

READ ME

IDE

The application may be run from the terminal in a Python IDE such as MS Visual Studio.

Initialising the Database

Before running the application for the first time it is important to initialise the database.

This can be achieved by executing the following command in the terminal ***flask --app flaskr init-db***

Figure 1 - Initialising the Database

```
PS D:\onedrive\1. University of Essex\4.0 Secure Software Development\flask\flask-tutorial> flask --app flaskr init-db
Initialized the database.
PS D:\onedrive\1. University of Essex\4.0 Secure Software Development\flask\flask-tutorial> 
```

The message, 'Initialized the Database' will be displayed in the terminal.

Running the Application

The application may be run from the terminal in a Python IDE such as MS Visual Studio. The application can be run using the command ***flask --app flaskr run --***

Figure 2 - Running the Application

```
PS D:\onedrive\1. University of Essex\4.0 Secure Software Development\flask\flask-tutorial> flask --app flaskr run --
* Serving Flask app 'flaskr'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit

```

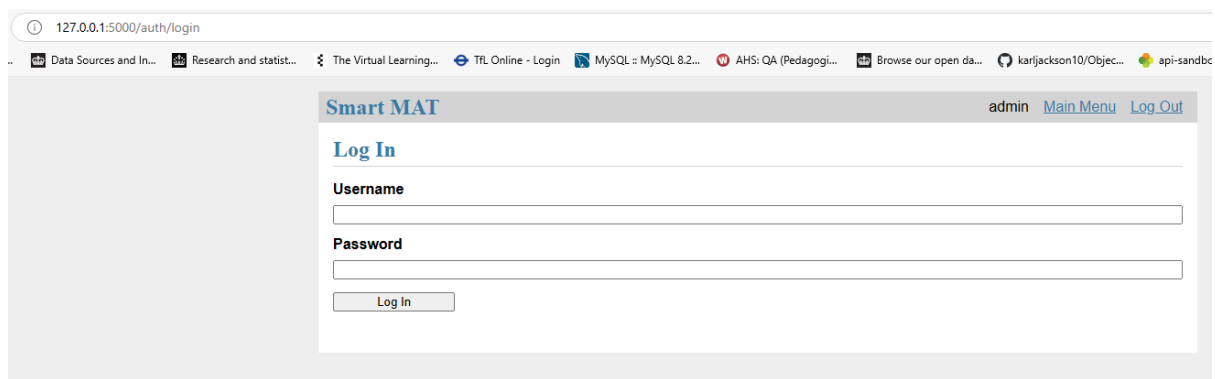
Using a Web-Browser

The application can then be viewed via a web-browser at ***127.0.0.1:5000/auth/login*** or using the link below:

[Log In - Smart MAT](#)

This provides access to the login page

Figure 3 - Login Page



127.0.0.1:5000/auth/login

Smart MAT

admin Main Menu Log Out

Log In

Username

Password

Log In

Initial Setting

The following default users have been created

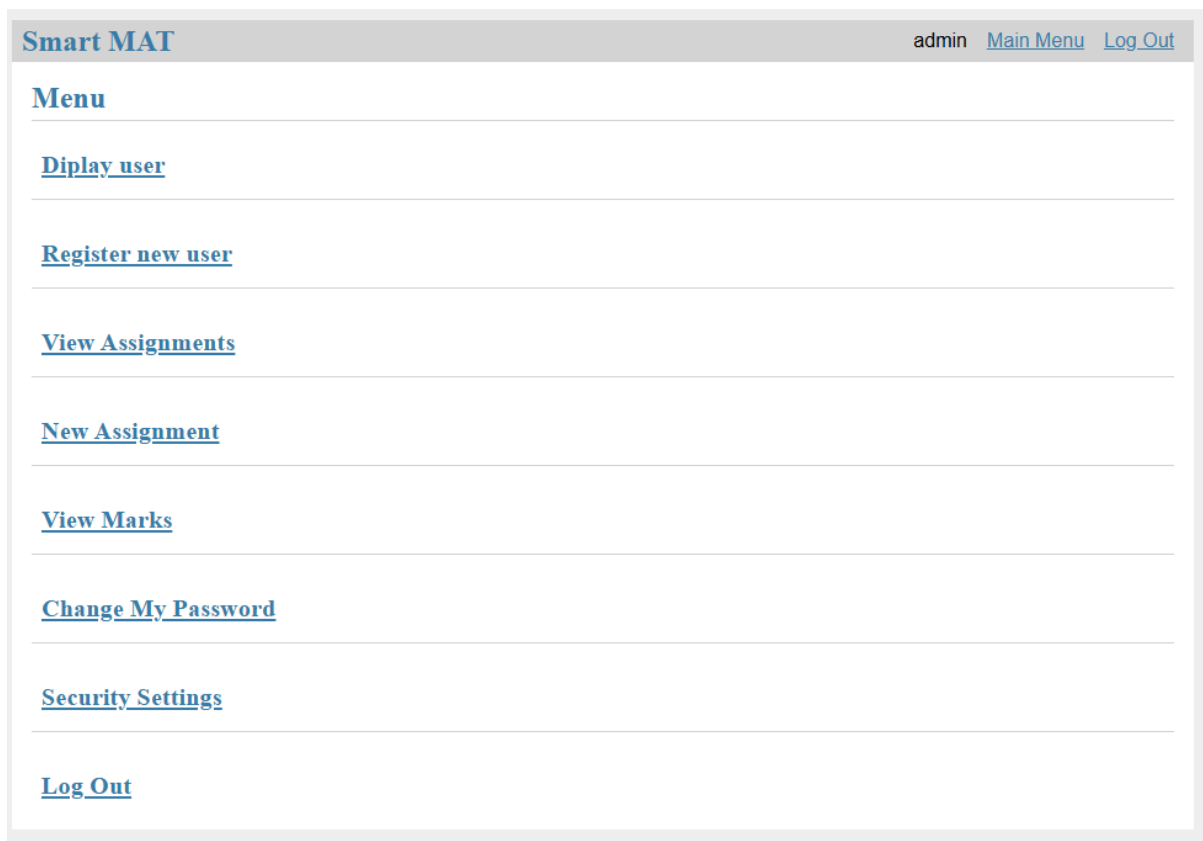
Figure 4 - Default Users

User	Password	Role
admin	123	Admin
student1	123	Student

Main Menu

When logged in, users have access to a range of options relevant to their role, from the Main Menu

Figure 5 - Main Menu (Admin View)



Application Programming Interface (API)

The system contains an Application Programming Interface (API). This can only be accessed by a user with Admin permissions. The API can be used to perform the following tasks:

Address	Task Description
127.0.0.1:5000/api/users/	List each user with their username and role
127.0.0.1:5000/api/users/?id=id Or 127.0.0.1:5000/api/user/id	List a specific user with the 'id' parameter

127.0.0.1:5000/api/users/?role=role Or 127.0.0.1:5000/api/user_r/role	List all users with a specified role using the 'role' parameter
127.0.0.1:5000/api/user_add/?username=username&password=password&role=role/	Adds a user to the system with the username, password and role specified as arguments
127.0.0.1:5000/api/user_delete/?id=id/	Deletes a user from the system using the specified user id from the id argument