Group 4 report for Assignment 2 – IDG2001 Dockerised website

Planning and development

This is a report for Assignment 2 – Dockerised website. The goal was to create a reddit-like website.

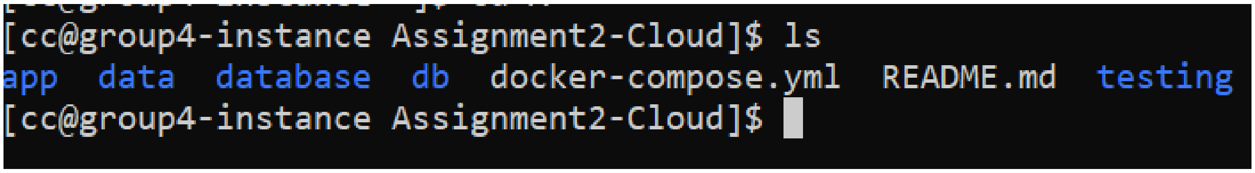
We chose to use flask for our API because it was what we used in the first assignment. Flask is a microframework used for building web applications in Python. (1) Chameleon cloud VM was chosen as the environment to deploy our application in.

We decided to give the user the ability to delete their account without deleting all their posts, as reddit does not remove the posts of deleted accounts, but simply shows their username as [Deleted].

We also used extensions for flask such as SQLAlchemy (2), which is a Python SQL toolkit that adds SQL flexibility. Pymysql (3) is a Python client library that is used to connecting to MySQL databases. Flask-Bcrypt (4) is used to hash passwords. Flask-Login (5) is used for user session management in Flask applications. UUID module (6) is used to generate universally unique identifiers (UUIDs) for deleted accounts.

VM-setup, docker and Docker-Compose

We created a VM instance on Chameleon Cloud, and set it up with ssh key, volume, other config. We setup a floating ip in which we could use to ssh into the vm. We used BitWise for this process and get access to the linux terminal. Here we installed the necessary stuff like docker and mariadb, which all functioned fine. We then added our github repository, so we could run the docker compose and dockerfile to build images.

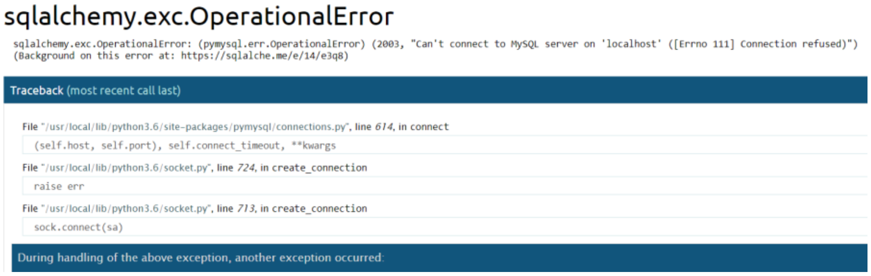


```
[cc@group4-instance Assignment2-Cloud]$ ls
app  data  database  db  docker-compose.yml  README.md  testing
[cc@group4-instance Assignment2-Cloud]$ 
```

We have successfully been able to build the flask app using docker compose and running “sudo docker compose up –build " in repo dir. Buildng and running the container on the Centos7 VM is working.

When it comes to the sql database image, we were able to build it to an extent without errors. But ultimately we had issues with connecting the flask app to the database container using sqlalchemy, getting errors like “connection refused” for instance. We decided after a lot of effort and configurations, to stop. At least the flask app container runs the way it was intended.

Connection error on vm ip -- 129.114.27.249:5000/register :

