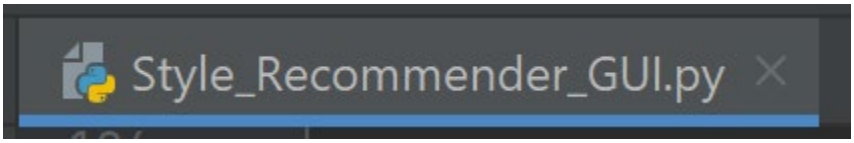
# Style Recommender App Documentation (How it works)

** Uses the similar GUI from the past batches codes (PyQt5)*

## Step 1- Launching the App

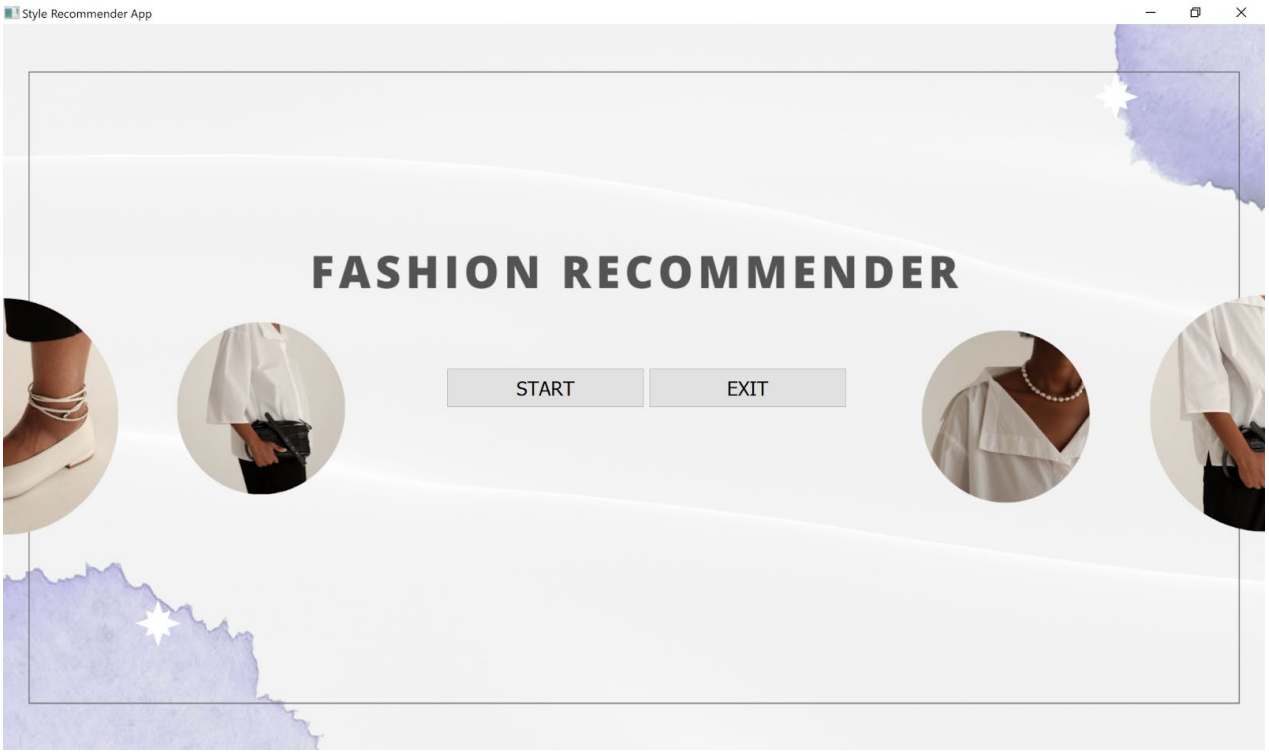
After the installation of all libraries, and setting up of the virtual environment, you can run the "Style_Recommender_GUI.py" from the project: **AI Facial Recognition (Style Recommender)** and you should see the home screen page pop up.




