**Durham College**

**Artificial Intelligence Analysis, Design and Implementation**

**Capstone Term II - AIDI 2005 – 01**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Facial Attendance Management System**

**Group-12 Members: -**

1. **Himanshu Dhiman - 100721563**
2. **Shubham Dikshit - 100688204**
3. **Harin Shah - 100674815**
4. **Ankitkumar Patel – 100717791**

**Date: - 21- Feb- 2020**

**Modelling**

* **Model(s) architecture and software pipeline: -**

The facial attendance system is designed to register student attendance via facial recognition technology. The system will also check the students are on time or has left early. We can also track the time of arrival and departure. We also want to be able to add or delete student.

We can place the camera in front of the door which will recognize the students name and save the time of arrival and departure. So, with that information it will help to determine if they are late or if they left earlier.

All that information stored in an online database to make it accessible from anywhere.

Here we are going to use three technologies to build the application: -

1. **Front-end:** ReactJs [logo React](https://github.com/GraphtyLove/Face-recognition-project/blob/master/assets/react-logo.png)
2. **Back-end API:** Python Flask [logo python](https://github.com/GraphtyLove/Face-recognition-project/blob/master/assets/python-logo.png)
3. **AI model:** Face\_recognition or FaceNet
4. **Installation and environment setup:** Bash [logo bash](https://github.com/GraphtyLove/Face-recognition-project/blob/master/assets/bash-logo.png)
5. **Database**: PostgreSQL [logo postgresql](https://github.com/GraphtyLove/Face-recognition-project/blob/master/assets/postgresql-logo.png)

* **Front-end**: - To create the **front-end** we use React which is perfect for processing information in real time.

1. React is a JavaScript library for building user interfaces.
2. React is used to build single page applications.
3. React allows us to create reusable UI components and it has a simple and clear syntax.

* **Back-end**: - We will use Python Flask to create an API which can receive request and data, then send back and answer.

Flask is a lightweight and micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

The purpose here is to receive the data from our face recognition model and redistribute it to the front when requested. But also, to have the possibility to add a new student details with his name and photo if face is not recognized and to be able to delete some only by recovering his name.

For example, the API will receive a student name, make a request to the database to have all the data about this student and send back those data.

* **Database**: - We will use **PostgreSQL** database which is free and open source relational database management system. It allows us to add custom functions developed using different programming languages such as python, C/C++, Java, etc.

PostgreSQL has many advanced features that other enterprise database management systems offer, such as:

* User-defined types
* Table inheritance
* Sophisticated locking mechanism
* Foreign key referential integrity
* Views, rules, subquery
* Nested transactions
* Multi-version concurrency control (MVCC)
* Asynchronous replication
* **Face recognition**: - For the**face recognition,** we will use a python library called "**face\_recognition**". And FaceNet which is google face recognition technique.

Face\_recognition library is recognizing and manipulate faces from python or from the command line with the world's simplest face recognition library.

It is built using dlib's state-of-the-art face recognition built with deep learning model. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

**Workflow of architecture: -**

1. **Data processing: -**Camera detects the student’s face and it will check if the student is in that particular class or not and if so, it will retrieve the data, the name of the student and time it detected him. If this is the first time this student is detected, an arrival time will be assigned, and faculty will get notification.

API

Database

Camera with face recognition

Front End

1. **API**: - When system will detect the student face, get the time of detection and send that information to the API. Then API will ask to the database if the student is in condition to be on time and he is belonging to that class or not and send back all those data to the DB.