Odd Semester (2022)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work****)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | |  | |  | |
| **Student Information**: **Surname** | | | | | **Given Names**  Hengky | | **Student ID Number**  2201852492 | |
| 1. | | Sanjaya |  | |
|  |  |
| **Course Code** | **:** COMP6056 |  |  | | **Course Name** | | **:** Program Design Methods | |
| **Class** | **:** L1CC |  |  | | **Name of Lecturer(s)** | | **:** Jude Joseph Lamug Martinez | |
|  |  |  |  | |  | |  | |
| **Major** | **:** CS |  |  | |  | |  | |
| **Title of Assignment** | : FaceRecognition Attendance | |  |  | |  | |  | |
| **Type of Assignment**    **Submission Pattern** | **:** Final Project |  |  | |  | |  | |
| **Due Date** | **:** 21-11-2018 |  |  | | **Submission Date** | | **:** | |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

Hengky Sanjaya

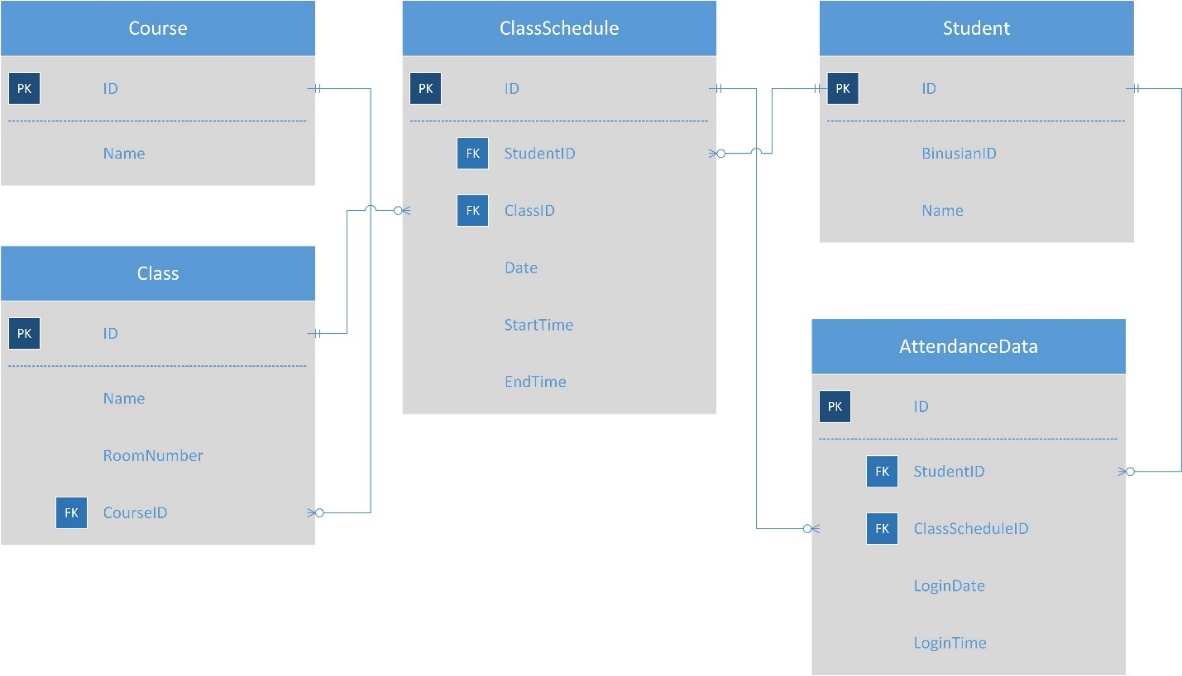
**“Face Recognition Attendance”**

**Name : Hengky Sanjaya**

**ID : 2201852492**

1. **Program Description**

This application is a web-based application to help people do their attendance easier using face recognition. This application is built using python version 3.7, Django framework version 2.1.2, SQLite for the database management, bootstrap for the user interface design. This program is meant to help people who always forget to bring their identity card.

1. **ERD & Class Diagram**
2. Course

To store all the courses data

1. Class

To store the class data with the room number and the course.

