RASPBERRY PI, FACIAL RECOGNITION SURVEILLANCE SYSTEM

Beng(Hons) Software and Electronic Engineering

G003745358, Junior Ajala

INSPIRATION

During my third year work placement in Xperi I was assigned to the image and video processing team here I designed and wrote programs that implement automated test cases to check the quality of image and audio processing products. During this placement, my interest in facial recognition grew and I wanted it to be implemented in my final year project

WHAT IS MY PROOJECT

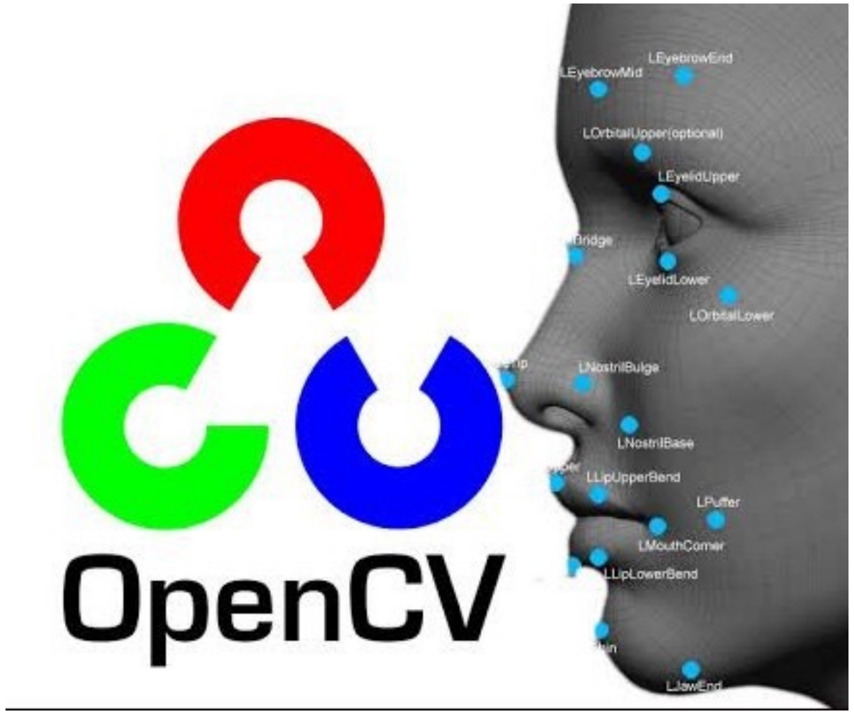
My Project is A Surveillance webpage that uses Flask frame work to run. My Project uses a raspberry pi to execute the code and has an external Pi camera attached to it. In this webpage users login or create an account which will lead a live Steam view of the contents of the PiCamera, with help of OpenCV and the Facial recognition Libary, the camera is now able to detected a persons face, if the users is not in the database. the camera will detect the user as unknown, if the camera does the detect a face from the profiles folder, the name of the person will then be printed below.

RASPBERRY PI

Raspberry Pi is the name of a series of single-board computers that runs Linux, but it also provides a set of GPIO pins, allowing you to control electronic components”.Raspberry Pi is used learn programming skills, build hardware projects, do home automation, implement Kubernetes clusters and Edge computing

OPENCV

OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more.

