce.org/basketbolerkek/

Fenerbahçe spor kulübü resmi sayfası

Figure 6.3. Feed

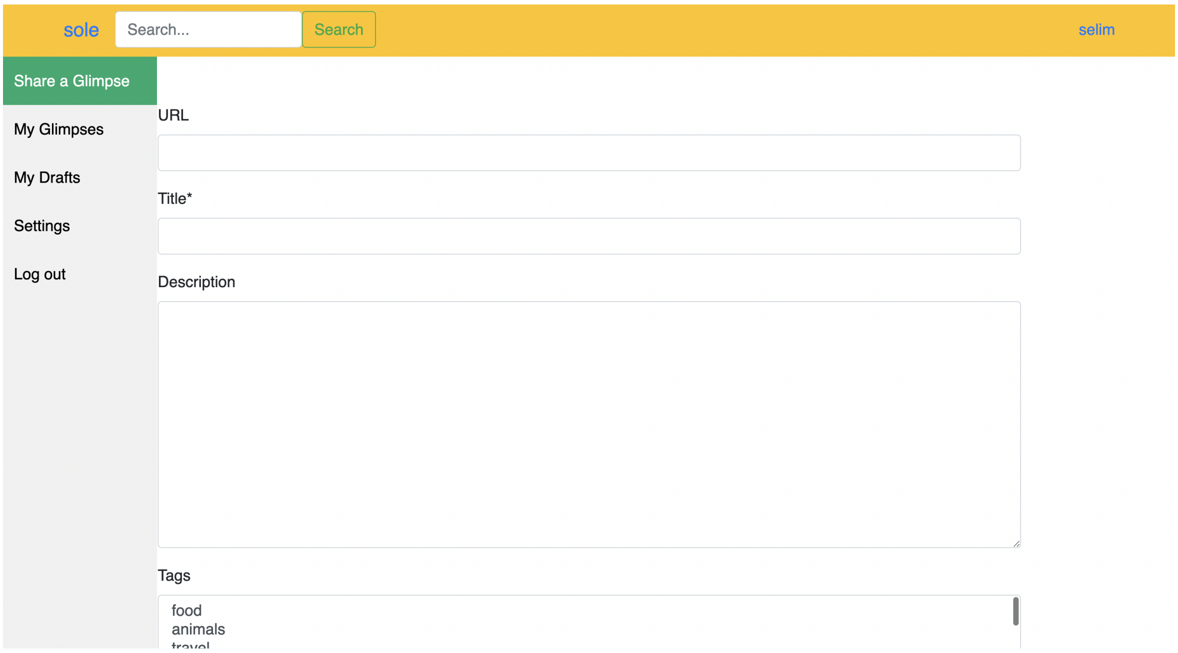


Figure 6.4. Share a glimpse

Creating a glimpse requires logging in. If the user is logged, they can easily share a glimpse by clicking the green *Share a glimpse* on the top left corner and will go to the related page.[Figure 6.4] Each glimpse requires URL, title, description, tag, category and status.

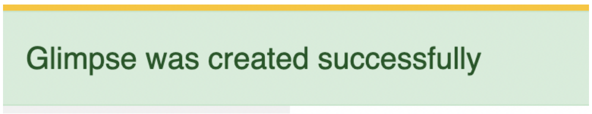
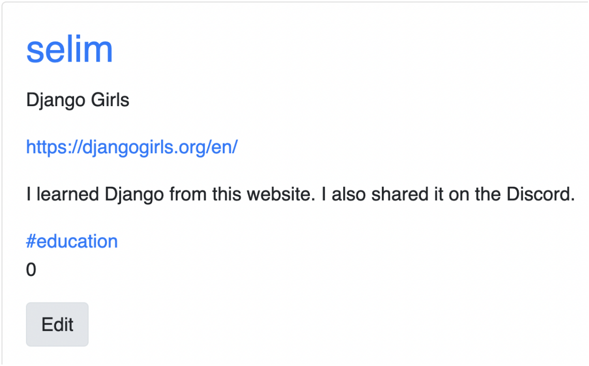

