ECE1779 Assignment 1 Web Development

Zichuan Wang 1000474300

Zhaohui Qu 1005783127

Huan Qi 1006437214

1 Project Summary

The website we developed allows users to register with username and password, and then login with the same user credentials. Once logged in, users can upload photos on which openCV is automatically applied for object detection. Rectangles will be drawn around detected objects along with their corresponding category labels. Thumbnails of uploaded photos are displayed within the gallery. By clicking on the thumbnail, original image as well as the the modified image with detected objects will be presented. The block diagram outlining the application flow is shown in figure 1.

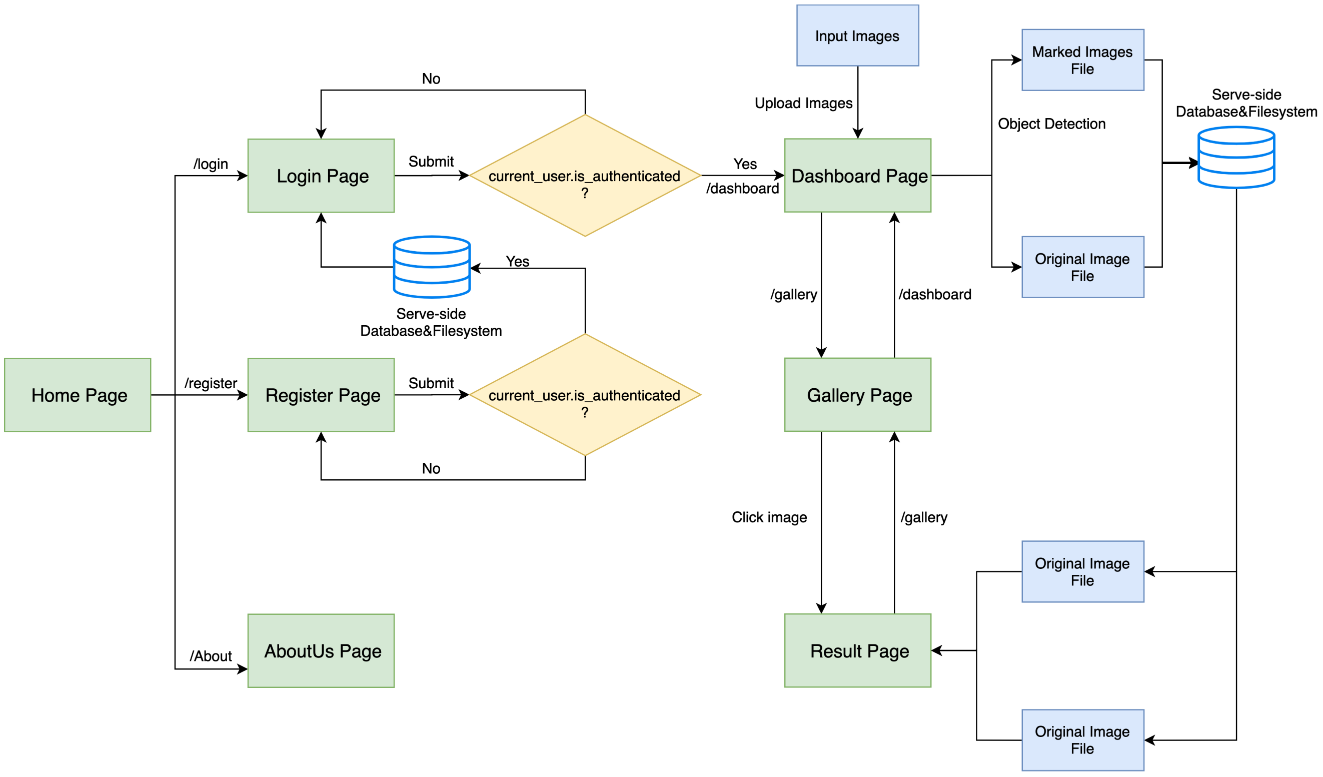


Figure 1: Block Diagram

2 Features

1. User Registration

2. User Login
3. Photo Uploading
4. Photo Object Detection
5. Photo Thumbnail Display and Onclick Enlargement

3 Package Requirement for Virtual Environment

1. flask
2. flask-login
3. flask-wtf
4. wheel
5. flask-sqlalchemy
6. mysql-connector-python
7. mysqlclient
8. opencv-python-headless
9. validators
10. passlib
11. gunicorn

4 How to Run

Assuming MySQL server is connected, run bash scripted called `start.sh` from desktop to use `gunicorn` to start the flask web application. Once the web application is initialized, it can be access on port 5000 from outside the instance through url `<EC2_instance_IP:5000/>`.