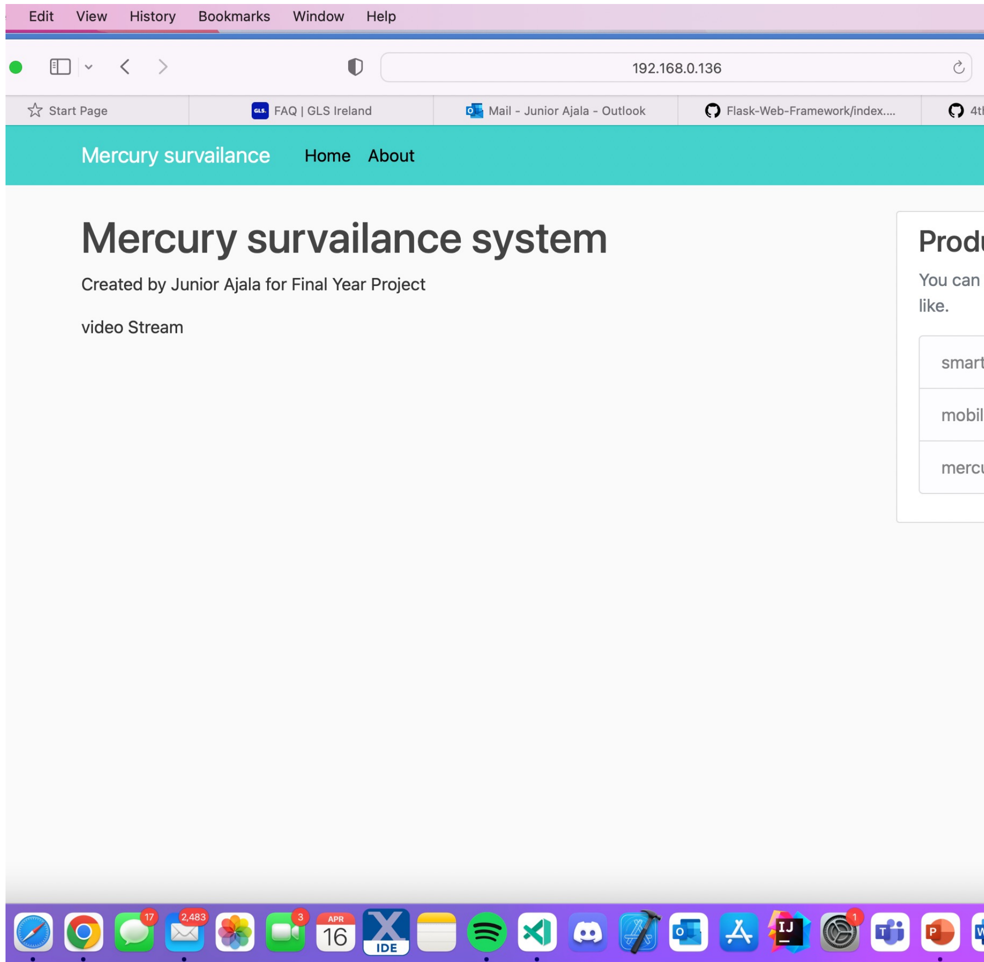


FLASK

Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. Since most of my code was already in python I chose this web frame and it was very quick and easy to learn the basics

MY MERCURY WEBPAGE

For my project I decided to call the start up company mercury. In my webpage users can register an account, when users are done registering, it is first checks for any errors (e.g username being to short) after it successfully passed all error checks, the users details are then added to the database and they are now able to sign in successfully on the log in page. The library used to help create the database is Sqlalchemy.


Edit View History Bookmarks Window Help
192.168.0.136
Start Page
FAQ | GLS Ireland
Mail - Junior Ajala - Outlook
Flask-Web-Framework/index....
Mercury survailance
Home
About

