TROCBENIN

BIG DATA

AUTHORS

GNOGUEM F. C. NESTOR | HOUNGE ISAAC | SAIBOU AZIZ

INSTRUCTORS: CARLYNA BONDIOMBOUY|PATRICK VALDURIEZ

Introduction

This project is about creating a platform for exchange of objects in Benin. On our list of functional requirements, we had two users. A client and an administrator. We identified the following functional requirements as those of the client:

Client functional requirement

- A client should be able to register (registration involves paying a registration fee through an online payment gateway). We did not do paid registrations for now but we will do it in the future
- A client should be able to login
- A client should be able to search for objects
- A client should be able to view the details of an object
- A client should be able to propose an item for the exchange (only logged in clients)
- A client should be able to delete his/her objects
- A client should be able to exchange messages with another client if they have an exchange going on

The following functional requirements were those of an administrator

- The administrator should be able to delete clients and objects from the system
- The administrator should be able to active a user account
- An administrator should be able to see site statistics

Architecture of platform

Flask and MongoDB we used through python to realize this project.

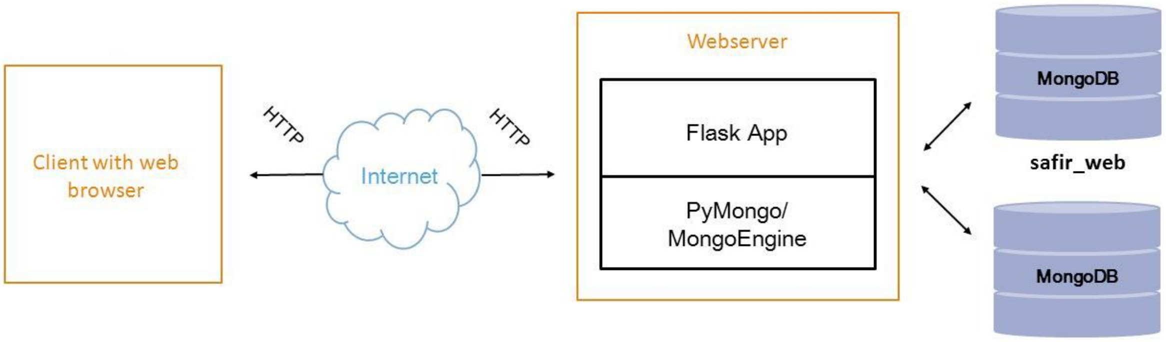

Figure 1 Architecture of Platform

Flask is responsible for handling user request. It keeps what is represented to the user in the template folder and all resources like images and scripts are kept in the static folder. The model has been implemented through the controller folder and through the database folder. The starting point of the application is __init__.py. This file handles the various routes through the platform. There are three main folders in our application.

The template folder

This folder contains the list of all our views or templates

The database folder

This folder contains a file named queries.py which contains all the queries that were made to database. The file test.py is used to export the database to a file named colletion.json The migration.py file is used to create a database with the dataset in the collection.json file

The controller folder

This folder is a helper folder and its purpose is to assist the routes with their task of managing requests.

We have not been able to implement all the function requirements. It has been quite challenging meeting these requirements and we have not been able to work on the administrator's panel. In addition, clients are not yet able to exchange messages. The site links are not working. Below are the links for various actions:

The homepage is the site route

- The login page is at: /login
- The registration page is at: /register
- The dashboard is at: /dashboard
- The full list of objects is at: /trocs
- The contact page is at /contacter
- To add an object to be exchanged, the link is at: /troquer . You must be logged in to access this link
- To log out of the system the link is /logout
- This route leads you to a particular object but it is called with one parameter; the image name of the object

  ```
  # This is the route that leads you to a particular object
  @app.route('/trocs/<trocname>')
  def objet(trocname):
      return obj.objet(trocname)
  ```

- When you have searched a found and object you want to do an echange with, this route is called @app.route('/proposertroc/<trocname>') it takes into parameter the image image of the object you have found.

Conclusion

This project has permitted us to familiarize ourselves with some to the tools used to handle big data. We have been able to learn how to work with MongoDb and Flask. It is rather unfortunate that time has caught us short but we are still working on this project and we hope that by September, we will be launching the project.

We wish to thank you for all the time you have given us. You gave us much time. The truth is the project is a robust one.

Thank you!