ATTEND-A.M.S. Manual

Prerequisites

- Python (https://www.python.org/downloads/)
- JavaScript
- NodeJs (https://nodejs.org/en/download/prebuilt-installer)
- ExpressJs (*npm init && npm install express)*
- ReactJs

Python Dependencies

- Tensorflow 2.13.0 (pip install tensorflow===2.13.0)
- Keras 2.13.1 (pip install keras===2.13.1)
- Flask 3.0.2 (pip install flask===3.0.2)
- Flask-Cors 4.0.0 (pip install flask-cors===4.0.0)
- Numpy 1.24.3 (pip install numpy===1.24.3)

Initial Set-Up

After cloning/downloading the three (3) folders: attendamsfrontend, attendamsebackend, and attendamsprediction, ensure that they are extracted.

1. Open a terminal inside the attendamsbackend folder.
2. Input **npm install** in the terminal.
3. Open a terminal inside the attendamsfrontend folder.
4. Input **npm install** in the terminal.

After completing the instructions above, the ATTEND-A.M.S. will be now ready for use.

Starting/Running the application

Every time you want to start/run the application, the following must be done:

1. Open a terminal inside the attendamsprediction folder.
2. Input **python predict.py** in the terminal.
3. Open a terminal inside the attendamsbackend folder.
4. Input **npm start** in the terminal.
5. Open a terminal inside the attendamsfrontend folder.
6. Input **npm start** in the terminal.

Application Navigation

A. SignUp/LogIn
    1. Upon running the npm start in the terminal of attendamsfrontend folder, the LogIn page will open.

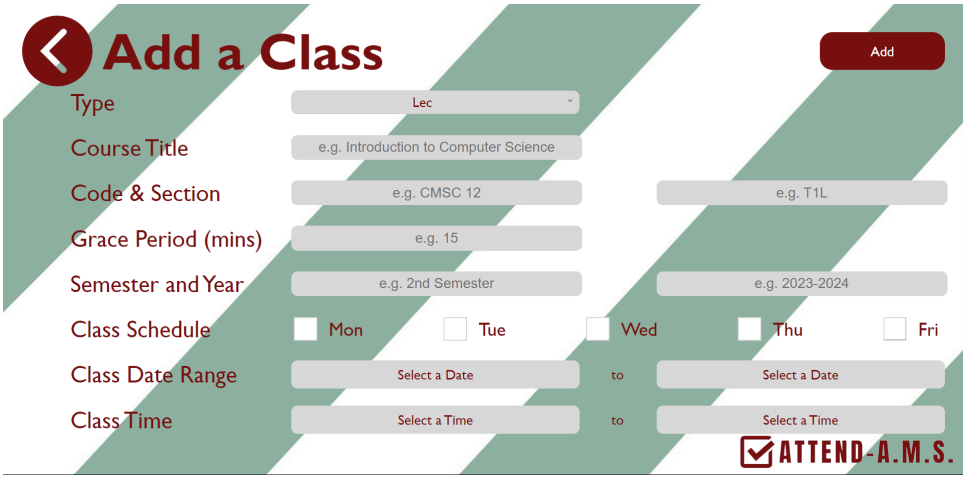
2. To create an account, click on the **No account yet? Sign Up!** Input your email address and a password and then click the **Sign Up** button.