1. Student

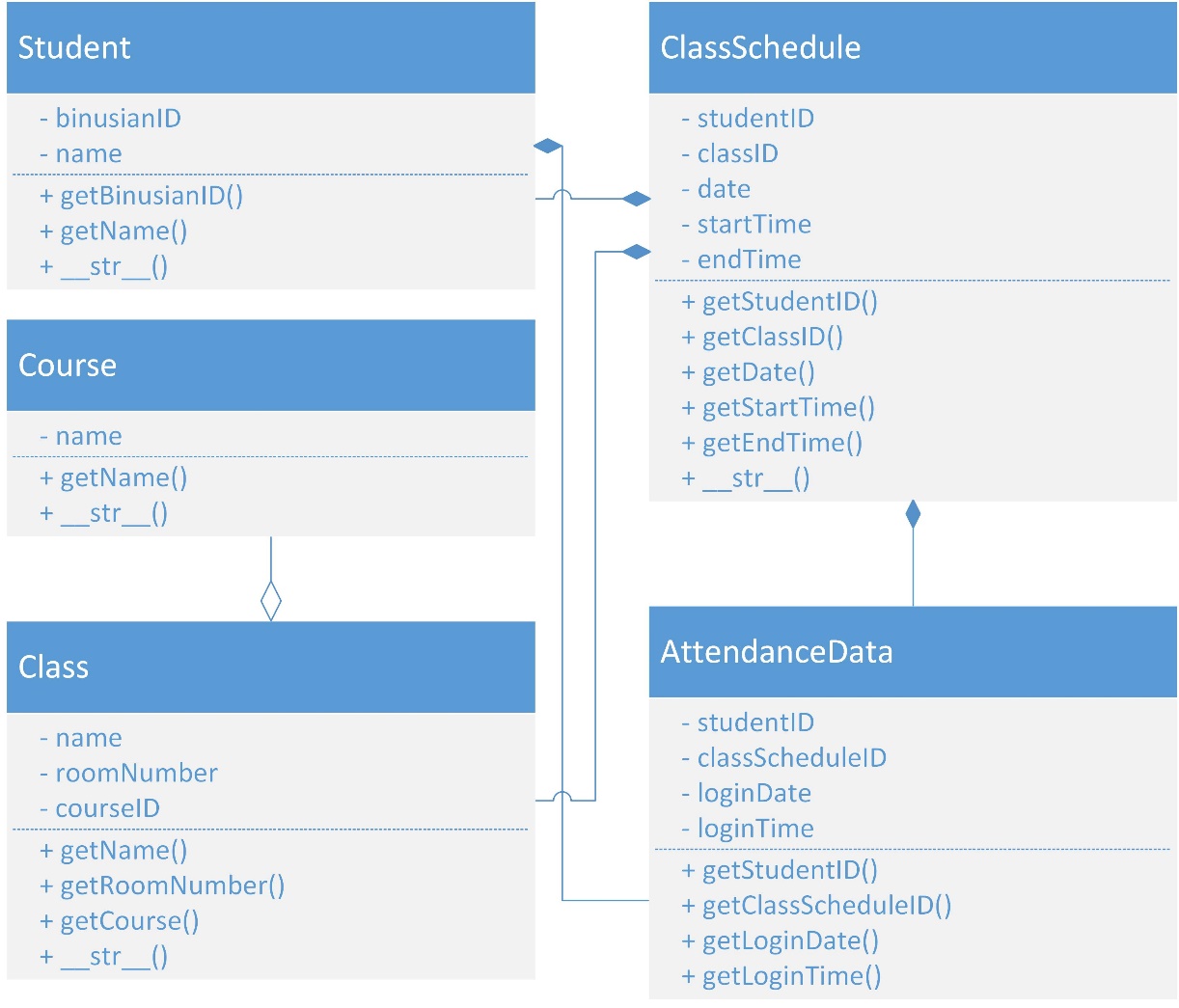
To store all the student data

1. ClassSchedule

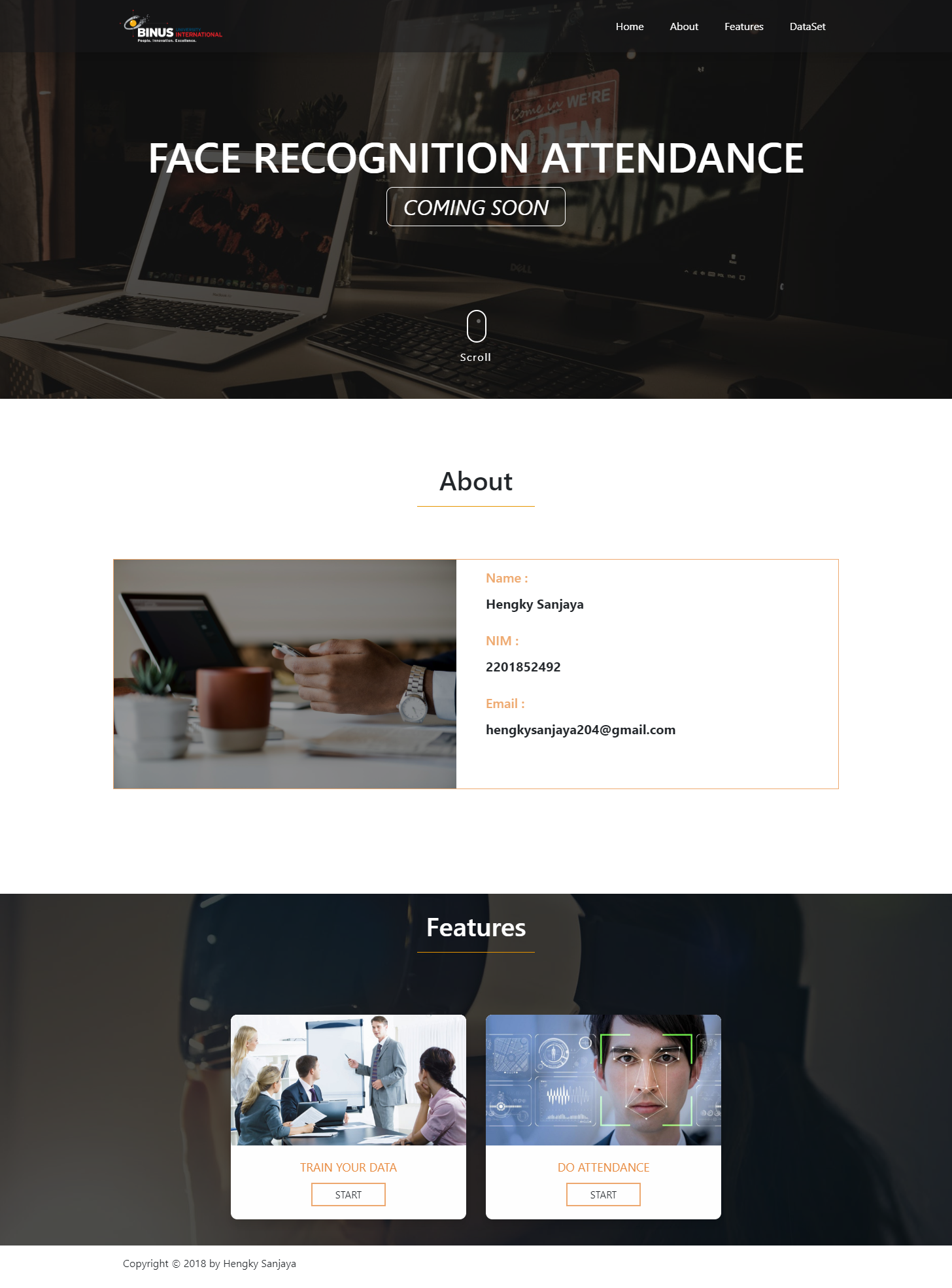
To save the student’s schedule data with the start date time, end time and what course in it.

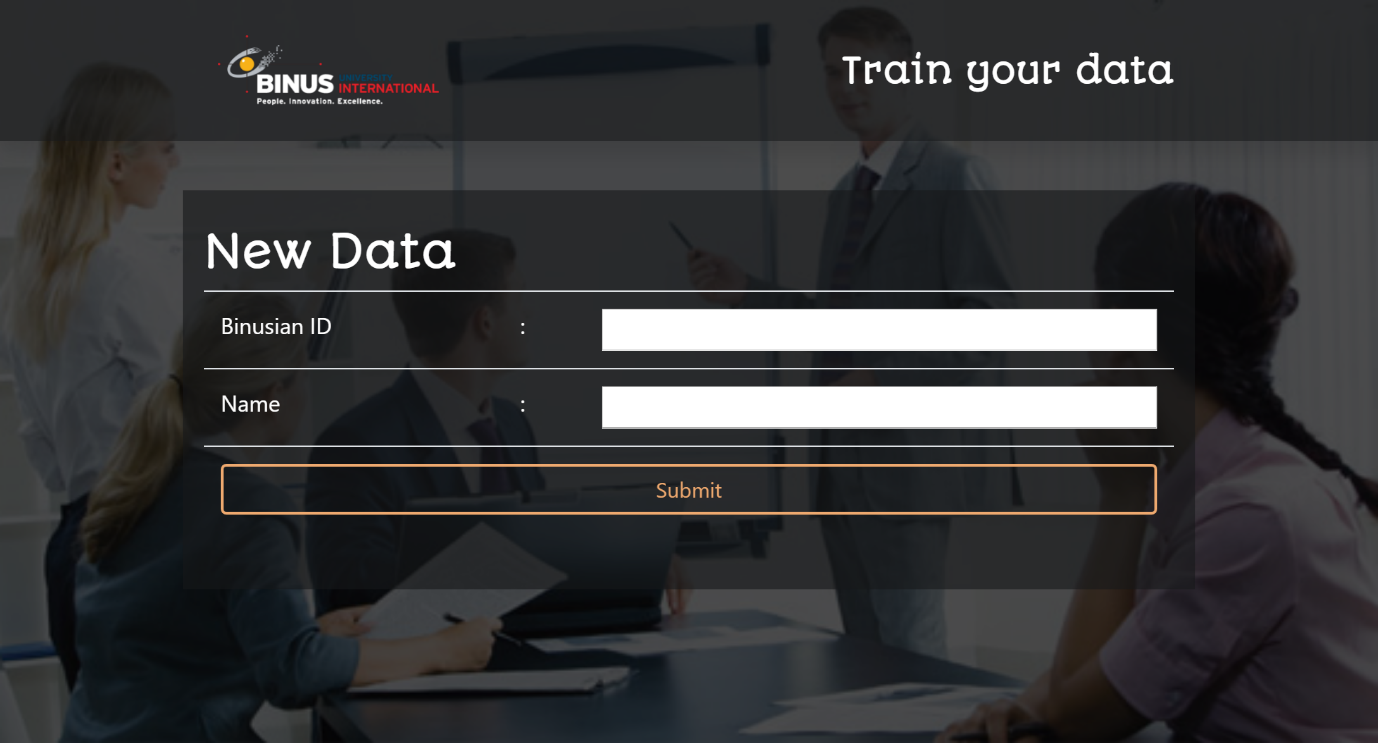
1. AttendanceData

To record the student attendance data in specific ClassSchedule



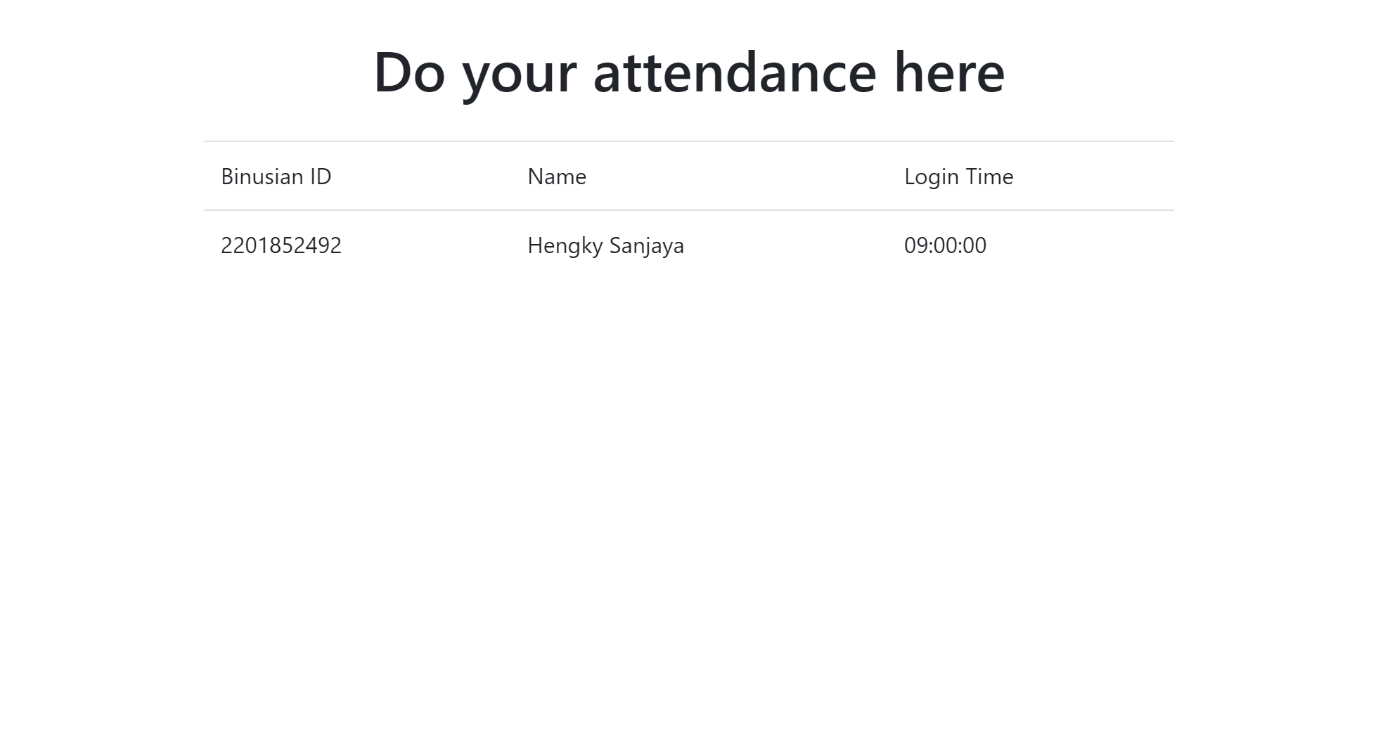
This is my class diagram. For each model I create the setter getter for each variable and also built in function called \_\_str\_\_() to return a string with specific format when we call the instance of the model.