5 User Instruction

1. For new users, go to `<EC2_instance_IP:5000/>` which opens the welcome page, select **Register** from the navigation bar at the top to create new credential with username and password. For returning users, select **Login** to use existing credential to log in.
2. Once logged in, users will be redirected to dashboard page. Users can select images to upload by clicking **Choose Files** button. Selected images can be uploaded by clicking **Upload** button.

3. By clicking the **Gallery** button on the top navigation bar, users will be redirected to gallery tap with thumbnail views of all uploaded images. By clicking the thumbnail, original image and its corresponding modified image with object detected will be presented to users in a new browser tab.

4. Users can logout at any time by clicking the **Logout** button on the top navigation bar.

6 Developer's Documetation

6.1 Front End

1. **Templates**: This folder contains all HTML templates referenced by each routes. `layout.html` defines the base layout structure of all other pages. Before login, `home.html`, `login.html`, `register.html`, `about.html` defines the layout of welcome, login, register and aboutus page respectively. Once logged in, `gallery.html`, `dashboard.html`, `about.html`, `showimg.html` defines the layout of gallery, dashboard, aboutus and separate show image pages respectively.

6.2 Back End

1. **__init__.py**: Initialization module that runs every time the web application is deployed. It instantiates a Flask instance, establishes SQL connection, creates client side session key for secure connection.

2. **routes.py**: Handles all route requests except register from the website, a detailed description of each route is presented as follow:

   a. **home**: If the user is not logged in, stay at `home.html` and display welcome message. If the user is logged in, redirect to `dashboard.html`.

   b. **login**: If a match is found for the user credential entered, user will be allowed to login. If the password doesn't match the record or if the username doesn't exist inside database, access will be denied with error message shown on the page.

   c. **logout**: After the logout button is clicked, user will be logged out and redirected to `login.html`

   d. **dashboard**: User can upload images for object detection. Uploading without image, uploading image without valid format and uploading image exceeding size limit will all be denied with error message shown on the page.

   e. **gallery**: All thumbnails of uploaded images will be shown in the gallery. Clicking the thumbnail will redirect the browser to `showimg.html` with original image and marked image.

   f. **api/upload**: Pre-set load generator for image uploading for TA marking purpose.

3. **register.py**: Handles register route requests from website.

   a. **register**: If the username doesn't exist inside database, it can be used to register. Upon successful registration, username and SHA256 hash of the corresponding password would be saved onto database.

   b. **api/register**: Pre-set register API for TA marking purpose.

4. **form.py**: This module defines the format and constraints of the LoginForm and RegisterationForm. The constraints are listed as follow:

   a. **LoginForm**: Username and password must be present to login. Number of character a valid username can be consist of is between 1 and 25. Number of character a valid password can be consist of is between 6 and 255.