*If it's not showing up or you see any red lines in the codes, do install the respective packages required – through pip installing them*

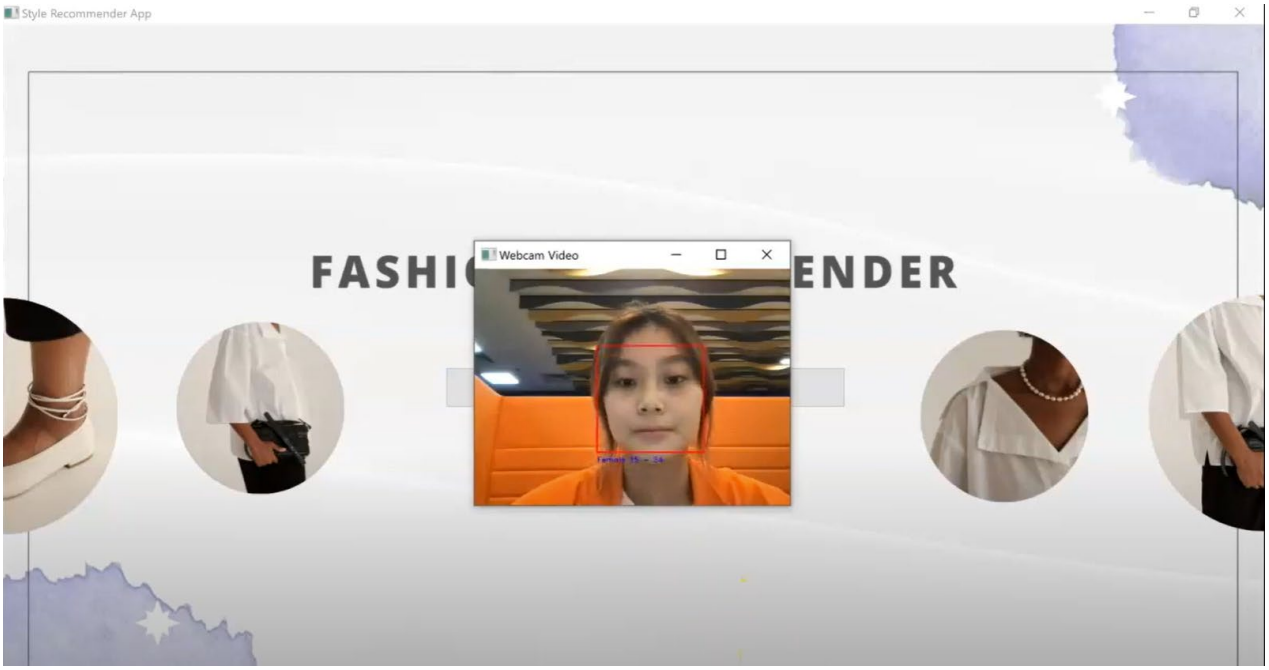
## Step 2- Click on the "Start" button on the Home Screen

Once the start button is clicked, there will be a loading screen that will pop-up. The loading screen is implemented as it takes some time for the facial recognition to load. The facial recognition might be slow for computer that is slow as well. I chose not to add the loading screen as its hardcoded and it will cause my app to crash. As u need to adjusting the timing for the loading screen to load manually. And every time the timing for facial recognition to load is different hence it will cause the app to crash more easily.




*remember to plug in your tobii pro eye detector device else you will face trouble in the next step.*

After pressing the start button, facial recognition window(face_updated.py) will pop up to detect the users age and gender. There is a timer and a counter to turn off the facial recognition window after 5 seconds. So, you do not have to do anything other than adjusting your face so that the prediction will be more accurate.