3. To log in, input your account credentials and click the **Login** button.

B. Add Class
    1. Assuming that you are logged in, click the **Register** button.

2. Click the **Add a Class** button.

3. Fill-Up the form in the page and then click **Add**.



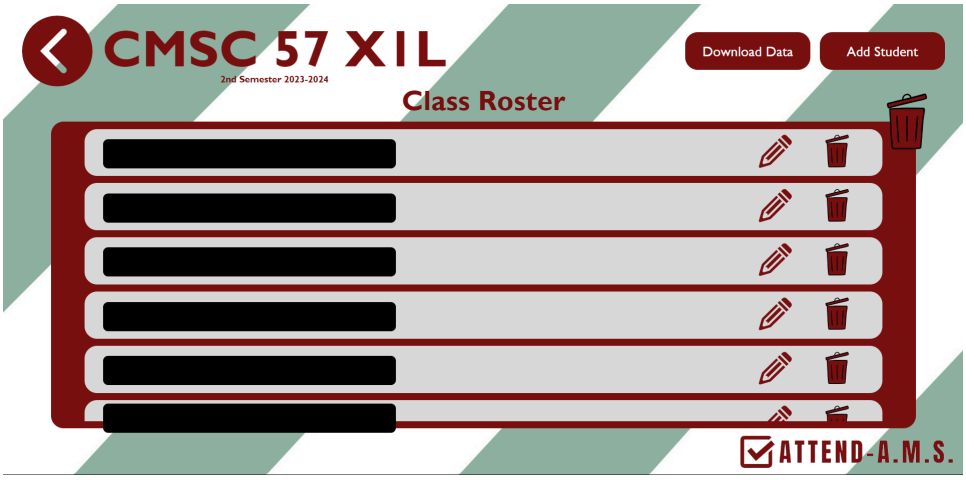
C. Add Student

1. Assuming that you are logged in, click the **Register** button.

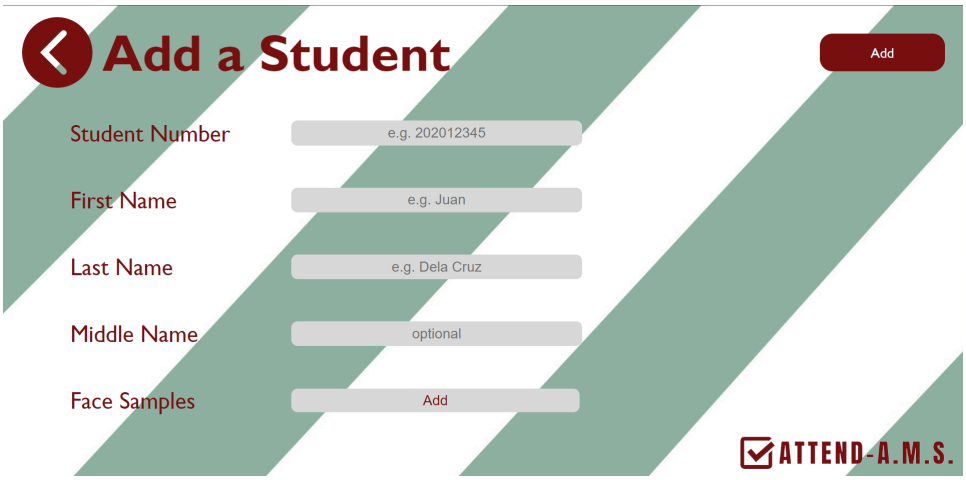
2. Click the **My Classes** button.

3. Select the class you want to add a student to.

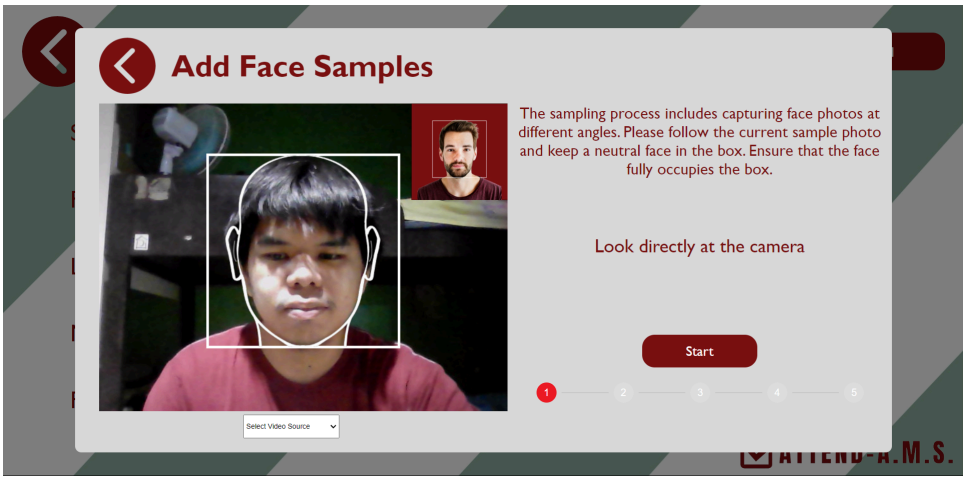
4. Click the **Add Student** button.