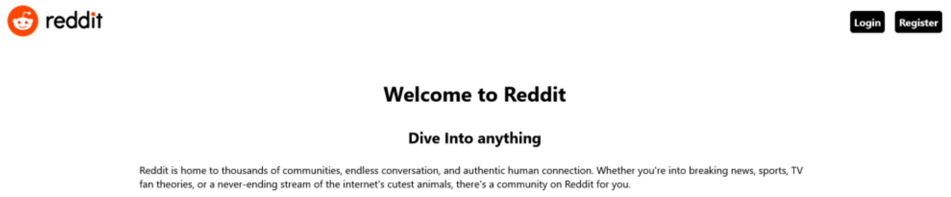


```
sqlalchemy.exc.OperationalError

sqlalchemy.exc.OperationalError: (pymysql.err.OperationalError) (2003, "Can't connect to MySQL server on 'localhost' ([Errno 111] Connection refused)")
(Background on this error at: https://sqlalche.me/e/14/e3q8)

Traceback (most recent call last)

  File "/usr/local/lib/python3.6/site-packages/pymysql/connections.py", line 614, in connect
    (self.host, self.port), self.connect_timeout, **kwargs
  File "/usr/local/lib/python3.6/socket.py", line 724, in create_connection
    raise err
  File "/usr/local/lib/python3.6/socket.py", line 713, in create_connection
    sock.connect(sa)

During handling of the above exception, another exception occurred:
```

API system functionality and structure

Registering an account

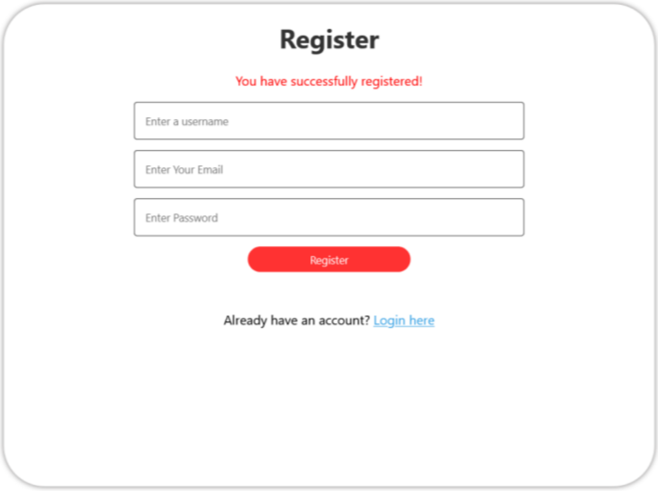
On index.html, a user is met with the option to log in or register. They register by submitting a username, email and password. If they try to register with a username / email that is already in use in the database, they will be informed and be unable to register. Once a user registers successfully, they get assigned a user id and their user shows up in the database. The database stores information like user ID, username, email, password and account creation date. The password is hashed with Bcrypt and the hash is stored in the database for security reasons. During login the submitted password is hashed and compared to the hashed password in the database.



Welcome to Reddit

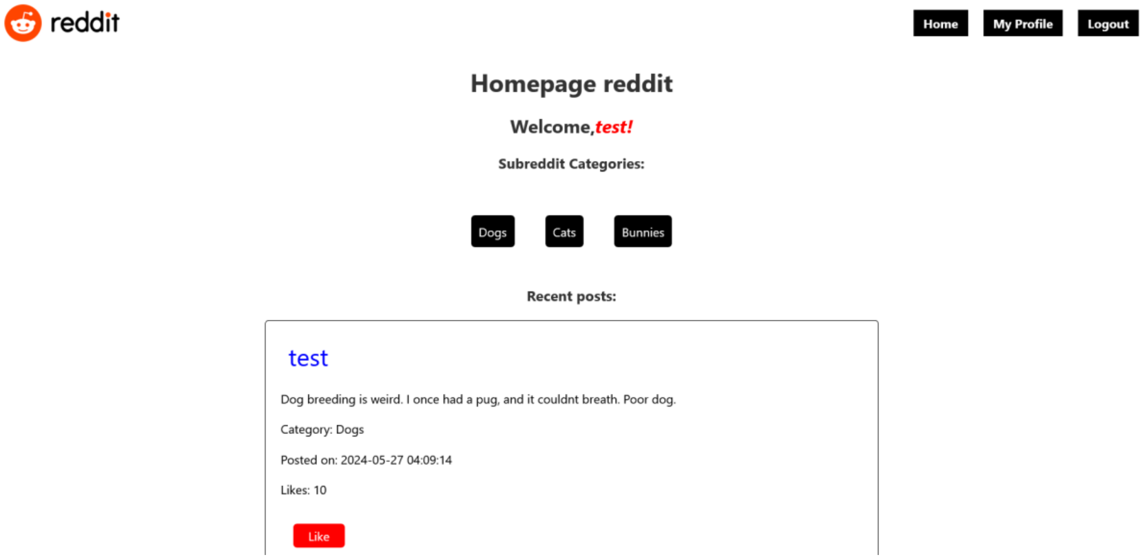
Dive Into anything

Reddit is home to thousands of communities, endless conversation, and authentic human connection. Whether you're into breaking news, sports, TV fan theories, or a never-ending stream of the internet's cutest animals, there's a community on Reddit for you.