1. **Application Interface**
2. Homepage
3. Train Data Page

This page is used for train the face recognition dataset, so when the submit button is clicked the system will open a new window to take a picture of the student new save the picture and save the data into database.

1. Do Attendance Page

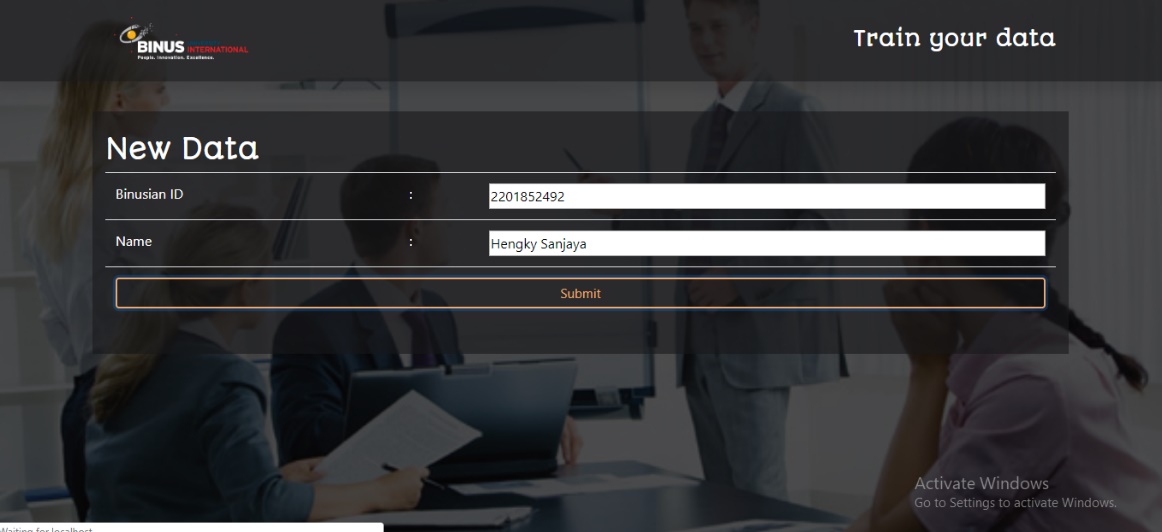
To see the detail attendance data, we go to this page by click the start do attendance button in the homepage.



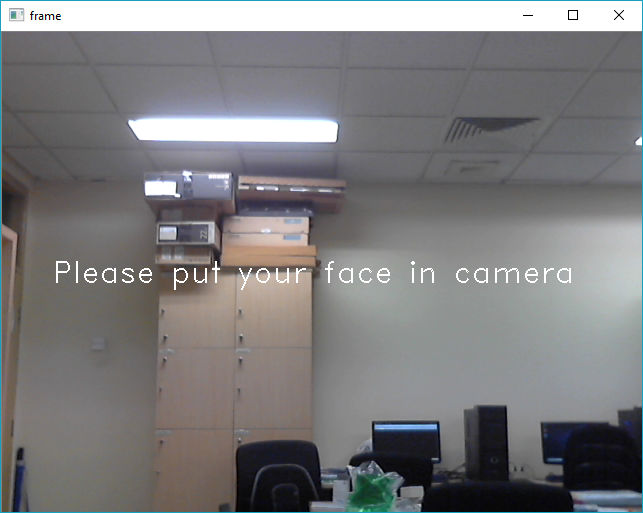
1. **How the program works**

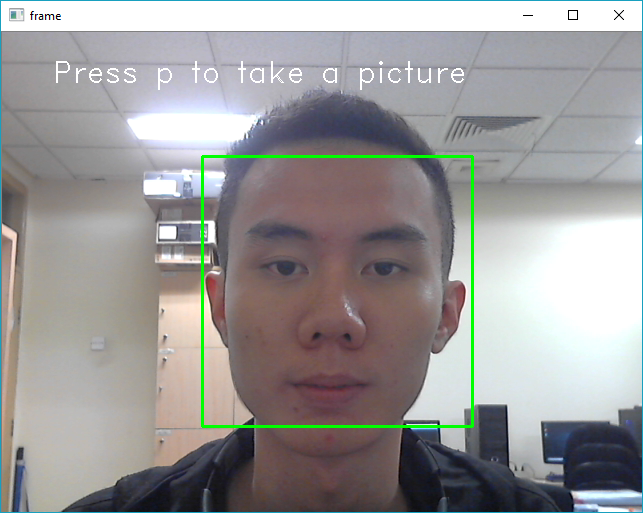
* **To Train the Data**

After input the BinusianID and name, then click ‘Submit’ Button. Then the system will open a window to take your picture.