Glimpse was created successfully

Figure 6.5. Success Message

After saving the glimpse a success message would occur on the top of the screen.[Figure 6.5]

Each glimpse in the feed shows the author, title, url, description, tag, like count and if it is the user's, an edit button.[Figure 6.6] If the user clicks that edit button, they will be forwarded to a form page same as the create page except all the fields are filled with the pre-existing data. User can make changes and save or delete or cancel the editing.[Figure 6.7]



selim

Django Girls

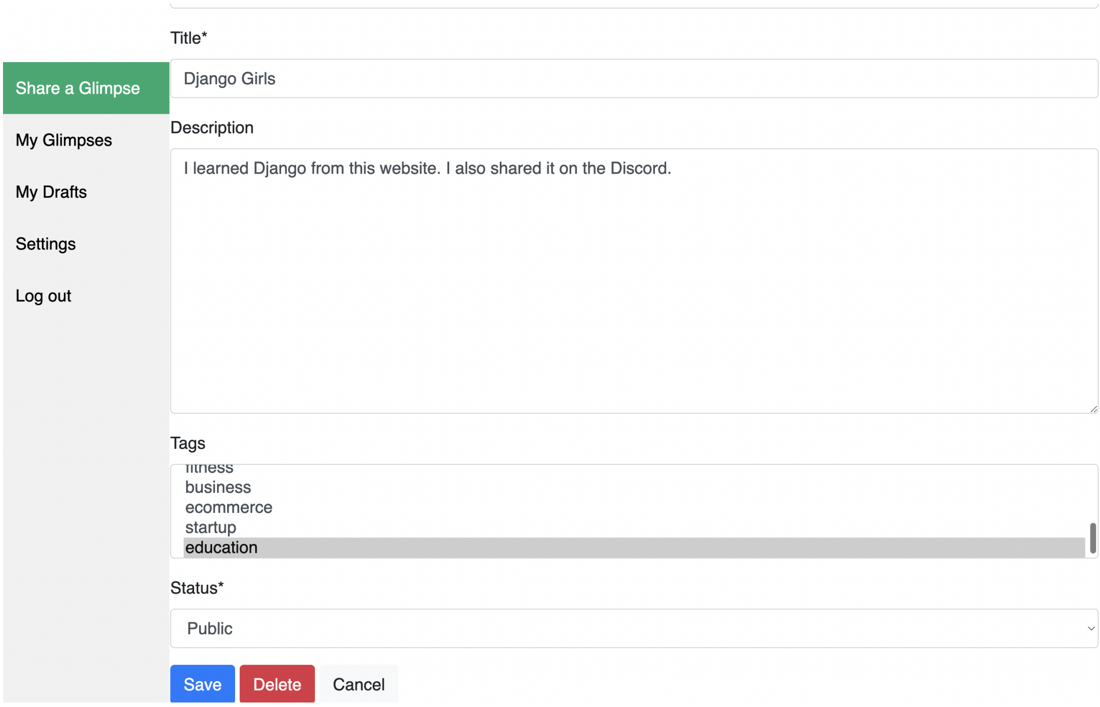
https://djangogirls.org/en/

I learned Django from this website. I also shared it on the Discord.

#education

0

Figure 6.6. glimpse Preview



Title*

Django Girls

Description

I learned Django from this website. I also shared it on the Discord.