Mercury survailance system

Created by Junior Ajala for Final Year Project

video Stream

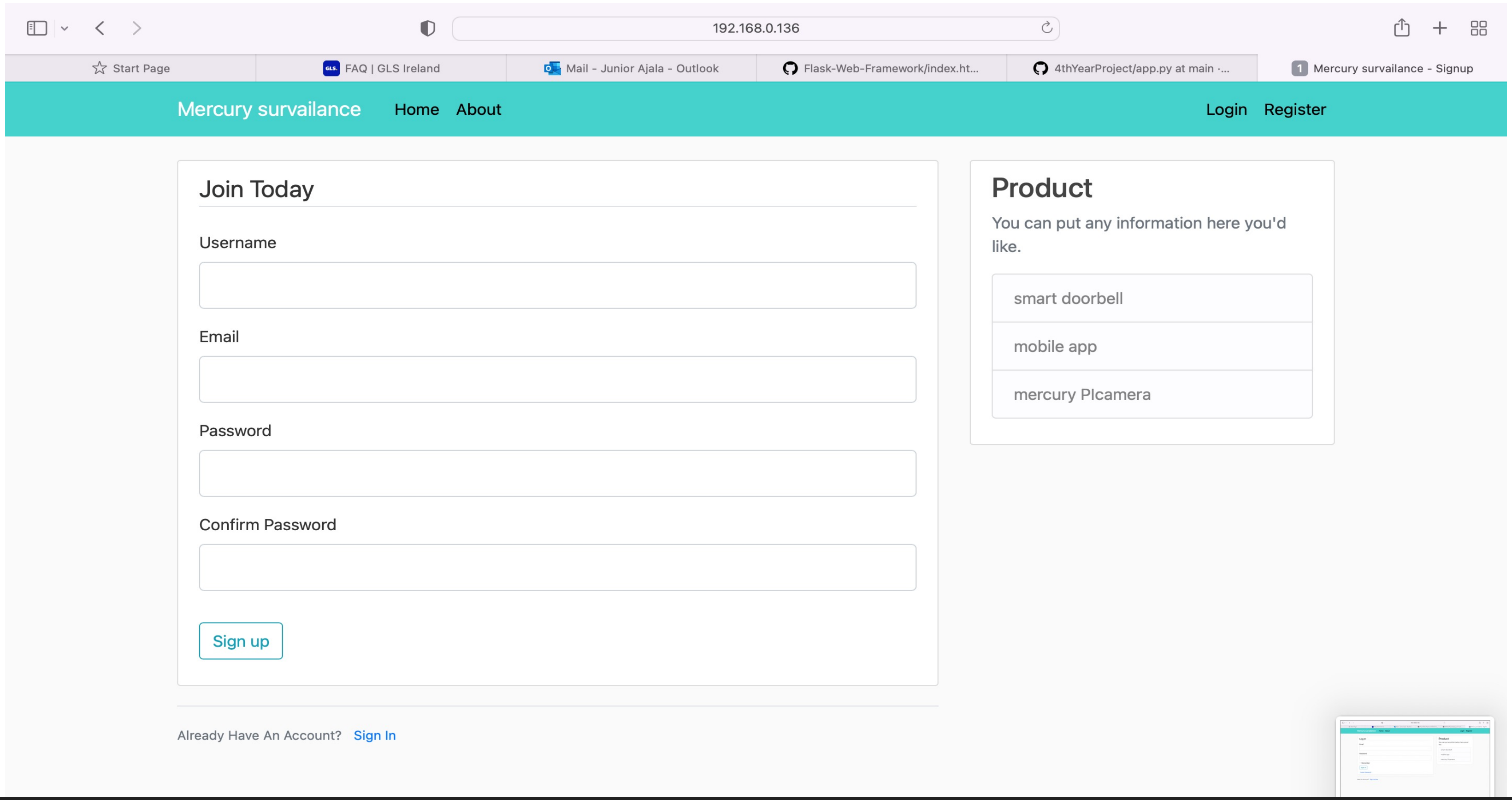
Produ

You can
like.

smart

mobil

mercu



192.168.0.136
Start Page
FAQ | GLS Ireland
Mail - Junior Ajala - Outlook
Flask-Web-Framework/index.ht...
4thYearProject/app.py at main -...
Mercury survailance - Signup

Mercury survailance Home About Login Register

Join Today

Username

Email

Password

Confirm Password

Sign up

Product

You can put any information here you'd like.