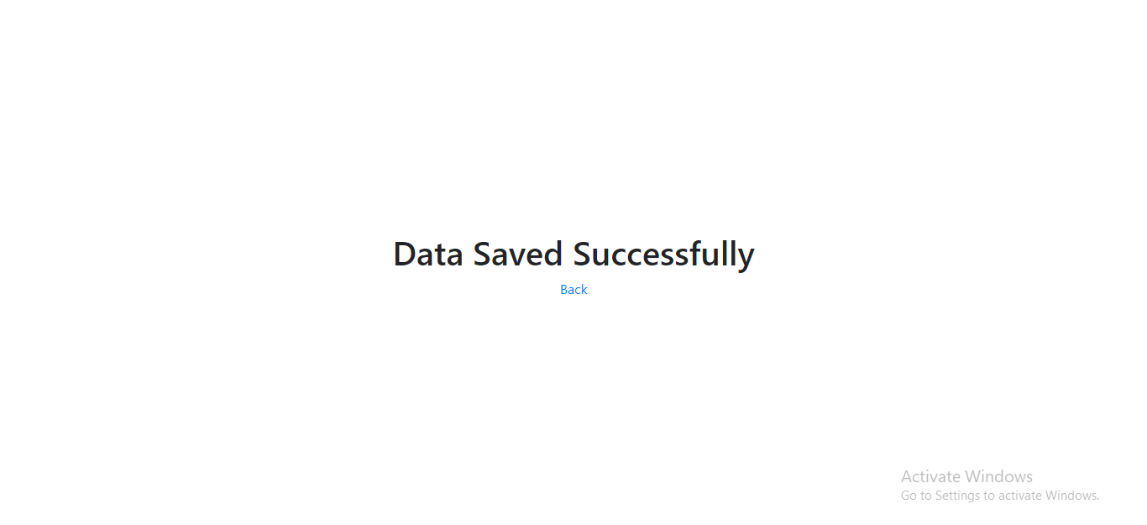


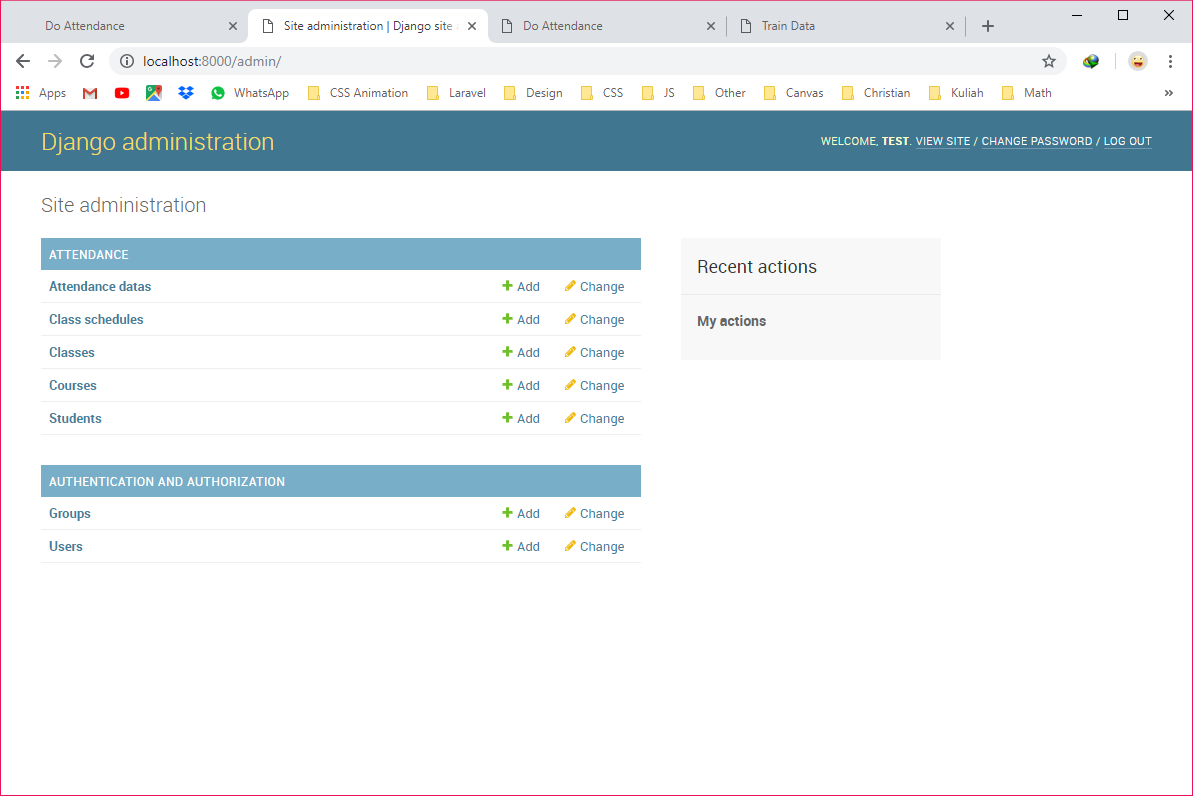
If the system does not find any face then a message will appear in the window to ask the user to put their face in camera.



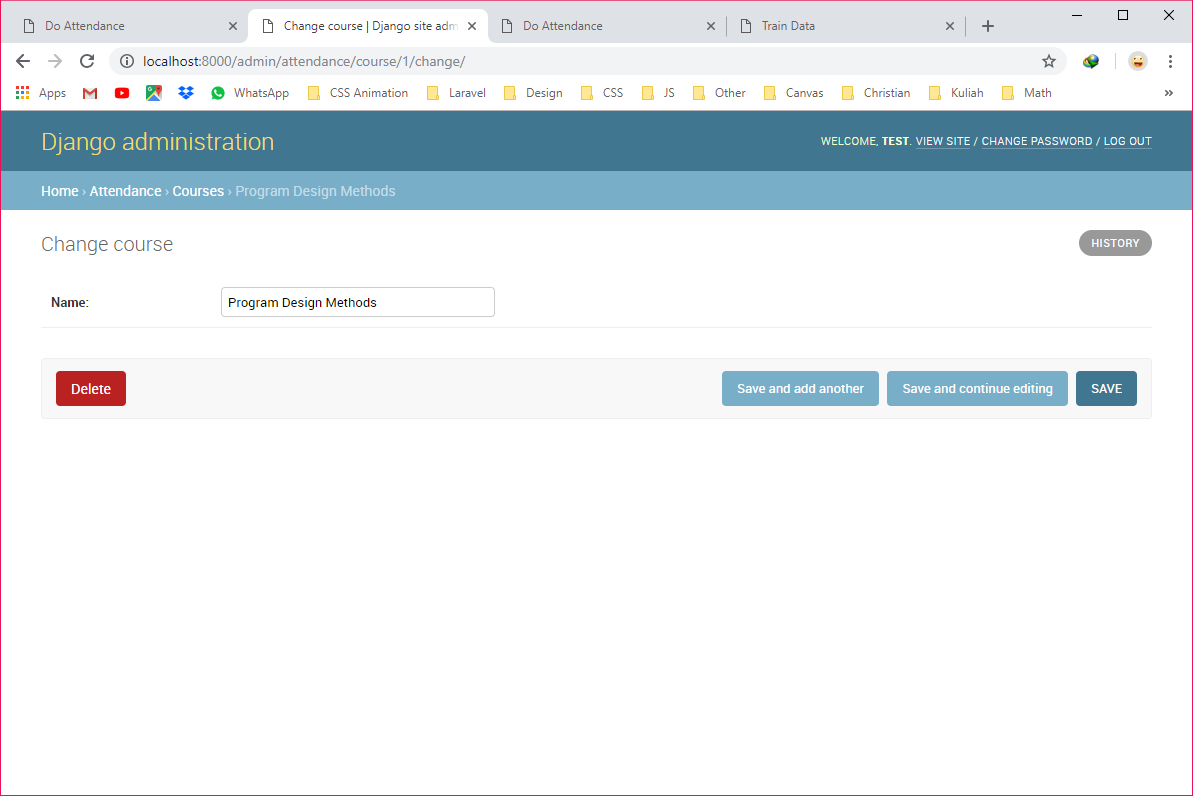
Then if a face founded, you can press ‘p’ in your keyboard to take a picture

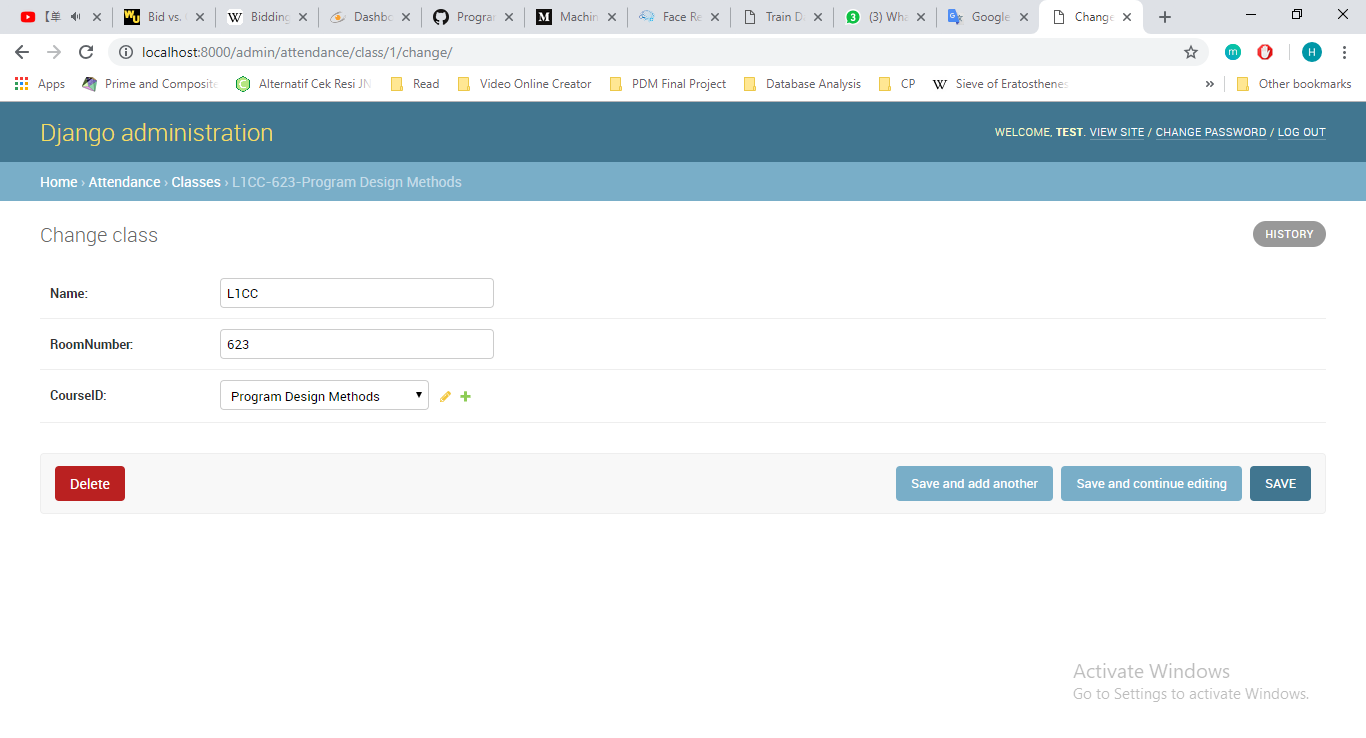
After you press ‘p’ in your keyboard the system will save your picture, and also save your data into database to be used for the attendance then Success message will appear in the page.