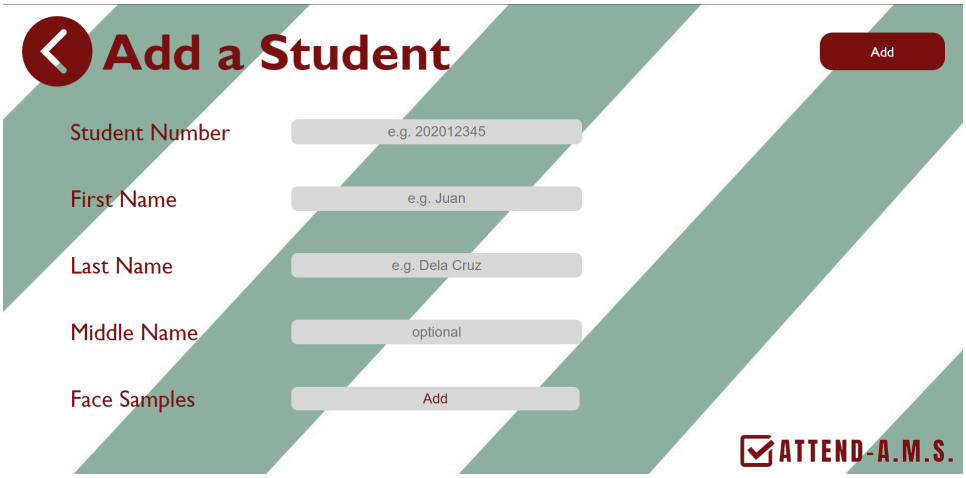
5. FIllup the details.



6. Add face samples.



7. Click the **Add** button.



D. Downloading Face Data for Model Training

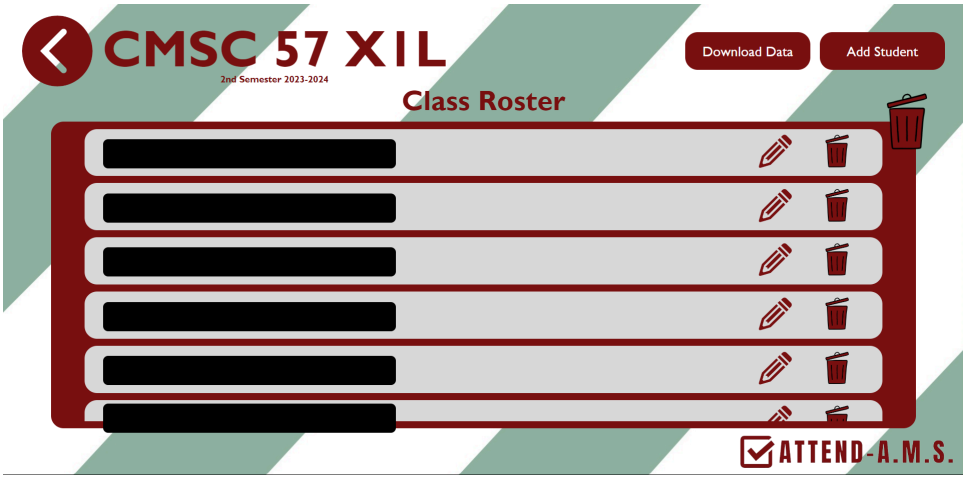
Once all students of a class are already added, you need to download the face data to be able to use the face recognition feature.