smart doorbell

mobile app

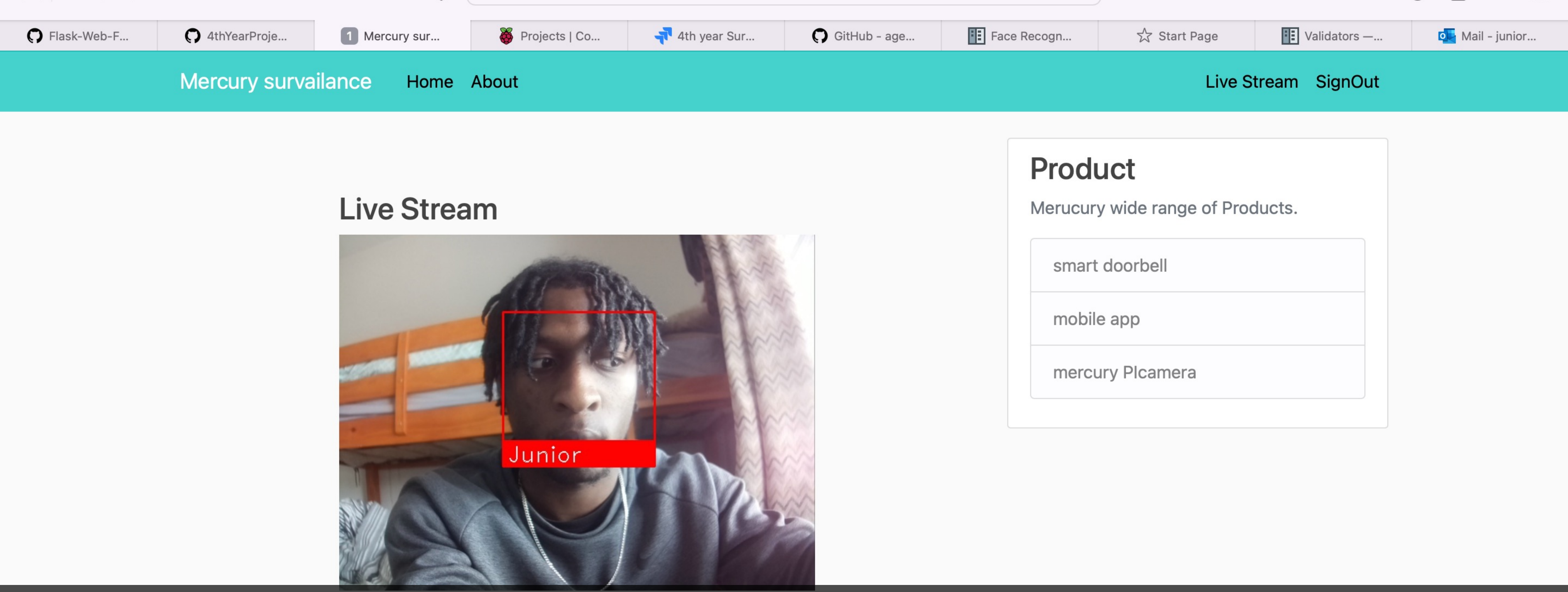
mercury PIcamera

Already Have An Account? Sign In

REGISTRATION PAGE

LIVE STREAM PAGE

When a user has logged in successfully. They are navigated to the live stream page here users can catch live footage from the Pi camera which is an external camera I connected the raspberry pi. With the code written it can detect faces and if this face is in the database is will recognize it



Live Stream

Product

Merucury wide range of Products.

- smart doorbell
- mobile app
- mercury PIcamera

FACE RECOGNITION

VALIDATORS

Before adding the users Log in details into the the db we must check for validators. In the forms.py file we write these validators this is thanks to the wtform library with this we can make validators like the min characters in the username or if the email is a real email

SQLALCHEMY DB

One the users details have passed through the validator we pass it into the database. In the database.py I created a table for the db. In this table we have columns for the users Username, email and password. So once the user has registered we add these to are db table and the users are now capable of signing in

HASHING THE PASSWORD

When adding the users password in to the database table, we don’t wont the users actual password, we do this in order to prevent and and keep the users account secured. In order to do this we must hash the password, we do this using the Bcrypt library which is a flask extension library. With this library are users password are hashed and if any hacker got a hold of the database they wouldn’t know the users password

FACIAL RECOGNITION

On the live stream page we see live footage from the Pi camera and thanks to the facial recognition library we can detect and recognize faces. With this library I first load and sample a picture from the profiles and it uses machine learning to recognize it.

Once the camera has detected a face it then compares it to the pictures it learned to recognize , with opencv we draw a box around the persons face and printing there name below or if not recognizes it will print “Unknown”