



## Facial Recognition Project

Arnas Steponavicius      Aaron Moran      Thomas Kenny

April 20, 2020

### 1 Introduction

When we formed our team there was many ideas floating around of what to design and make. We tried to think of an idea that was new and exciting for the three of us. We wanted to make design and make something that was challenging and would improve our ability as software students. The idea we settled on was Facial Recognition.

With extensive research into this field we looked at many languages to use. We thought it would be a good idea to learn a new language while working on this project. This language was Python. Python is one of the top languages used worldwide at the moment so we thought this would be beneficial for us to learn this language. We chose this language for various reasons, one of these reasons being that there is various packages python offers. We found online that there is a library called **face\_recognition** that suited our needs for this project.

We had various meetings in the beginning to find the pros and cons of Facial Recognition. We had to map out the project and treat this like it was a job. This included making a plan of what we wanted to have done to present in our weekly meetings, breaking up the project into steps so we wouldn't get overwhelmed or held up on one task and discussing what each member of the team would focus on. Each task will be explained in further detail.

GitHub was vital with our project. It was important that we constantly used GitHub to display the progress we were making. We could handle issues we were having, test different ideas and we could all be on the same page as we work.

## 2 System Requirements

- Python 3.x
- C/C++ compiler installed
- CMake <https://cmake.org/download/>
- Camera

## 3 Technology Used

### Language:

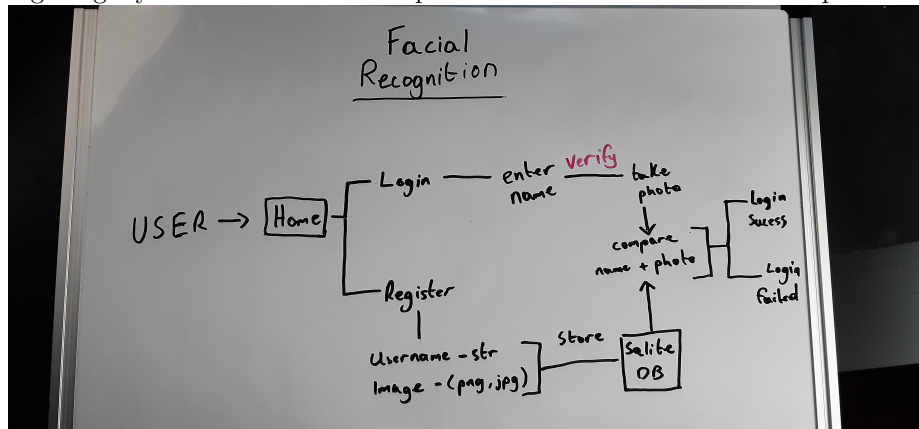
- Python

### Dependencies:

- pip install face\_recognition
- pip install opencv-python
- pip install numpy
- pip install pillow

## 4 Architecture of the solution

We designed this project to be user friendly and simplistic. From the following images you can see the UI is simple which makes for a better user experience.



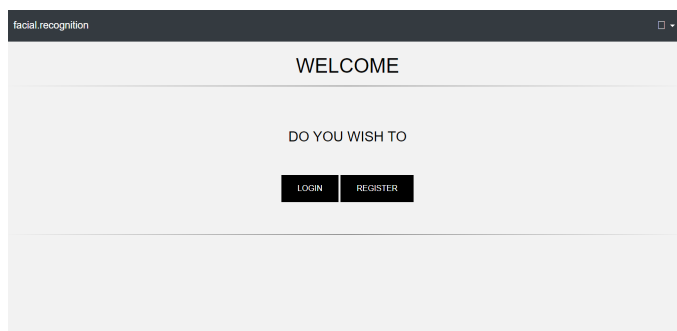
The thought we had in mind for this project was for companies to use our Register and Login software and enhance it to suit their needs. This software can be easily tailored from colour to design with customer request. The UI design is located on one CSS file. This makes for updating style simple and quick to achieve. The python and html files are formatted for ease of navigation.

The order of the files are simple and understandable. This makes for an easier approach to edit specific files.

## 5 Design Methodology

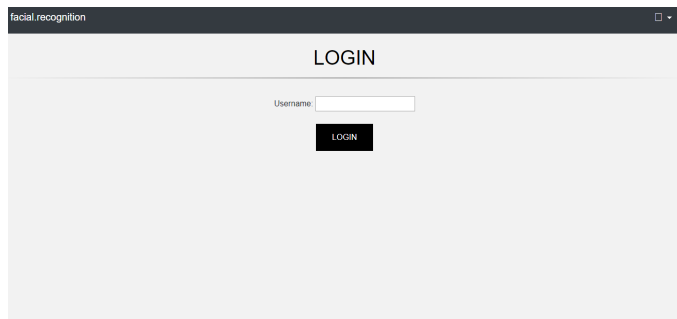
After trial and error this is the design we settled on. As you can see the design is very simple and clear. We thought about adding more to the pages like a template website but decided on "a less is more" approach to the design.

