**8.2 Outcome of requirements exercise**

Listing of requirements proves to be of utmost importance as a means of making sure that the system was implemented correctly. Upon utilisation of the system, it is important to keep in mind that there exist three main roles with regards to users:

* The secretary, being the administrator responsible for the management of students and supervisors, along with the performance of allocations and publishing of results.
* The supervisors, being lecturers that submits areas for final year project having as a primary overseer.
* The student, being able to submit a preference form which will save data to be later employed as a means of allocating students to supervisors.

The system has pages segregated by three authorizations. Firstly, the pages made available to supervisors. The pages that any given supervisor may access consist of, a page to submit areas, being accessible from the ‘Staff’ section located in the header of each page. The supervisor is required to choose this option as a means of adding any areas to be supervised by the given lecturer. After the allocation has taken place, the supervisor assumes the role of supervising the area inputted and the student allocated to that area.

When filling in the area submission form, the following are the fields made available:

* Title/Area
* Co-Supervisor (if any)
* Main Subject Area/s (keywords)
* Brief Project Description
* Resources required
* Recommended prerequisites
* Foreseeable ethical issues
* Area quota

The area quota is mainly used to see how many students the supervisor can accept according for the given area. The quota can even be left as zero if the supervisor does not want to set a limit for each particular area, and therefore take students in accordance with a global supervisor quota.

Secondly, pages made available to students. Each student is required to login to the system with the necessary login credentials. If the student is logged in successfully, there exists an option to view all available areas submitted by the supervisors. The student is then required to read through all these areas as a means of making a more informed decision with regards to the preferences to be chosen.

When the faculty decides to begin accepting FYP submissions, students must fill in a final year project preference form known as ‘FormA’. The student is not allowed to choose the same supervisor or area more than once. It is compulsory for all students to choose preferences and submit each respective form to be able to perform allocations.

Lastly, is the administrative role of the FYP allocation system. The features made available to a given administrator include, the importation of supervisor and student data, being vital as to enable the functionality of the system. This effects functionality both in terms of student preferences and supervisor area submissions. When the administrator is ready from importing the necessary files, students and supervisors may utilise the system within appropriated time frames, across any given scholastic year.

Once the allocation is performed, all student preferences together with the time at which forms were submitted are taken into consideration. The allocation happens according to preference. To reiterate an instance given in a previous section, if two students have the same preference, the student with the higher average mark is allocated first. If both students have the same average mark, then the area will be allocated to the student who submitted the form first. If there exist students that have not been allocated to any supervisor, the administrator be made able to allocate these students by being able to perform allocations manually.

**8.3 Overview of Tools**

**8.3.1 System hardware and software**

The FYP Allocation system requires no specialised hardware, however, from a software standpoint, in order for the system to function a running database hosted on a server is required. Moreover, for the users to be able to utilise the system, data must be stored within a hosted database. Since the system is web-based and therefore will be running online, there internet connectivity is a must. Upon accessing the system, users must be in possession of university email address, and therefore must be enrolled within a course at the University of Malta. In order to gain access to the web, a browser is essential in order to reach the domain hosting the system. Microsoft SQL Server Management Studio (MSSMS) proved to be a vital piece of software that greatly benefitted in the construction of the entire schema for the system.

**8.3.2 Technologies, protocols, external libraries, and reused/reusable components**

The main technologies used in the FYP Allocations system include:

* Microsoft SQL (MSSQL)
* Microsoft SQL Server Management Studio (MSSMS)
* Visual Studio 2019
* Microsoft excel
* .NET Core
* Entity Framework Core
* ASP.Net MVC.

The ASP .NET MVC (Model View Controller) is a framework that provides developers with the ability to manipulate auto generated classes of code, whilst maintaining the core functionality of the existing framework. In MVC, the model is responsible for all the instantiation of data used by the system. Views output the contents through web scripting languages. Controllers are responsible for dealing with incoming requests and user interaction with the system. Microsoft excel was utilised for the compilation of imported data for students and supervisors, whilst also being used to export data to ensure correctness before allocations are performed. Important to note is that Microsoft excel was also utilised as means of storing test data such as sampled areas for supervisors.

MSSMS is a management service that serves to manage, configure, and develop SQL based databases. Entity Framework allows for the utilisation of Object-relational-mapping (ORM), meaning that any given database may be manipulated through a code first approach. Both Visual Studio 2019 and MSSMS are maintained by Microsoft, meaning that both systems benefit from support that ensures optimal communication between the two tools.

With regards to the external libraries, ASPNetCore Identity was employed as part of the system. This is a package that supports the auto configuration of login and registry-based UI, whilst also providing a variety of account management related features, including but not limited to the capability of segregating users into defined roles. For this reason, to be able to perform the login which allocates users to respective roles, being either student, supervisor, or administrator, a dependency on email address exists. Therefore, ASPNetCore Identity proved to be ideal for such a purpose. Another package, responsible for delivering results of FYP Allocation was the ASP SMTP (Simple Mail Transfer Protocol) Client. The SMTP Client enables sending of email messages in a secure manner via an external SMTP server. The main use for the ASP mailer was after the allocation had been performed and every student is assigned to a supervisor. The mailer is used to send the results to the respective students, therefore, if some students remain unassigned, results may not be sent.