1. Assuming that you are logged in, click the **Register** button.

2. Click the **My Classes** button.

3. Select the class you want to download the face data.

4. Click **Download Data** button and save it to the models folder of the attendanceprediction folder.



E. Train model for Face Recognition
   1. Unzip the downloaded face data zip.
   2. Open the **model_training.py** in any text editor. Configure/Change the `TrainingImagePath` to the absolute path of the unzipped face data.

```python
import os
import pickle
import time

from keras.models import Sequential
from keras.layers import Convolution2D, MaxPool2D, Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator

#Specifying the folder where images are present
TrainingImagePath = 'C:/Users/ANDREAU/Documents/school/up/FOURTH/second/sp2/program/preditionbackend/models/CMSC 57 X3L 2nd Seme

folder_name = os.path.basename(TrainingImagePath)

# Specify the folder where you want to save your files
models_folder = 'C:/Users/ANDREAU/Documents/school/up/FOURTH/second/sp2/program/preditionbackend/models/'

# Save the model with the folder name as the filename in the models folder
model_filename = os.path.join(models_folder, folder_name + ".h5")

# Save the pickle file in the models folder
map_filename = os.path.join(models_folder, folder_name + "_map.pkl")

# Defining pre-processing transformations on raw images of training data
train_datagen = ImageDataGenerator(
```

3. Also change the `models_folder` to the absolute path of the model folder.

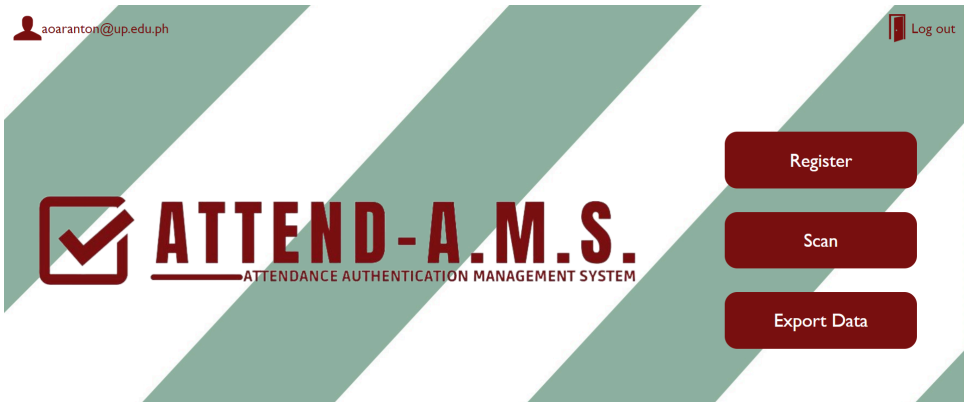
```python
import os
import pickle
import time

from keras.models import Sequential
from keras.layers import Convolution2D, MaxPool2D, Flatten, Dense
from keras.preprocessing.image import ImageDataGenerator

# Specifying the folder where images are present
TrainingImagePath = 'C:/Users/ANDREAU/Documents/school/up/FOURTH/second/sp2/program/preditionbackend/models/CMSC 57 X3L 2nd

folder_name = os.path.basename(TrainingImagePath)

# Specify the folder where you want to save your files
models_folder = 'C:/Users/ANDREAU/Documents/school/up/FOURTH/second/sp2/program/preditionbackend/models/'

# Save the model with the folder name as the filename in the models folder
model_filename = os.path.join(models_folder, folder_name + ".h5")

# Save the pickle file in the models folder
map_filename = os.path.join(models_folder, folder_name + "_map.pkl")

# Defining pre-processing transformations on raw images of training data
train_datagen = ImageDataGenerator(
```

4. Run the python script by opening a terminal in the models folder and entering **python model_training.py**.

Once the model training is finished, the face recognition feature will now be ready for use.