You have successfully registered!

Already have an account? Login here



Homepage reddit

Welcome,*test!*

Subreddit Categories:

Recent posts:

test

Dog breeding is weird. I once had a pug, and it couldnt breath. Poor dog.

Category: Dogs

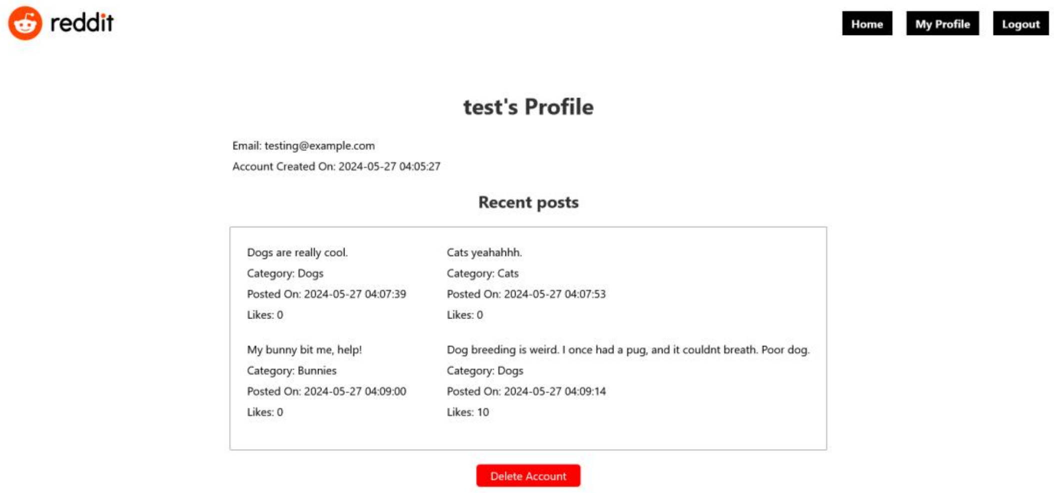
Posted on: 2024-05-27 04:09:14

Likes: 10

Logging in and using the website

When a user tries to log in, the app checks if the username and the hashed password matches the one saved in the database. If both match, the user will be successfully logged in, and redirected to home.html. If not, they will receive an error message.

Once logged in, the user will be able to view, like and create posts in the categories Dogs, Cats and Bunnies. Home and category feeds show the 10 most recent posts. They can also look at the user profiles of the post authors, which show their recent posts, username and email. On their own user profile, they can view when their recent posts, when account was created, username, email, and they have the option to delete their own account. Deleting their account will not delete their posts. Once they are done using the website they can log out and be redirected back to the login page.



