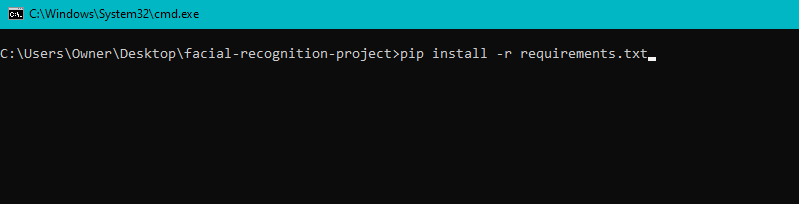
**Introduction --Mark**

**System Requirements -- Blaine**

In order to run the project, there are several requirements needed.

1. A webcam is required to use the App as when it comes to logging in you will be prompted to scan your face for two factor authentications. You are also required to upload a self-portrait Image of yourself in order to register an account.
2. You are required to install the latest version of python 3. Link(<https://www.python.org/downloads/>)
3. The project decencies are required to run our application. To install them follow the steps below!
4. Clone the project to your desktop.
5. Open the project and open command prompt
6. Enter ‘pip install -r requirements.txt’ to download the dependecies
7. You are required to have Cmake(<https://cmake.org/>) and Dlib(<http://dlib.net/>) installed
8. A text/Script editor you have installed. For example Visual Studio Code

**Technology used—ALL -- What and why used it**

In the process of making our project we used several different technologies.

1. Python(<https://www.python.org/downloads/>)

We used python as our programming language as it has wide variety of libraries we could use making it our best option over other languages. Python is also very simple to use , proof being that this is our first python project .

1. Flask(<https://flask.palletsprojects.com/en/1.1.x/>)

We used Flask as our python framework for our web application.

Flask is a small and powerful web framework for Python.

It's easy to learn and simple to use, enabling you to build your web app in a short amount of time. Flask can be used for building complex, database-driven websites

We chose flask due to its extensive documentation online so if we had issues we could easily find a soloution

1. MongoDB(<https://www.mongodb.com/>)

MongoDB is an open source database management system (DBMS) that uses a document-oriented database model which supports various forms of

Data. This is one reason we chose it as we have several different types of data we needed to add such as numpy and binary arrays. It also has a very flexible data model and is very easy to learn!

1. Webcam

We chose to use a webcam as we required one for facial recognition.

A webcam is essential for our application as without a webcam the user cannot login to our App due to our facial recognition two factor authentication.

Any webcam should work that is connected to your Computer.

1. PIL(<https://pillow.readthedocs.io/en/stable/>)

PIL is the Python Imaging Library by Fredrik Lundh and Contributors.

We used PIL to convert images to bytes then to base 64 so we can easily store it on our database.

1. Face recognition Module(<https://github.com/ageitgey/face_recognition>)

We used Adam Geitgeys (<https://github.com/ageitgey>) Face recognition Module(<https://github.com/ageitgey/face_recognition>) in our project for our two factor authentication when logging in.

Python , flask , face recognition module ,Webcam,pil,mongoDB ,

**Design methodology – ALL**

When starting our project we specified our language we would use ,our end goals and what imports we could use.

1. Goals

**Features of the implementation – ALL**

**Limitations and known bugs – ALL**

**Testing plans – ALL**

**Recommendations for future development - 1 – Conor**

There are many aspects of this project that could be expanded upon after the duration of the allotted project time. We were limited by time but certainly we could improve many aspects of our software. There were three main areas that we developed as part of our project. I developed the facial recognition scripts and camera, Blaine developed the MongoDB backend aspect, and Mark developed the frontend Flask web application.

**Current state of the software**

Currently the software is extremely malleable and not concrete by any means. It could certainly be taken in almost any direction. As of right now it features a database connected to the frontend flask application. With two factor authentication (Enter details and also facial recognition) as part of logging in. This software could potentially be used in several ways as I will highlight below.

**Conclusions - ALL**