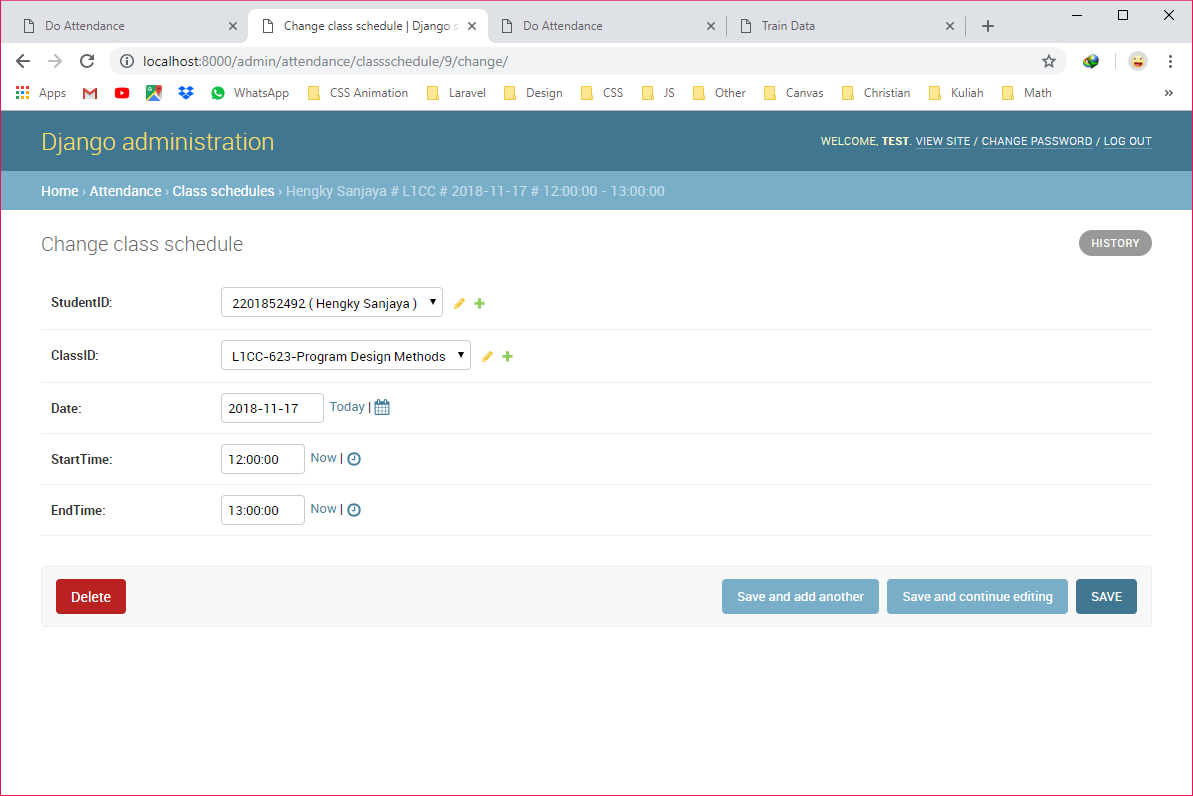


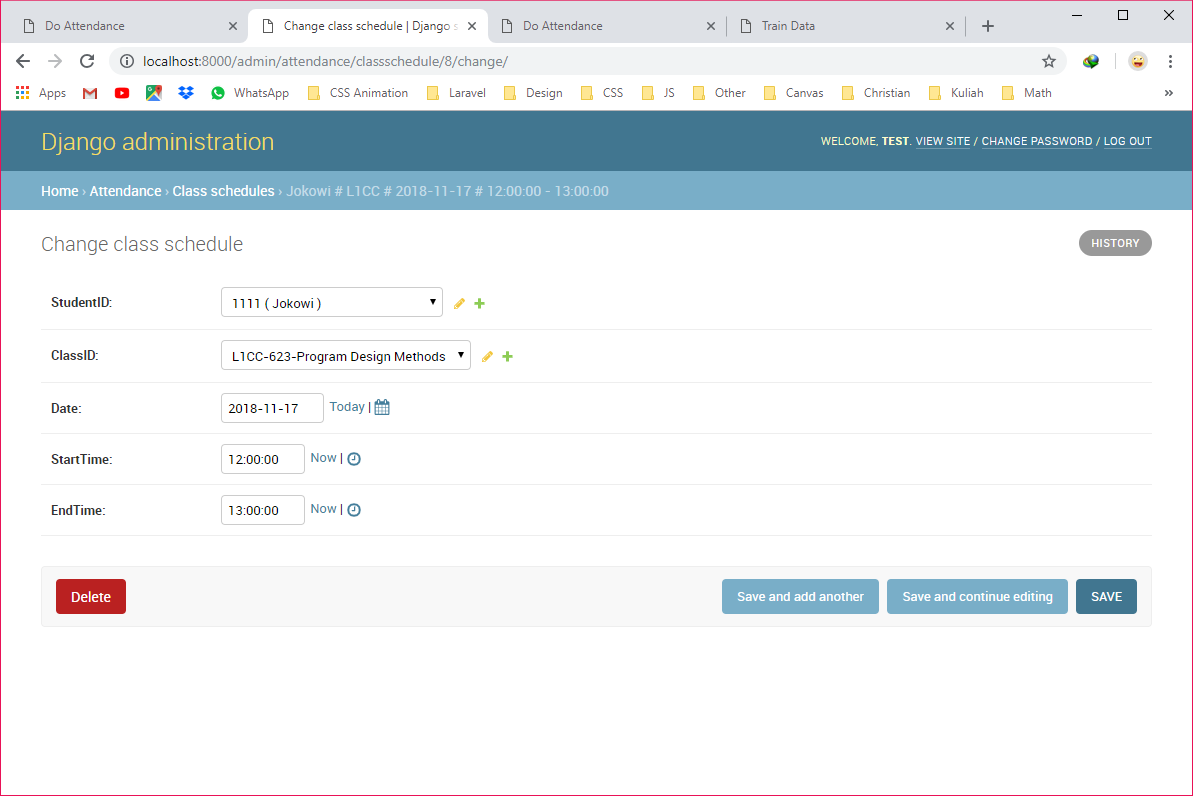
Before you can do your attendance of course you should have a class schedule. So, we need to add some data to several master table.

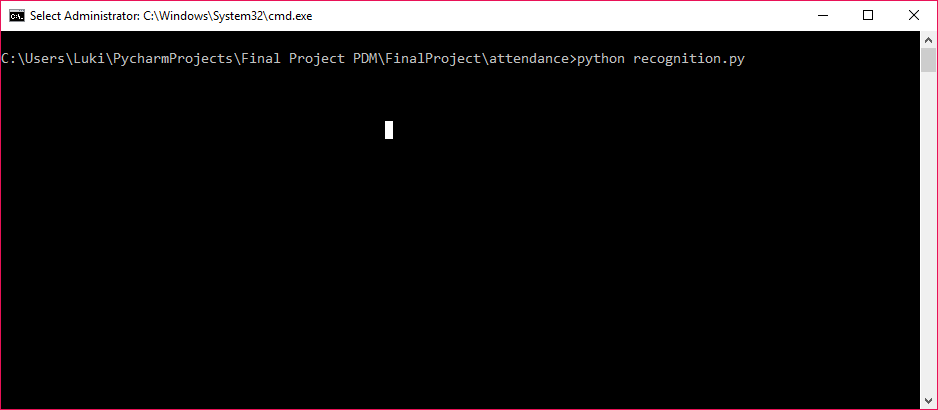
This is the Django administration homepage.

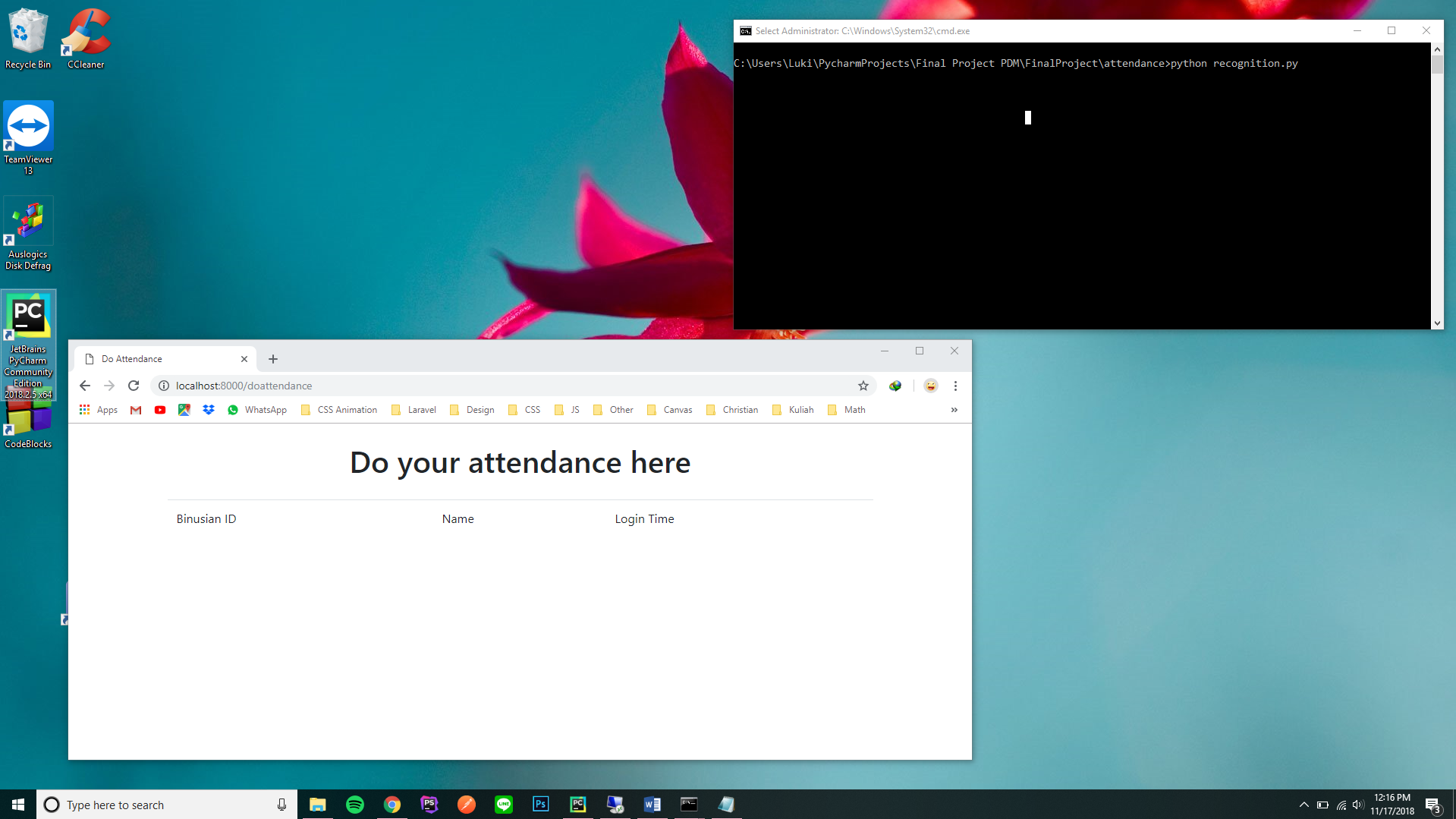
First, add course data for example “Program Design Methods”