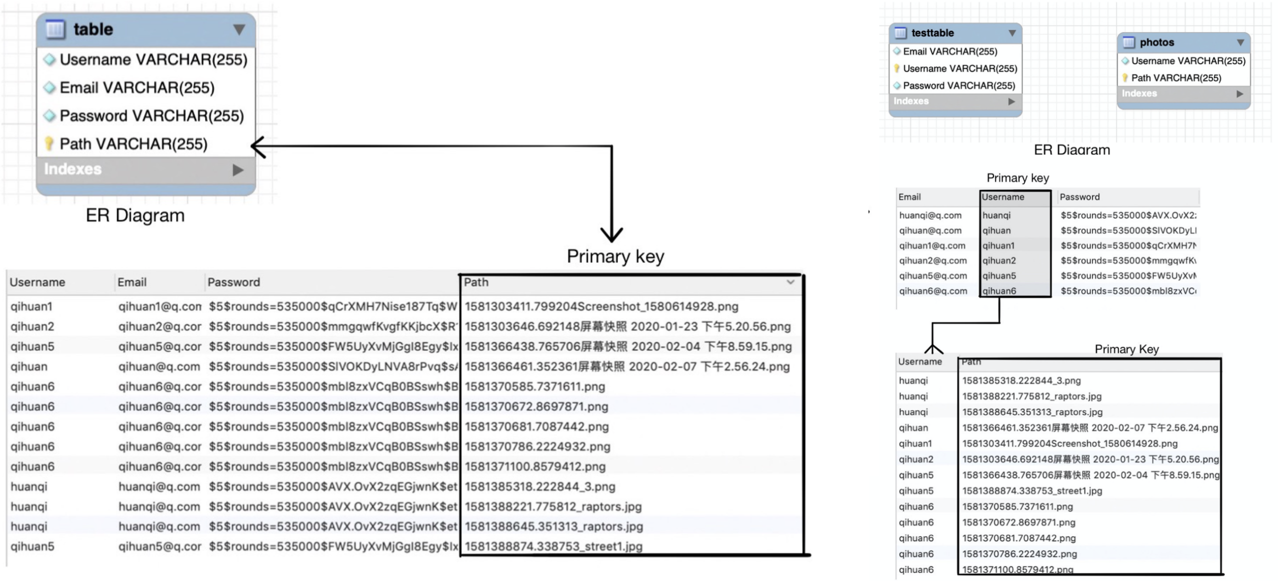
   b. **RegisterationForm**: Username, email and a pair of matching password input must be present to register. Number of character a valid username can be consist of is between 1 and 25. Number of character a valid email can be consist of is between 6 and 50. Number of character a valid password can be consist of is between 6 and 255. Duplicates of username and email entities are not allowed.

5. **object_detection.py**: This module performs the object detection on the uploaded image using openCV. Labelled rectangles are drawn around the objects detected in the image. Original images and marked images are stored in `website/images` and `website/output` folders respectively with the same filename. Filename is the current time stamp.

6. **Photo.py, User.py**: Custom defined classes for easy database manipulation.

7 Database Schema

Database schema before and after normalization are shown in figure 2a and figure 2b respectively. Before normalization, user information is duplicated for each image under that specific user. After normalization, unique user id is used to represent each user in order to identify the owner of each image.



Username	Email	Password	Path
qihuan1	qihuan1@q.con	5rounds=535000$qCrXMH7Nise187Tq$W	1581303411.799204Screenshot_1580614928.png
qihuan2	qihuan2@q.cor	5rounds=535000$mmgqwfKvgfKKjbcX$R	1581303646.692148屏幕快照 2020-01-23 下午5.20.56.png
qihuan5	qihuan5@q.cor	5rounds=535000$FW5UyXvMjGgl8Egy$Ix	1581366438.765706屏幕快照 2020-02-04 下午8.59.15.png
qihuan	qihuan@q.com	5rounds=535000$SlVOKDyLNVA8rPvq$s	1581366461.352361屏幕快照 2020-02-07 下午2.56.24.png
qihuan6	qihuan6@q.cor	5rounds=535000$mbl8zxVCqB0BSswh$B	1581370585.7371611.png
qihuan6	qihuan6@q.cor	5rounds=535000$mbl8zxVCqB0BSswh$B	1581370672.8697871.png
qihuan6	qihuan6@q.cor	5rounds=535000$mbl8zxVCqB0BSswh$B	1581370681.7087442.png
qihuan6	qihuan6@q.cor	5rounds=535000$mbl8zxVCqB0BSswh$B	1581370786.2224932.png
qihuan6	qihuan6@q.cor	5rounds=535000$mbl8zxVCqB0BSswh$B	1581371100.8579412.png
huanqi	huanqi@q.com	5rounds=535000$AVX.OvX2zqEGjwnK$et	1581385318.222844_3.png
huanqi	huanqi@q.com	5rounds=535000$AVX.OvX2zqEGjwnK$et	1581388221.775812_raptors.jpg
huanqi	huanqi@q.com	5rounds=535000$AVX.OvX2zqEGjwnK$et	1581388645.351313_raptors.jpg
qihuan5	qihuan5@q.cor	5rounds=535000$FW5UyXvMjGgl8Egy$Ix	1581388874.338753_street1.jpg