Tags

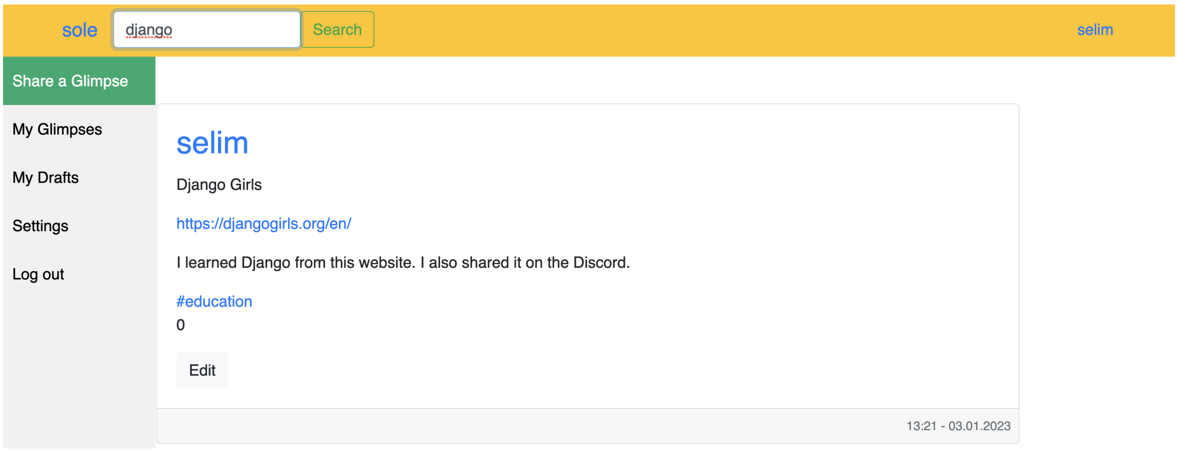
fitness
business
ecommerce
startup
education

Status*

Public

Figure 6.7. Edit

Users can also search for glimpses and users from the search bar on the top of the page.[Figure 6.8]



selim

Django Girls

https://djangogirls.org/en/

I learned Django from this website. I also shared it on the Discord.

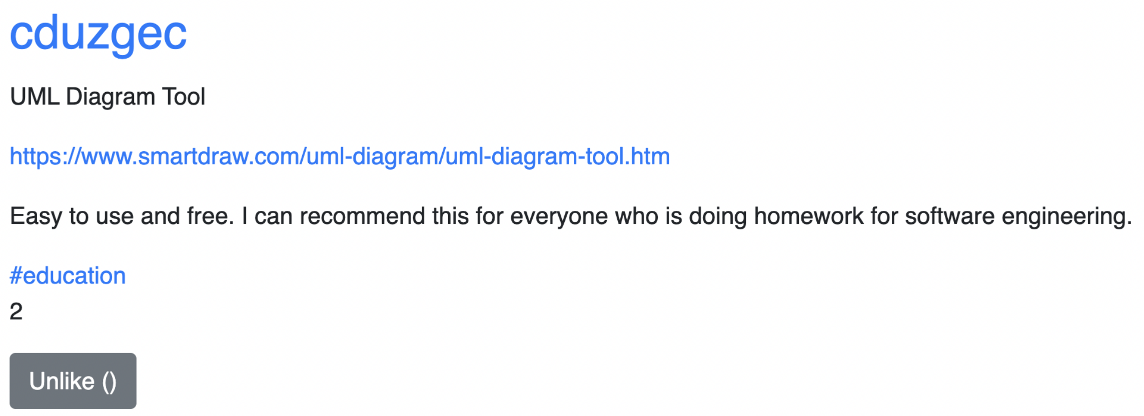
#education

0

13:21 - 03.01.2023

Figure 6.8. Search Bar

If the user is not the owner of the glimpse on the feed, they can like and unlike other user's glimpses.[Figure 6.9]



