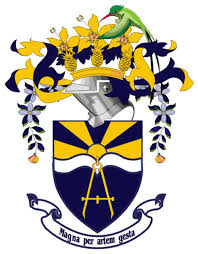
University of Technology, Jamaica



Faculty/College: Faculty of Engineering and Computing

School of Computing and Information Technology

Academic Year: 2023/2024

Semester: 1

Module: ARTIFICIAL INTELLIGENCE (CIT3006)

Title: Academic Probation Alert Python and Prolog Project

Group Members:

Hugh Scott - 1908850

Tutor: Howard James

Due Date: Nov 27, 2023

# 

***Table of Contents***

[**User Manual 3**](#_3z7raklanbp0)

[Figure 1.0 3](#_3zsfsu8moqfv)

[Step 1: Getting dummy records to test 3](#_y59f7g9pflf1)

[Step 2: Autochecking 4](#_q53znjt1ce14)

[Step 3: Manual Check 4](#_exyq7qprtugs)

[**System Design 5**](#_cbnob2jlqtam)

[Database Tables: 5](#_uijver4njki9)

[Functionally 6](#_nfcy8zuxrdh4)

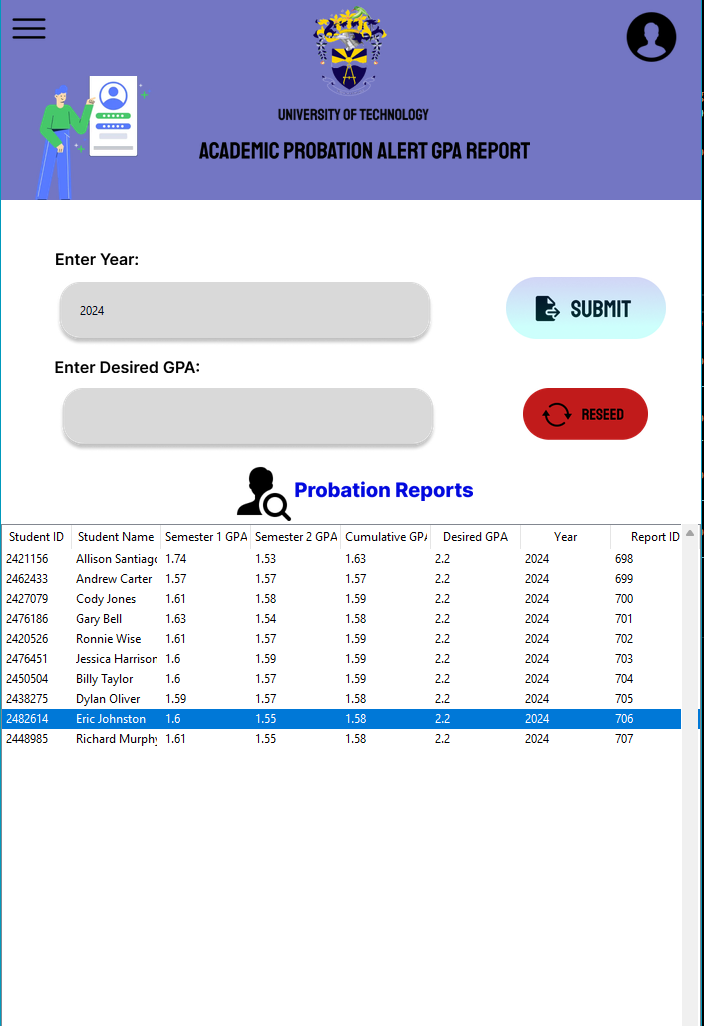
[Dependencies: 6](#_bv8jq21qy6va)

[Steps to setup the app: 7](#_gnjeqatcyi6q)

[**Project Group Report 8**](#_gnj4ygp9asll)

# 

# User Manual



## *Figure 1.0*

## Step 1: Getting dummy records to test

To get started you can seed the database with the red button, it removes all old data and adds in a fresh set of 50 students into the database. Each student will have the relevant courses. For the purposes of space for this demo the years that are supported are 2019 to 2024.

## Step 2: Autochecking

When the program is launched the system will have the records of everyone who was previously checked displayed if none are in the database none will be displayed on launch. The app will also check everyone below the default threshold every 5 mins. It will send an email to all relevant persons if that student falls below the default.