test's Profile

Email: testing@example.com

Account Created On: 2024-05-27 04:05:27

Recent posts

Dogs are really cool.
Category: Dogs
Posted On: 2024-05-27 04:07:39
Likes: 0

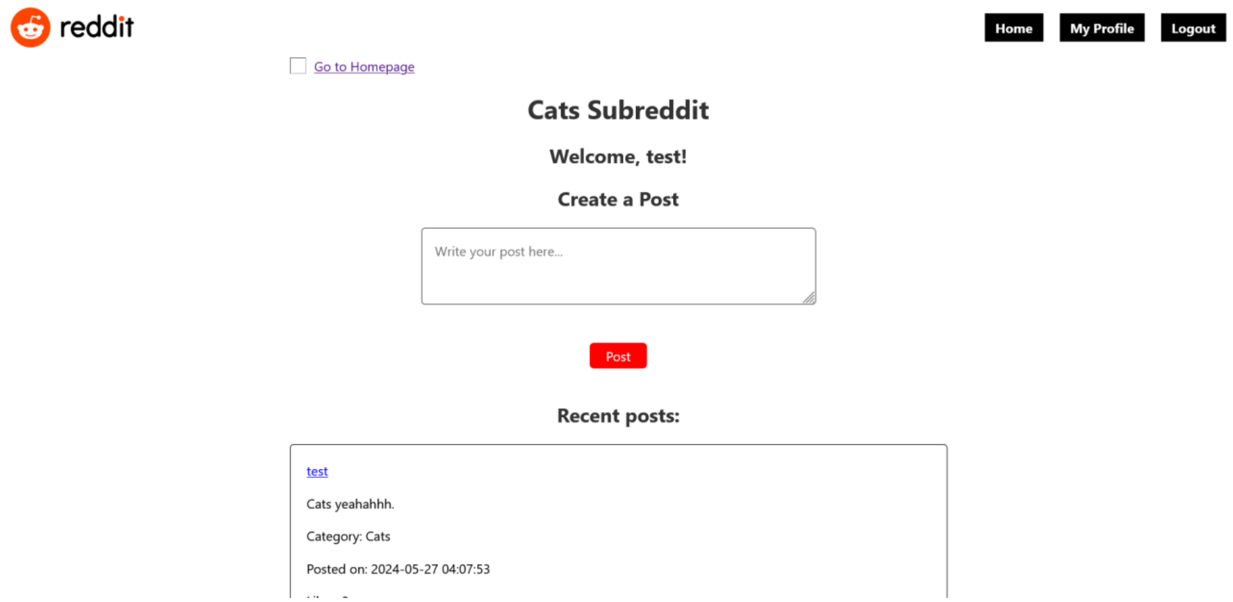
Cats yeahahhh.
Category: Cats
Posted On: 2024-05-27 04:07:53
Likes: 0

My bunny bit me, help!
Category: Bunnies
Posted On: 2024-05-27 04:09:00
Likes: 0

Dog breeding is weird. I once had a pug, and it couldnt breath. Poor dog.
Category: Dogs
Posted On: 2024-05-27 04:09:14
Likes: 10

Creating a post

When a post is created, information such as ID, user ID, username, text content, category, time of creation and like count will be added to the "Post" table in the database. While on a normal website, you would usually be able to delete your own posts, this is not possible on our website. You can delete your account, but your posts will persist regardless. Length of post is limited to 255 characters.



Cats Subreddit

Welcome, test!

Create a Post

Recent posts:

test

Cats yeahahhh.

Category: Cats

Posted on: 2024-05-27 04:07:53

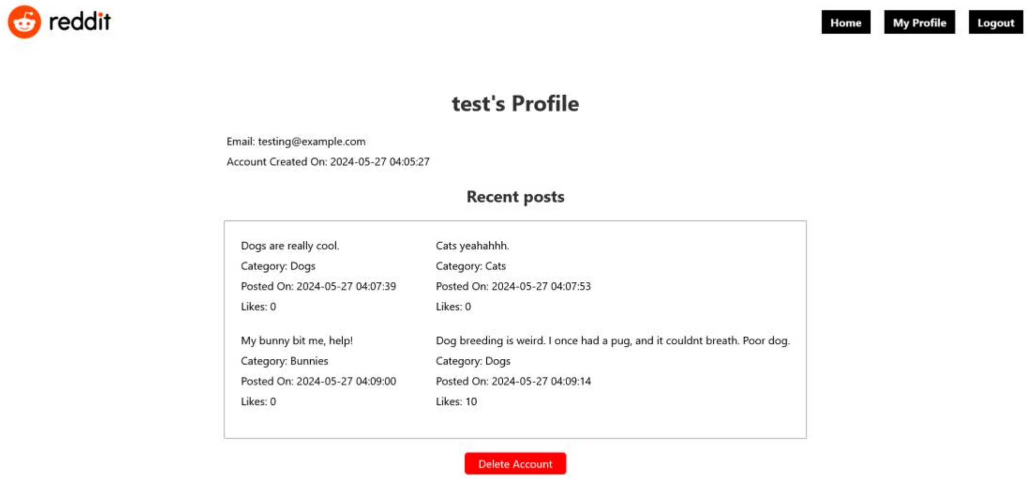
Liking posts and like batcher

We chose to track likes as a column called like count on the "Post" table. While ordinarily a reddit-like website will limit a user to liking or unliking a post once, our users can like posts as many times as they want. Tracking which user liked which posts would require an extra table in the database to track that information. As that was not required for this assignment, we chose to simply track like count on posts.