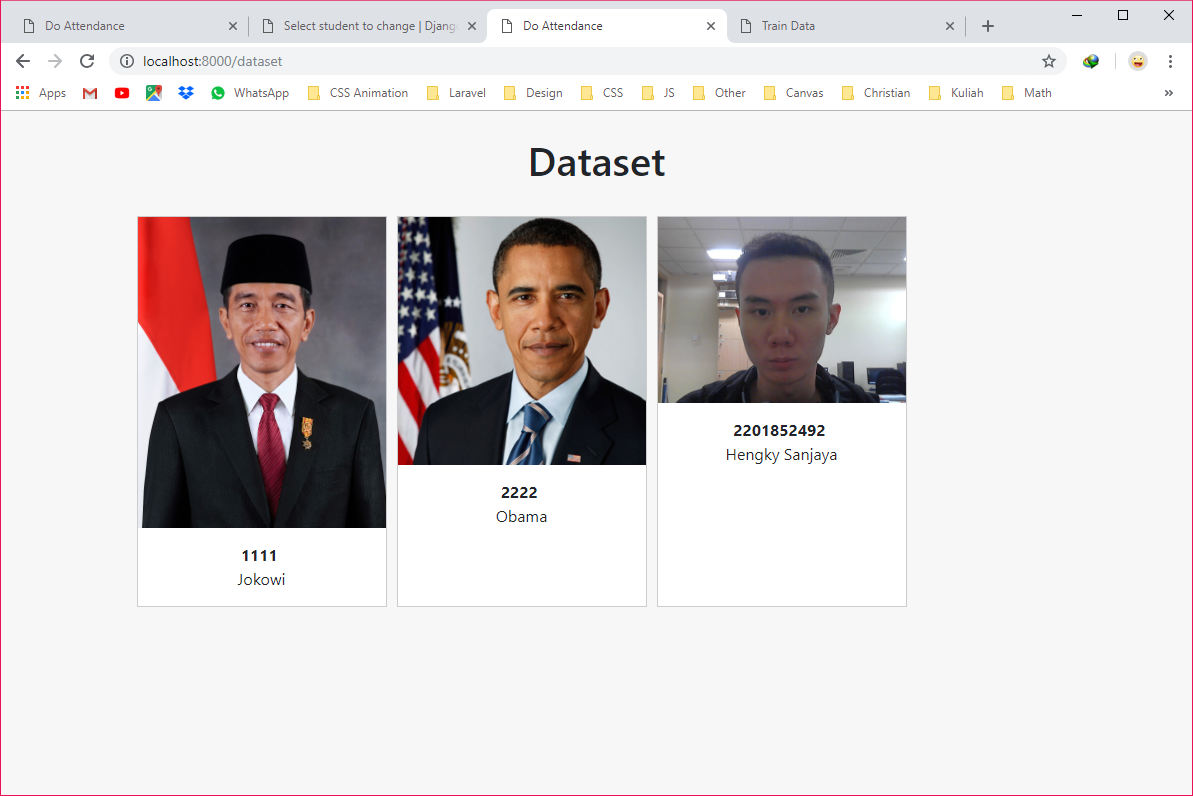
Add class data such as Name, Room Number, and choose Course that has been added before.

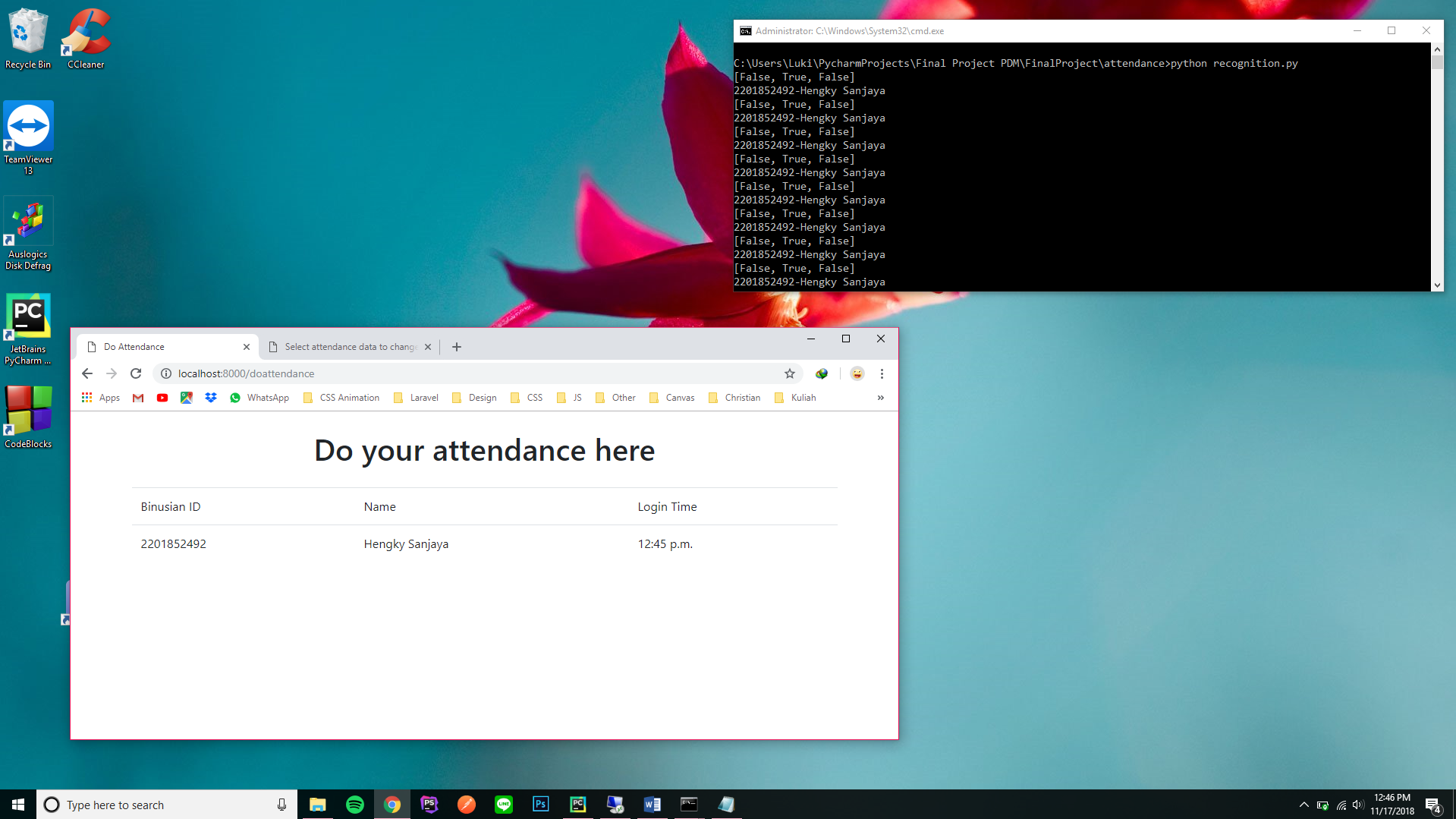
Then we add Class Schedule data. The student can only do the attendance within their class schedule time range



Then to do the attendance I run the face recognition to run in background

Then I also open the Do attendance page to see the result.