### Home Screen:



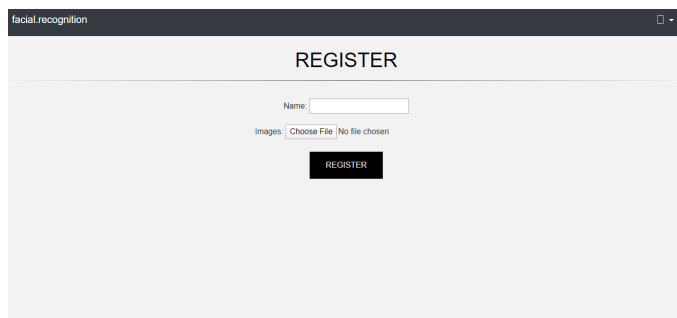
A mockup of a web browser window titled "facial\_recognition". The page has a light gray background. At the top, the word "WELCOME" is centered. Below it, the text "DO YOU WISH TO" is centered. Underneath, there are two black buttons with white text: "LOGIN" and "REGISTER".

### Login Screen:



A mockup of a web browser window titled "facial\_recognition". The page has a light gray background. At the top, the word "LOGIN" is centered. Below it, the text "Username:" is followed by a white input field. Underneath the input field is a black button with white text: "LOGIN".

### Register Screen:



A mockup of a web browser window titled "facial\_recognition". The page has a light gray background. At the top, the word "REGISTER" is centered. Below it, the text "Name:" is followed by a white input field. Underneath the input field, the text "Images:" is followed by a file selection interface showing "Choose File" and "No file chosen". Below this is a black button with white text: "REGISTER".

## 6 Features

The main aim of the features implemented in this project is to provide security by using facial recognition to be able to access the inner system. Whether this software will be used for a personal use or within a company environment.

- The first feature is the register page which needs the user needs to submit their name and a clear frontal picture of themselves. These details are then saved to the database.
- The main feature is the login. The user needs to enter a valid name that has been registered otherwise an error will pop up. If the name is valid the user is then brought to the verification page, where they need to take a photo of themselves. If the photo matches the image the users registered image they are let through into the site.

## 7 Limitations and Bugs

As with any programming project we encountered a plethora of bugs and limitations whilst working on this project. The most frustrating one of all was in the initial stages of the project. We had chosen to work with the **Flask** framework and had a problem with taking an image on the web page using javascript, encoding the image and sending it to our database via Ajax. This problem that we had set us back for a while until we ultimately decided to swap framework to **Django**.

Within a few days of making the switch we managed to get our images uploading to a database and then setting up the routes for the other links in our web application.

Another bug we had encountered was an Index out range error whenever an image that did not have a recognisable face on it was loaded in, but it was tackled with simple try catch statement.

## 8 Testing Plans

We went through extensive testing whilst making this project. Every commit that was uploaded to our repository was tested beforehand before the commit. Testing played a vital role in the production of our project. We had to make sure each small part was tested so the user experience would be up to standard. We found problems along the way whilst making of the project.

One of these problems we encountered was the upload of an image. Through White Box Testing we discovered and fixed the errors we had. We also use a Black Box Testing approach to see what the user experience was like on a user that wasn't apart of the production group. This gave us insight into what we need to improve on.

## 9 Recommendation for Future Development

For future development we will take on board everything we learned whilst doing this project. The team enjoyed the challenge of working with and learning a new language. This is something we can carry over to future development.

We now know even with extensive testing you can still hit many roadblocks along the way. Our experience of this was the framework. At the start we decided on using flask as our framework but we encountered multiple problems with our database and images. With this in mind we made the choice to switch to **Django**. This decision was made for the better of the project and once changed the project came together quickly. These types of problems will be beneficial for us in the future. Our learning of GitHub will be useful in the future. Through various lectures we learned how to use Issues on GitHub this made dealing with problems more manageable.

## 10 Conclusions

To conclude,

The making of this project proved to be no easy task. Through making this project each of us learnt and improved various skills, whether it be improved communication skills, working as a team, design or coding ability. After finishing the project and looking back we feel we could have done better in some areas. For example we could have researched more into certain areas like the database or framework. This however gave us challenges to overcome. We now know how to deal with these types of problems if they were to occur in the future when we are working professionally as software developers.

## 11 Acknowledgements

<https://www.fullstackpython.com/table-of-contents.html>

<https://www.fullstackpython.com/django.html>

<https://www.fullstackpython.com/flask.html>

<https://www.sqlalchemy.org/>

<https://www.pythonpool.com/face-detection-using-opencv-and-python/>

**face\_recognition library**

[https://github.com/ageitgey/face\\_recognition/issues/175](https://github.com/ageitgey/face_recognition/issues/175)

**Django File Uploading:**

<https://docs.djangoproject.com/en/3.0/topics/http/file-uploads/>

<https://medium.com/@muehler.v/node-js-opencv-for-face-recognition-37fa7cb860e8>

## **Directory Traversal in Python**

<https://www.pythoncentral.io/how-to-traverse-a-directory-tree-in-python-guide-to-os-walk/>

---

## **Sentdex**

<https://www.youtube.com/channel/UCfzlCWGWYyIQ0aLC5w48gBQ>