## Step 3: Manual Check

If the user wants to check a year manual they can enter it into the field along with a desired GPA. If no desired GPA is entered then it will default to 2.2. If the input in any of the fields are unsupported it will prompt the user to enter a valid one. After entering them click the submit button and wait for the records to update. The table will update with all the relevant students and send out an email for each one.

# 

# System Design

## Database Tables:

This is an example shown at ***Figure 1*** at the beginning of the document. The following are the database tables and example data.

Module Details (**module\_details**)

| module | year | semester | student\_id | grade\_points |
| --- | --- | --- | --- | --- |
| PHY28 | 2024 | 1 | 2482614 | 1.14 |

Module Master(**module\_master**)

| module | number\_of\_credits |
| --- | --- |
| PHY28 | 3 |

Probation(**probation**)

| student\_id | student\_name | **semester1\_gpa** | **semester2\_gpa** | **cumlative\_gpa** | **desired\_gpa** | **year** | **report\_id** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2482614 | Eric Johnston | 1.6 | 1.55 | 1.58 | 2.2 | 2024 | 706 |

Student Master(**student\_master**)

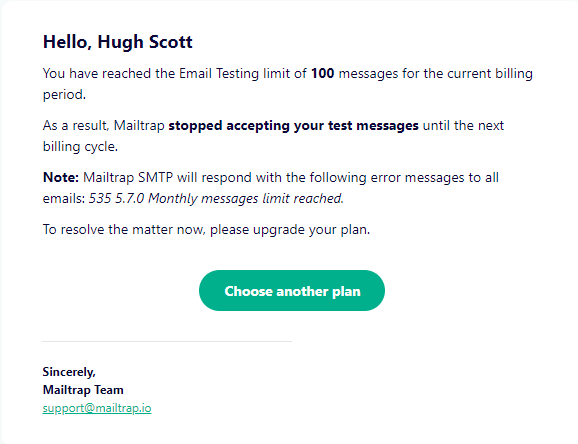
| student\_name | student\_email | school | programme |
| --- | --- | --- | --- |
| Eric Johnston | ericjohnston@mail.com | Utech | BUSI |

## Functionally

The system was designed to be a Report generator for Academic Probation Alerts. It needs the year from the user along with the desired GPA. The app then gets the year and queries the **module\_details** sql table. It will retrieve all the relevant columns with the year entered. The **module\_details** carries the modules, year, semester, student\_id and grade points for each module. The ***student\_id*** field is used to get the student information and the module field is used to get the credits related to each module.  
 With all the information loaded into the app it sorts each person by ID, assigns their modules, grades, semester and credit for each module in order. Then it sends that to the **calculate\_gpa(grades, credits, semesters, modules)** function. The **calculate\_gpa** works by getting the python data types looping over each array of values passed and adding them in as rules to the prolog code.

The prolog will take all the raw grades and calculate the semester 1, semester 2 and cumulative GPA. (It can calculate 3 semesters but the database will only save 2 at this time). Each output from prolog gets checked to see if the cumulative GPA is equal to or less than the desired GPA. If the statement is true then it will add that persons data to the **probation** database table. The probation database table has the fields

## Dependencies:

**Mailhog** was used to facilitate the large influx of emails. Below is the mailtrap message explaining that I had used too many messages.

In the project folder there is a file called requirements.txt, running pip install -r requirements.txt will install all the necessary dependencies for the project to work. The database sql file named ai.sql is in the assets folder and can be ran on XAMPP or any other sql client. Also in the assets folder is the **mailhog** exe, it needs to be running to see the emails. After clicking it open a tab at <http://localhost:8025/#>. Finally, the .env file holds the environment variables such as the database credentials and the mailhog credentials. The mailhog credentials are SMTP\_SERVER="localhost" & SMTP\_PORT=1025. Your database credentials may be different but the database should be DATABASE="ai", change the PATH\_TO\_PROJECT\_ASSETS\_FRAME0 to your computers path to the assets/frame0 folder.

## Steps to setup the app:

1. Import the sql file
2. Use the .env file to add your database credentials and save it
3. Run the mailhog .exe file and go to <http://localhost:8025/#>
4. Run the command pip install -r requirements.txt in the root of the folder
5. Go to the index.py file in the root of the folder
6. Run it
7. 👍🏾

# 

# Project Group Report

Due to the fact that I couldn’t find a group, this project was worked on by only me. All the implemented code was done by Hugh Scott 1908850. I would like to extend my heartfelt thanks for your exceptional teaching throughout this year and your continued patience and understanding with the Prolog project. Your guidance and expertise have been invaluable.