Here I have several datasets

after I run the face recognition server then I put my face in the camera. Then the doattendance page will show the attendance data. But the attendance data will be recorded once only in one class schedule for each student.

1. **Library used**

There are some library that I used to build my “Face Recognition Attendance” project:

* Face-recognition version 1.2.3

To do the face detection and recognition.

* OpenCV-python 3.4.3.18

This library is used to open camera / webcam.

* OS

To use some built-in function from python like getting the list of directories from specific path

* Datetime

To get current datetime value and also parsing function

1. **Lessons that Have Been Learned**

Since this is the first time I am learning python language so this final project makes me learn and get so many new things and it’s really fun, and I also learn new framework called Django to build my final project in web based.

1. **Project Technical Description**

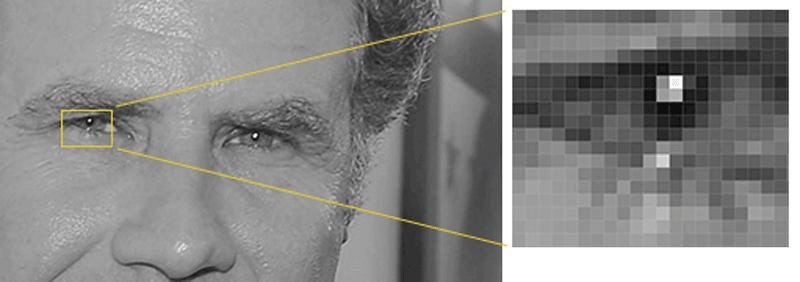
* Finding faces using OpenCV

There are basically two primary ways to find faces using OpenCV:

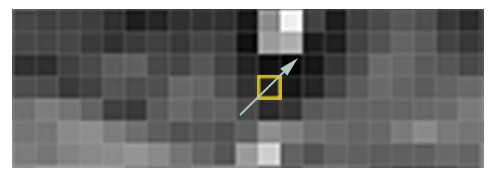
* + Haar Classifier
  + LBP Cascade Classifier

And I am using Haar Classifier because it is more accurate than LBP but it is also much slower, that’s why I have this file “haarcascade\_frontalface\_default.xml” to do the face detection.

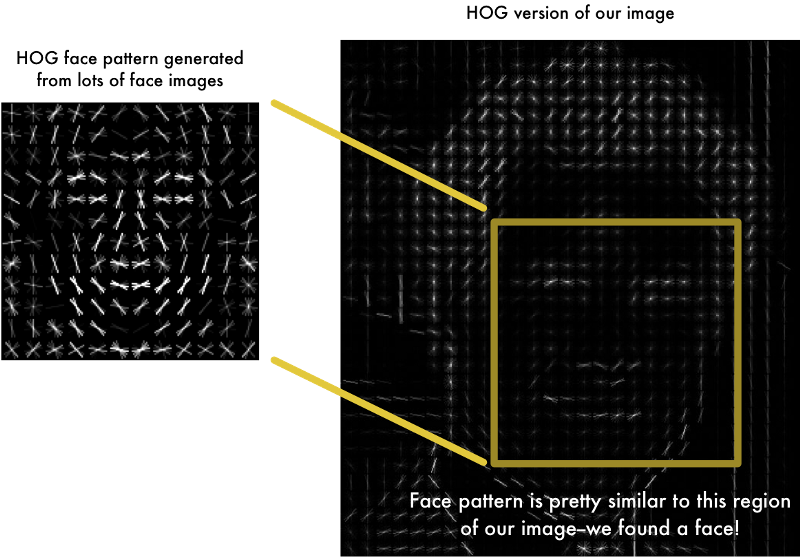
* Finding faces using face-recognition library

To find faces in an image, I start to make the image in black and white because we don’t need color data to find faces.

Then look at every single pixel in the image one at a time. Our goal is to figure out how dark the current pixel is compared to the pixels directly surrounding it. Then draw an arrow showing in which direction the image is getting darker.

These arrows are called *gradients* and they show the flow from light to dark across the entire image.

HOG = Histogram Oriented Gradient

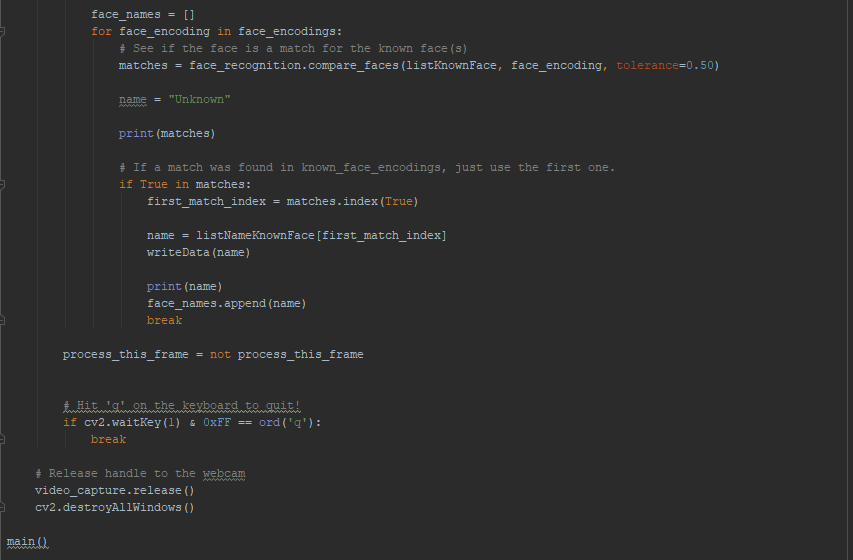


1. **Code Explanation**

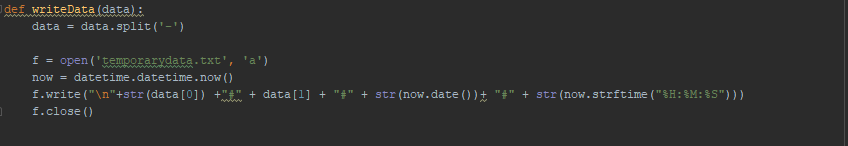
“Recognition.py” file

First, I open the camera using cv2 library then I get all the available dataset from the file and for each image I encode the image then store the encoded data into my list. After that, I read per frame of video and resize the frame into ¼ size for faster face recognition processing, and also I convert the image from BGR color (which OpenCV uses) to RGB color (which face\_recognition uses).

Then I find the face location and from the face location I encode it and compare with the list encoded known faces.

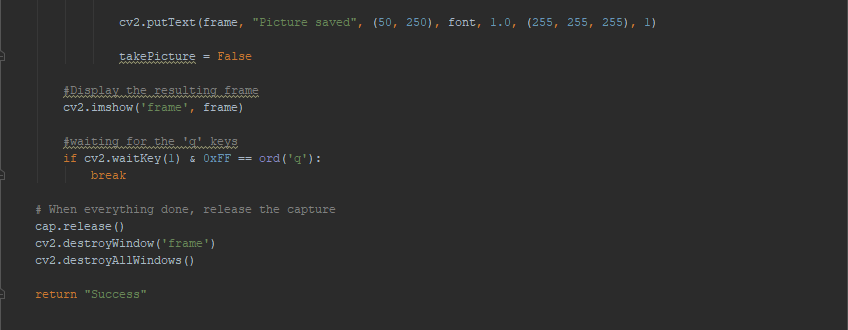
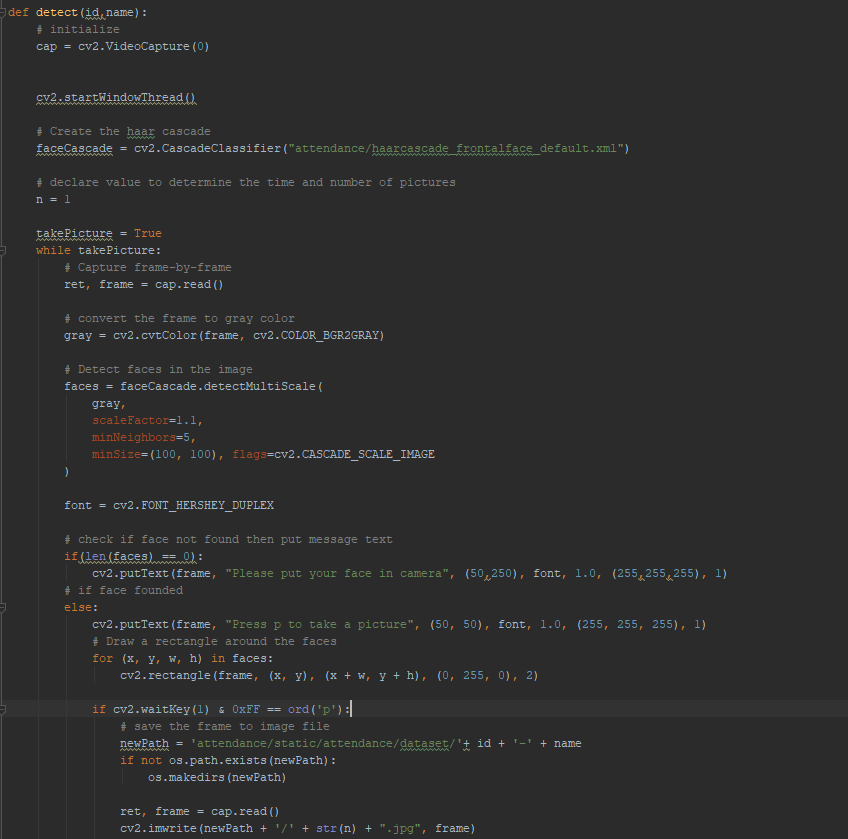


After that if a match was found I list known faces, I use the first one, and call the writeData function to write the founded face data to the text file. Then release the webcam.