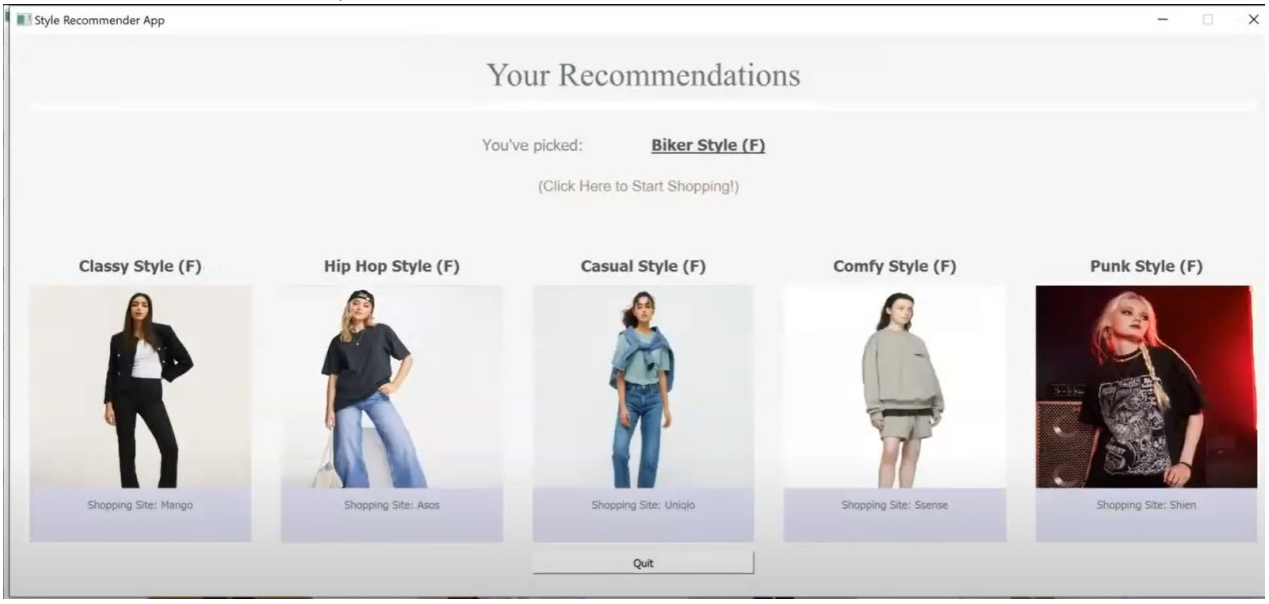



## Step 3- Gaze at the Fashion style that you are interested in:

Once the Facial Recognition feature has detected users age and gender, a randomized catalogue screen will then be shown. The tobii eye tracker device will capture your eye gaze and it will tally which Fashion style that's on screen has the highest eye gaze from your eyes and that particular style will be assigned to “most_looked”.

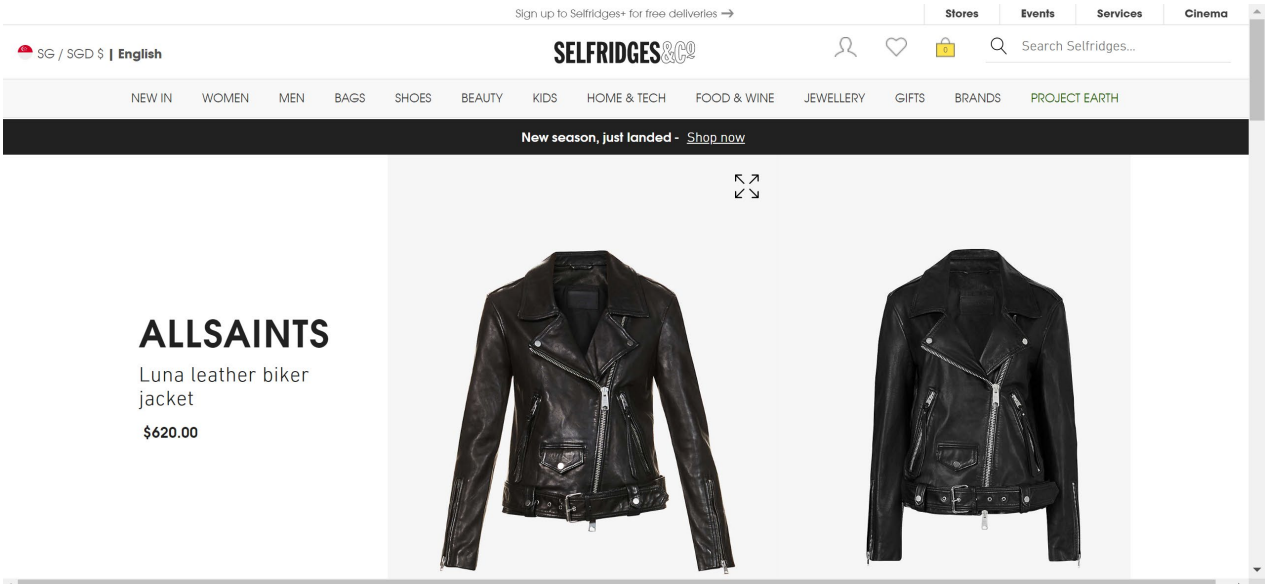
## Step 4- Recommended Fashion Styles

With the Users demographics and the chosen Fashion style from the previous step, the model chosen will be determined based on the Gender of the user detected from facial recognition. There is a total of 2 gender (female & male) that are included as a secondary level of filtering. The catalogue pages will include images of fashion style for each gender. The model will then recommend top 5 fashion style based on a group of similar users that are similar to you. (eg: they are the same age group and gender, and also pick the same fashion style, then the top 5 fashion style that are within the same gender that keeps appearing within this group will be recommended to the user)