Unlimited likes, however, means that the database would be spammed every time the like buttons on posts are clicked. To prevent the spam, we have a like batcher. Like_batcher.py will update the database when the size of the batch exceeds at least 10 likes. This means that when batched likes reach 10, they're sent to the database. The like_count column on posts is therefore increased in batches of 10.

Deleting an account

It was unclear whether the part about a delete button in the assignment meant that the user should be able to delete each of their own posts, or simply delete their account. We chose to set up the option for a user to delete their account while keeping their posts intact. A logged in user can delete their own account on their user profile. A prompt asks the user to confirm their choice, as it cannot be undone. Once the account is deleted, the user is redirected to index.html. Their account gets assigned a randomly generated UUID (universally unique identifier) for username and email and password is removed. By assigning a generated username and email in the database, it frees up the original username and email. Password is removed to prevent login. A user could delete their account and immediately register with the same info if they wished. Posts created by deleted accounts will have the generated username and email instead of the original credentials. This means other users can view posts even if the authors have deleted their accounts.



test's Profile

Email: testing@example.com
Account Created On: 2024-05-27 04:05:27

Recent posts

Dogs are really cool.
Category: Dogs
Posted On: 2024-05-27 04:07:39
Likes: 0

Cats yeahahhh.
Category: Cats
Posted On: 2024-05-27 04:07:53
Likes: 0

My bunny bit me, help!
Category: Bunnies
Posted On: 2024-05-27 04:09:00
Likes: 0

Dog breeding is weird. I once had a pug, and it couldnt breath. Poor dog.
Category: Dogs
Posted On: 2024-05-27 04:09:14
Likes: 10

Basic modelling of system and database

The database model for the application creates two primary tables: user and post. User contains columns like user ID, username, email, password (hashed), and account creation date. This is is where all the user information is stored after successfully registering. Password is stored as a hashed string for security. Each user is given an unique ID that serves as the primary key. The username and email fields and cannot be null, which ensures each user is distinct. When a user is deleted, they stay in the database but are given universally unique identifiers for both username and email. Password is removed after user deletion.

Post contains columns such as post ID, user ID, text content, category, creation date and like count. Post is therefore connected to user to ensure relational integrity. The user ID is a foreign key that links each post to a user, which enforces relational integrity. The text field contains the actual post content. Category shows which subreddit category the post belongs to. Created_at field is the timestamp of when the post was created. Like count stores the amount of likes.

Initial user and post data is then inserted into their respective tables.

How to build such a system

To build such a system you need to first decide system architecture and what you want it to do, and how to connect the main components. Then you need to select what tools to use, and consider how to best integrate them. For example, flask for the application, Redis for caching, and Apache HTTP Server as a load balancer. Modules and libraries included.

Set up docker environment. Install docker, create dockerfiles, create compose file, build images/containers.

Testing

In the testing directory we have testing_requirements.txt, which can be used to pip install the required tools for testing, for example Pytest. Mypy.ini is the config file for mypy. .flake8 is the config file for flake8.

Front-end design

For the styling we used CSS, html and JavaScript. The styling is consistent throughout the page. The front-end design draws direct inspiration from reddit.

Technologies that are used:

HTML/CSS, Javascript
Flask, Python

Flask-Bcrypt

Flask-login

SqlAlchemy

Sources:

1. Flask: https://en.wikipedia.org/wiki/Flask_(web_framework)
   Sqlachemy: https://docs.sqlalchemy.org/en/20/intro.html#documentation-overview
2. Pymysql: https://pypi.org/project/PyMySQL/

3. Flask-Bcrypt: https://flask-bcrypt.readthedocs.io/en/1.0.1/
4. Flask-Login: https://flask-login.readthedocs.io/en/latest/
5. UUID Module: https://www.uuidgenerator.net/dev-corner/python