cduzgec

UML Diagram Tool

https://www.smartdraw.com/uml-diagram/uml-diagram-tool.htm

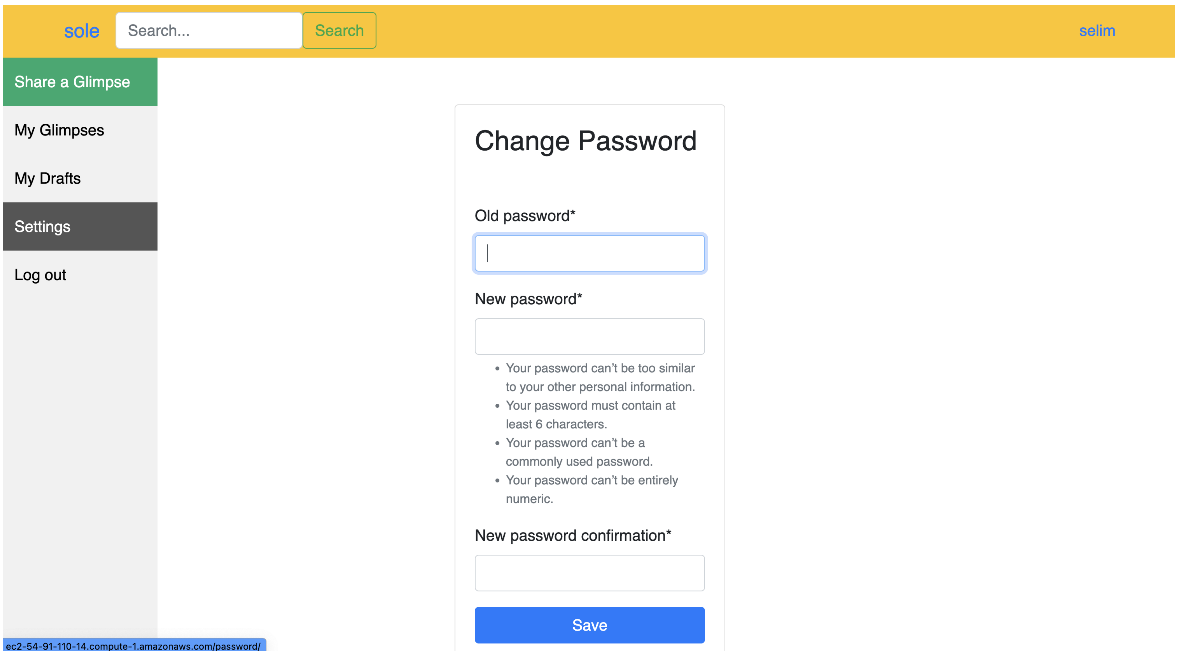
Easy to use and free. I can recommend this for everyone who is doing homework for software engineering.

#education

2

Figure 6.9. Like - Unlike

Users can also see their own glimpses and drafts from the sidebar and go to settings to change their password.[Figure 6.10] They can logout from here too.



Change Password

Old password*

New password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 6 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation*

ec2-54-91-110-14.compute-1.amazonaws.com/password/

Figure 6.10. Change Password

7. Tests

Sole app normal user can be created by sign up.

Superuser for deployed project:

username: uskudarli

password: Test573

If the tests wanted to be run locally, after running the Docker:

python manage.py test

7.1. User Tests

There are 4 unit test cases for the users:

1- Creating a new user and returning correct values

2- Leaving fields empty

3- Testing username restraints

4- Testing uniqueness of the email and username

7.2. Glimpse Tests

Glimpse tests are done through the website interface since both front end and back end needs to be tested in order to get accurate results. The test cases are:

1- Creating glimpse without logging in

2- Creating glimpse after logging in

3- Editing glimpse without logging in

4- Editing glimpse after logging in

5- Deleting glimpse without logging in

6- Editing glimpse after logging in

7- Liking a glimpse

8- Unliking a glimpse

9- Searching a glimpse

8. Demo

Demo can be watched from this url:

https://drive.google.com/file/d/1krPynX5PHD3iCUu2oiGBSq9szP93PGwi/view?usp=share_link

9. Resources

https://www.youtube.com/playlist?list=PLEsfXFp6DpzRMby_cSoWTFw8zaMdTEXgL

https://docs.djangoproject.com/en/4.1/topics/testing/overview/

https://www.youtube.com/watch?v=IHTP8-KskcQlist=PLOLrQ9Pn6cax6x4UhfCGb24JqZ

https://www.youtube.com/watch?v=xSUm6iMtREAt=13989s

https://www.youtube.com/watch?v=PtQiiknWUcI