(a) Before Normalization

Email	Username	Password
huanqi@q.com	huanqi	5rounds=535000$AVX.OvX2
qihuan@q.com	qihuan	5rounds=535000$SlVOKDyLI
qihuan1@q.com	qihuan1	5rounds=535000$qCrXMH7N
qihuan2@q.com	qihuan2	5rounds=535000$mmgqwfK
qihuan5@q.com	qihuan5	5rounds=535000$FW5UyXvM
qihuan6@q.com	qihuan6	5rounds=535000$mbl8zxVC

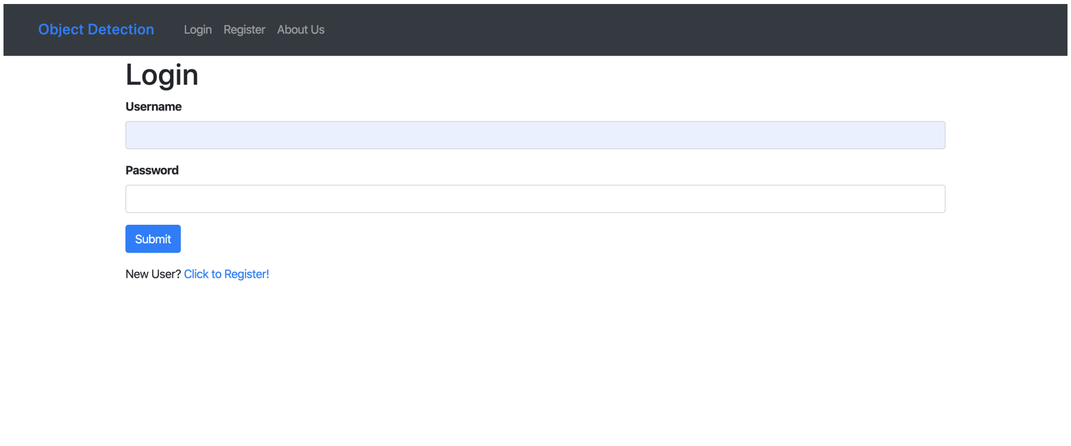
Username	Path
huanqi	1581385318.222844_3.png
huanqi	1581388221.775812_raptors.jpg
huanqi	1581388645.351313_raptors.jpg
qihuan	1581366461.352361屏幕快照 2020-02-07 下午2.56.24.png
qihuan1	1581303411.799204Screenshot_1580614928.png
qihuan2	1581303646.692148屏幕快照 2020-01-23 下午5.20.56.png
qihuan5	1581366438.765706屏幕快照 2020-02-04 下午8.59.15.png
qihuan5	1581388874.338753_street1.jpg
qihuan6	1581370585.7371611.png
qihuan6	1581370672.8697871.png
qihuan6	1581370681.7087442.png
qihuan6	1581370786.2224932.png
qihuan6	1581371100.8579412.png

(b) After Normalization

Figure 2: Database Schema

8 User Interface Screen Shot

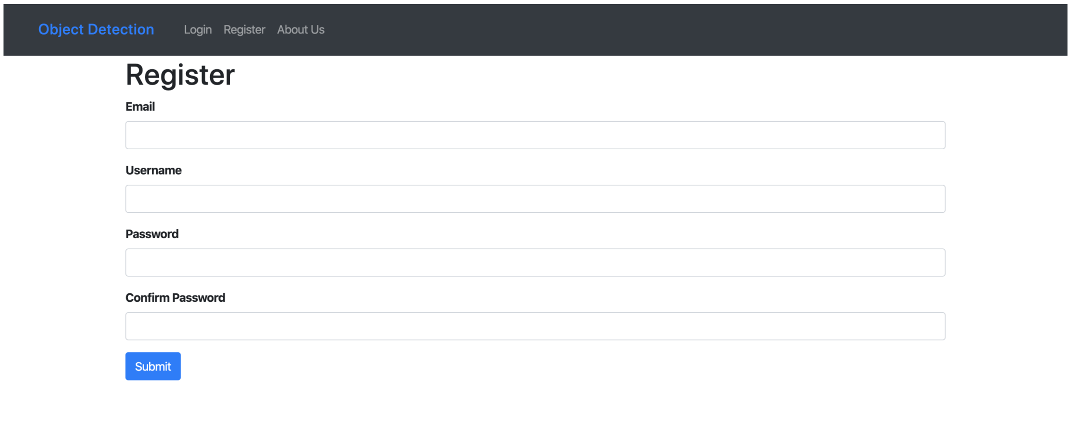
Screen shots of all pages from the website application are presented as follow:



Login

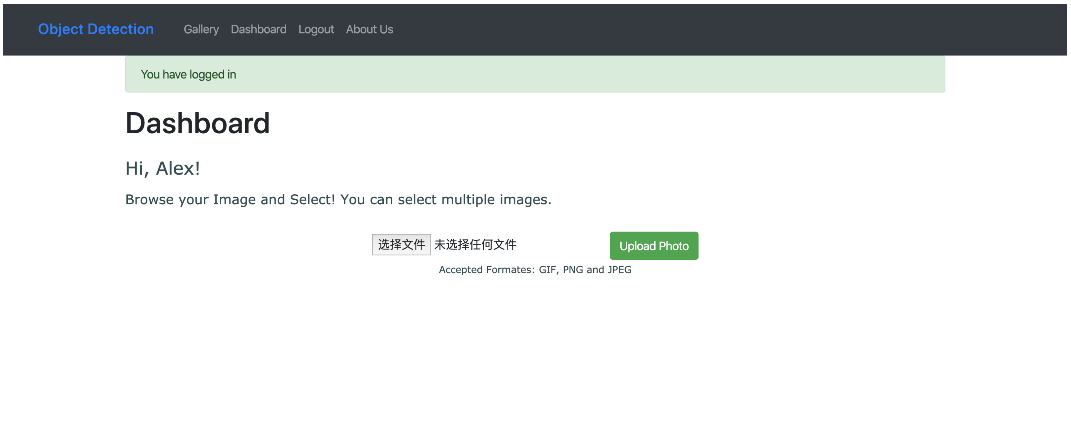
New User? Click to Register!

Figure 3: Login



Register

Figure 4: Register



You have logged in

Dashboard

Hi, Alex!

Browse your Image and Select! You can select multiple images.

Accepted Formates: GIF, PNG and JPEG

Figure 5: Dash Board

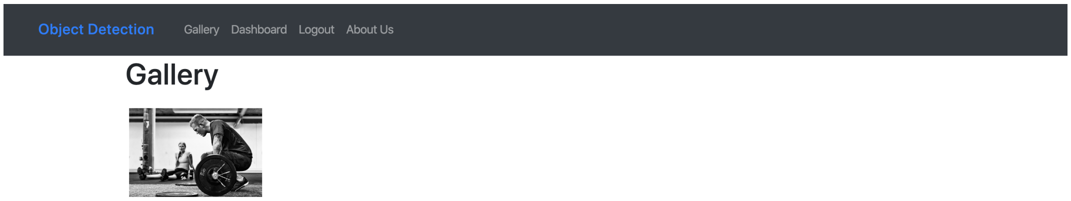


Figure 6: Gallery

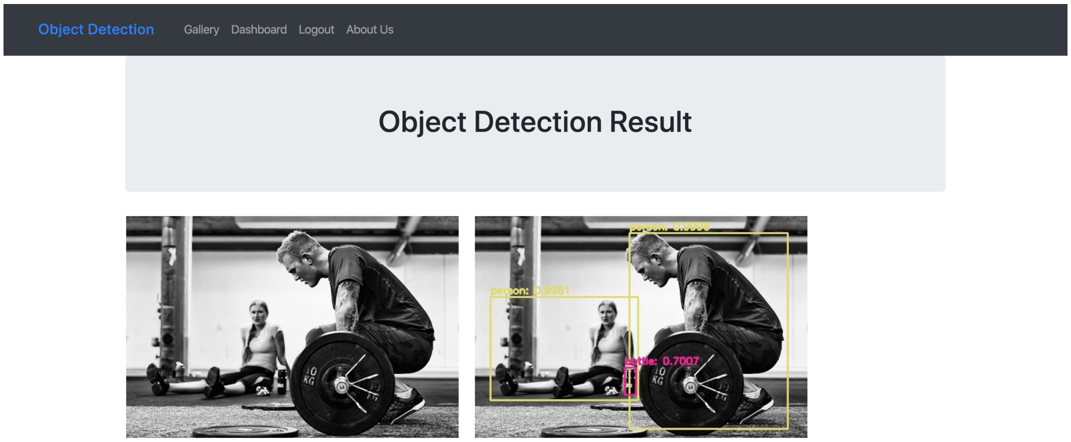


Figure 7: Show image