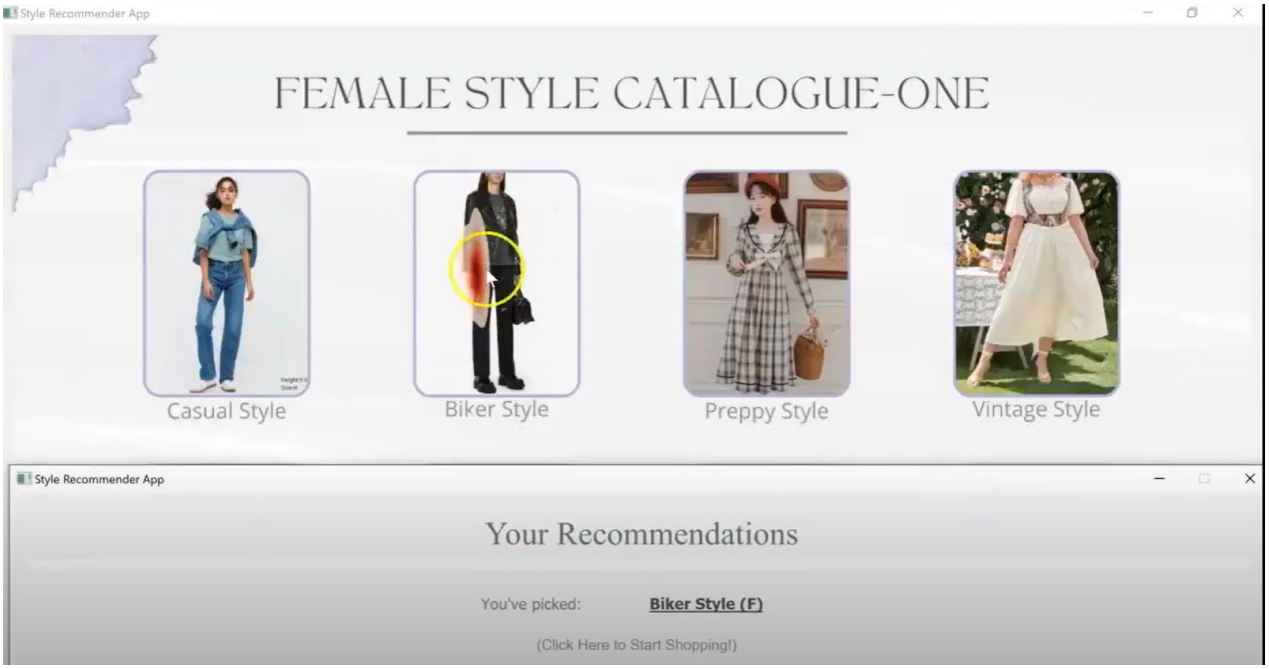

## Step 5- Click on the images or the “You’ve chosen:”

If you click on the image on the recommendation pane or the underlined chosen style name, it will redirect users to the respective shopping page. The shopping pages have more information on the fashion style where you can choose to purchase clothes of that style.

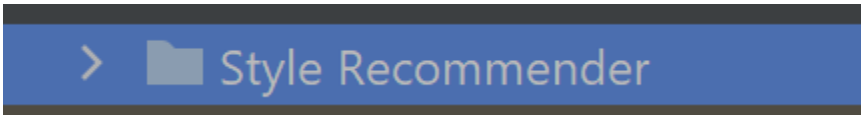



## Step 6- Heat Map

If you shift the recommendations page aside, you can see that there's a heat map beneath it. This heatmap is generated from the gaze data that was collected earlier, so you can cross check if you're looking at the “right style”.




Other Things to note:




- In the Style Recommender folder u can see Style_Recommender_GUI.py and Style_Recommender_GUI_Loading.py.
- I had implemented the loading screen for Style_Recommender_GUI_Loading.py. However, after various testing, I decided to not use the loading screen. As after testing, I found certain errors with the loading screen.
    1. Loading screen is hard coded, hence after pressing start button, u had to manually adjust the timing of the loading screen in Style_Recommender_GUI_Loading.py to make sure it fits the duration for the facial recognition to load.
    2. I encountered many times when the loading screen loads slower than the facial recognition and problems such as loading screen and facial recognition will pop up together at the same time and sometimes catalogue page will pop up as well causing the flow of the app to mess up and crash.

- Hence, I had two version of the app, one with loading screen and one without and after testing, I think the one without is better as it keeps the flow of the app.