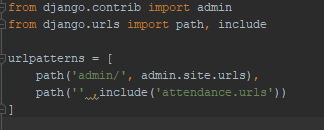
This is the writeData function to write the founded face into the text file with format binusianID#name#logindate#logintime. So, I can read from this file for attendance data.

“detectface.py” file



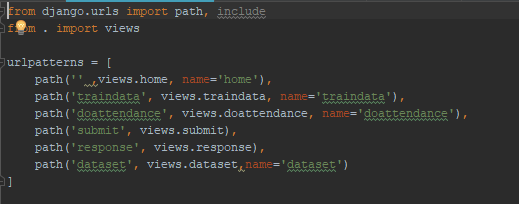
First, I open the camera using cv2 library, then I convert the frame into gray color and detect faces in the frame using cascade method. If the system doesn’t find any face, then I display a message to ask the user to put their face in camera, otherwise I will display a message to ask the user to press p to take a picture. After that I draw a rectangle around the faces and wait for the ‘p’ is pressed and save the frame into image file. Then release the capture when everything has done.

“urls.py” file



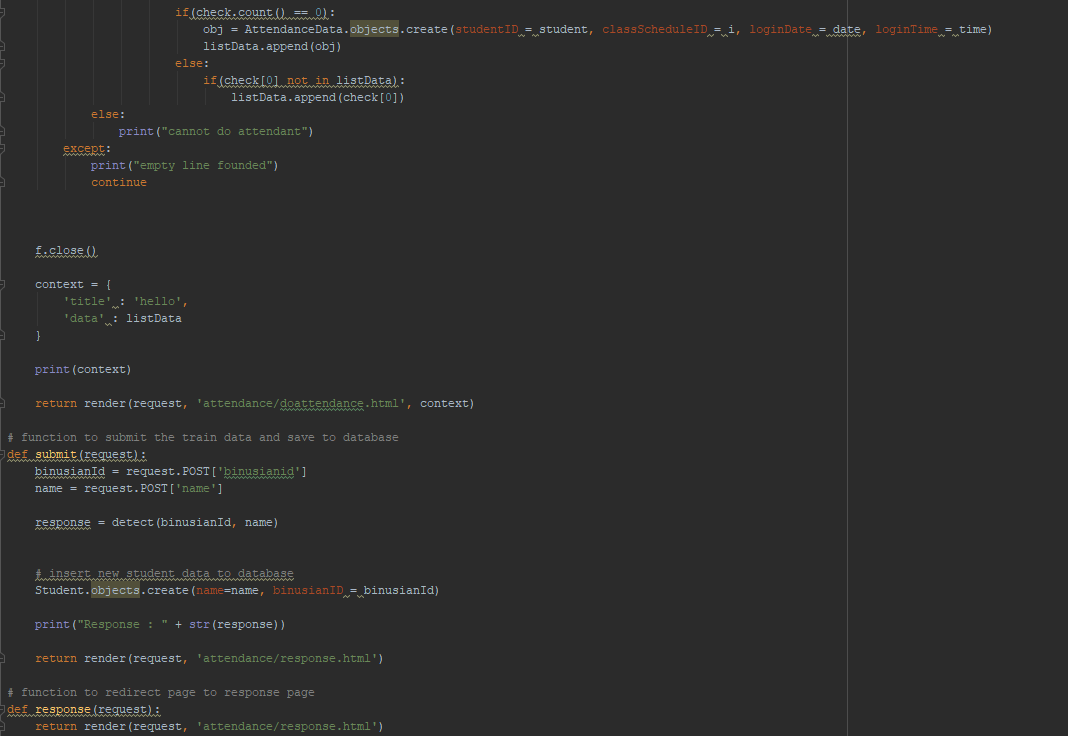
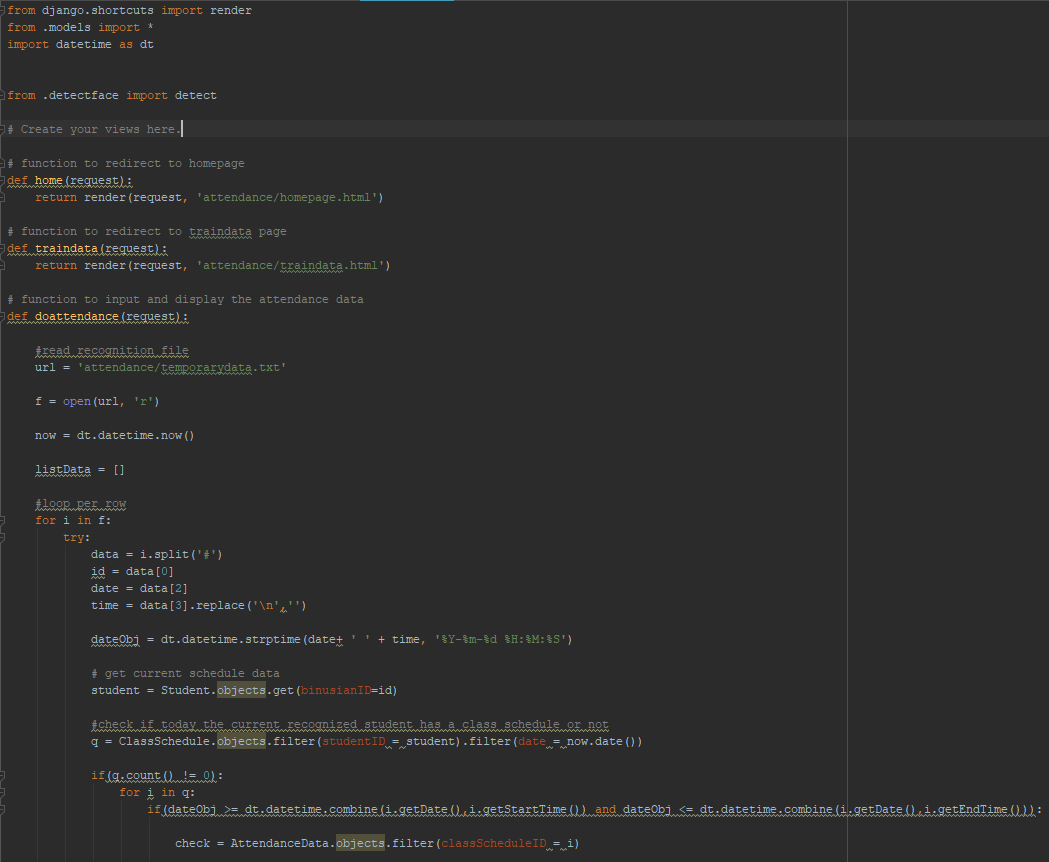
When the URL contains ‘admin’ then I will redirect the page into admin page, else if empty then I include the url file in my attendance apps

“urls.py” file in attendance apps



If the url is empty then I will redirect to the homepage, else I will redirect depends on the inputted url

“views.py” file



In my “views.py”file I have several functions such as:

* Home():

To redirect to homepage if the url contains empty string

* Traindata():

To redirect to “traindata.html” page

* Doattendance():

This function is used to read the face recognition result from the text file and validate the data with the available class schedule if the login date time within the class time then I will insert new attendance data into database. After that, I will send the attendance data to “doattendance.html” page to display it to the user

* Submit():

This function will run after the user input the data and clicks submit button in “traindata.html” page. This function will call another function in “detectface.py” file to open a camera window to take the user’s picture then save the data into database.

* Response():

To redirect to “response.html” page

* Dataset():

This function is used to open “dataset.html” page, but before that the function will retrieve the all the students data from the database and send it to that page.

1. **Project Link**

[*https://github.com/hengkysanjaya123/Final-Project-PDM*](https://github.com/hengkysanjaya123/Final-Project-PDM)

1. **References**

* [*https://www.blog.pythonlibrary.org/2018/08/15/face-detection-using-python-and-opencv/*](https://www.blog.pythonlibrary.org/2018/08/15/face-detection-using-python-and-opencv/)
* [*https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78*](https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78)
* *https://github.com/ageitgey/face\_recognition*