**Purpose of Website**

The purpose of this website is to create a test score management system where the student can enrol in modules and add scores that they have achieved in modules. To do this, I have implemented flask-login to store the session of each user. Also, flask-login has been used to authenticate each user that tries to log into the application. The unauthenticated users and authenticated users also have different views in the application. I have also used flask-admin to create, delete and view each of the Student, Module and Scores table in the database.

**Link to deployed website**

https://dromoxyz.pythonanywhere.com/

Username: dlartey

Password: lartey123

**Website Analysis**

**Web Forms**

This is implemented using Flask-WTF

* StudentForm (for student registration)
* Flask-login form (To log student in)
* ModuleForm (for selecting which modules the student takes)
* ScoresForm (for student to add their scores for each module)
* PasswordForm (to allow students to change their password)

All of the web forms listed above have the DataRequired validator, meaning that if information isn’t entered into each textbox the form will not submit.

**Database**

This has been implemented using sqlite3 and Flask-Sqlalchemy

* Student database (To store details about each student)
* Module database (To store details about each module)
* Scores database (to store details about each test the user takes)
* Enrolment table (Many to many relationships between students and modules)

**Sessions/Cookies**

* Flask-login uses sessions to store which user is currently logged in. This is very useful as I can display a message on the dashboard relating to which user is active. Also, when I show the user their modules and scores, sessions are useful as I can grab the current user ID and display the scores and modules that correspond to the current user.
* When the user clicks logout, their session is deleted

**Authentication**

* Checking that when a user registers, they don’t take a username that is already being used
* Making sure that a user logs in with the correct username and password

**Appropriate Styling**

* Bootstrap has been used to make website responsive

**Unit tests**

* I have written unit tests to ensure that when a specific page is loaded, the test data matches the expected output

**Logging**

* Logging has been implemented to ensure that whenever a user logs in, logs out or registers, that data is saved in ‘example.log’ file.

**Bootstrap & JQuery**

* Bootstrap has been implemented to ensure that the website is responsive no matter the screen resolution that the user has.

**Evaluation**

I have decided to keep the links to the different pages in the navbar so that it is presented in an ordered way. This will improve the user experience as the user will know how to navigate the page and all the links are grouped together. Furthermore, an alert message displays whenever a user enters incorrect details or creates a test score. This will improve the user experience, as the user will now have feedback as to what happened whenever they click a button. This also helps the user to navigate the website.

Also, I have decided to separate the website into two sections; those who are logged in, and those who aren’t. This will help to improve the user experience, so that in the navbar the links that are available to registered users aren’t visible to unregistered users. By decomposing the website like this, it will reduce the content on the website and make it easier to understand what is happening.

On the dashboard page, when the user completes authentication, I have displayed their name and a message showing that they have logged in to make it clear to the user what is happening. Furthermore, links to all of the pages have been displayed in the dashboard with text over each button. This will improve the user experience, as the use can map what each button does, and what purpose it serves. Headings have also been included to divide the buttons that link to different pages into different groups.

Finally, I have decided to make the tables for showing the student enrolled modules and student test scores dark. This has been chosen to that the black and white colour scheme of the website can be adhered to. This will improve the user experience as having a consistent colour scheme across all of the pages in the website creates a design pattern that will make the website more user friendly.

**Potential security issues**

A potential security issue with my application was regarding session management. To ensure that the authentication system could not be easily exploited, I used flask-login which uses sessions to authenticate each user instead of making my own authentication system. Also, for all the forms that are used in my website, flask-wtf has been used which provides CSRF protection and validation.

Another potential security issue is in the event that the database is compromised; I have used password hashes. This has been implemented using werkzeug.security. Therefore, this will stop sensitive data from being leaked as the passwords have been encrypted.

Finally, a security issue is regarding website views. I don’t want unauthenticated users to access view that are restricted to only authenticated users. Therefore, I am using flask-login to implement @login\_required to access certain routes in the website that can only be accessed by authenticated users.