F. Scan Class

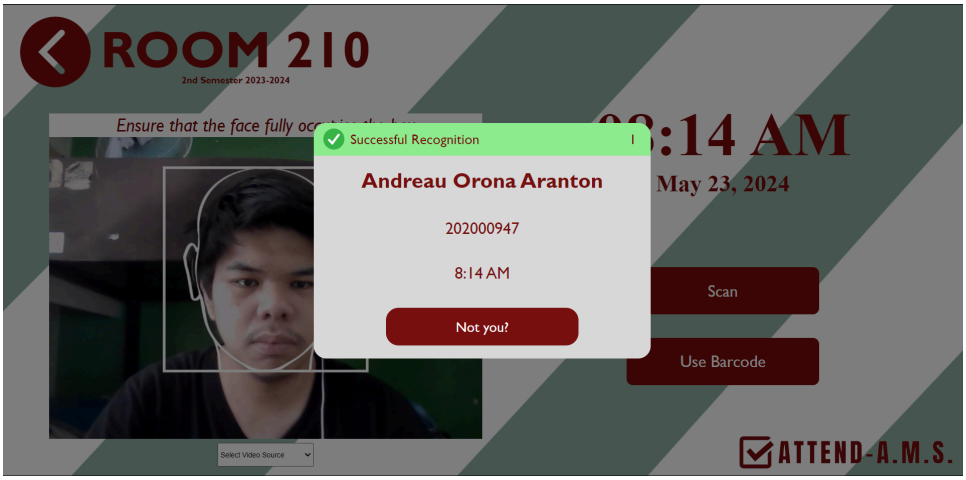
1. Assuming that you are logged in, click the **Scan** button.



2. Select the class you wish to scan to.

3. Scanning via face recognition is on by default. Align the face on the layout and click on **Scan** button to scan.

4. If the recognition is wrong, click on the **Not you?** Button. You may try to retry or use the barcode scanner feature instead.

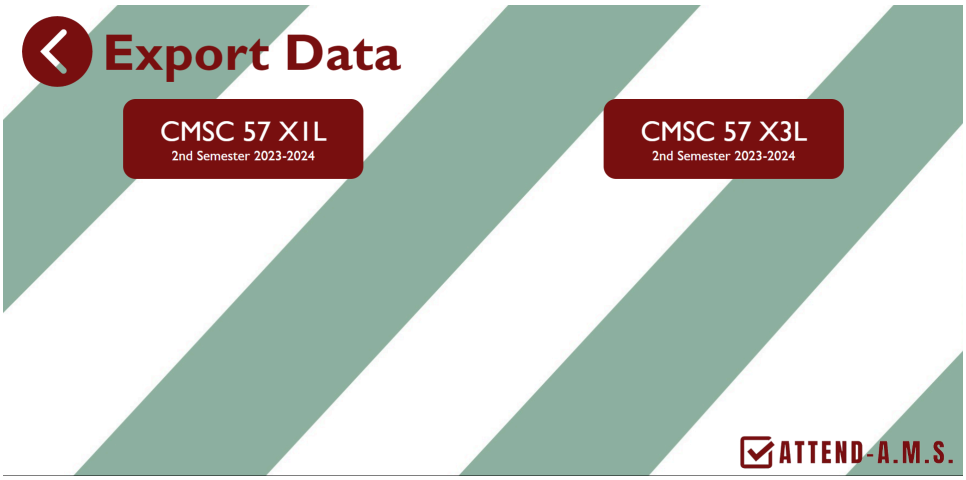


5. Should you choose the barcode scanner feature, align the barcode at the back of the I.D. to the guide and wait for the barcode to automatically be scanned.

G. Export Data

1. Assuming that you are logged in, click the **Export Data** button.

2. Select the class you wish to download data from.



3. Wait for the download to initialize and save it anywhere in the computer.