As a reused component, the system followed a design pattern similar to the University of Malta website. This included the primary colour scheme and the styling of CSS (Cascading Style Sheet) elements. Since the system would be potentially utilised by the University of Malta, it was decided to add such a design pattern similarity as a requirement and therefore an attempt was made to render the frontend as similar as possible to [reference here to um.edu.mt]

**8.4 User Interface Design**

The user interface created was targeted towards the aforementioned roles that pertain to the University of Malta. Therefore, as previously stated, the design was developed to be as closely related to the University of Malta UI as feasibly possible.

**8.4.1 Login**

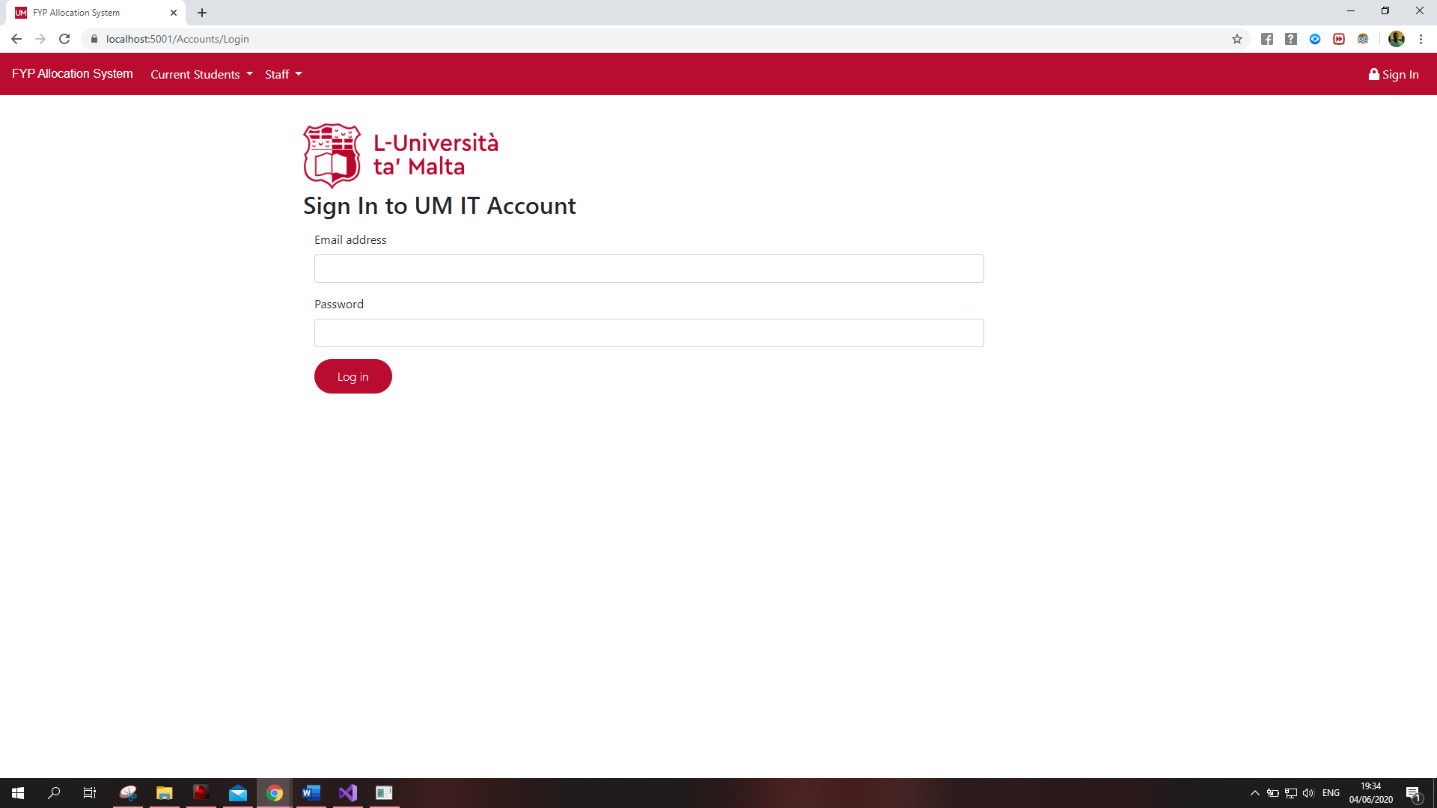


Figure : Login Page for users

To begin with, each user that may access the system will have to login to be able to view the content made available to the given user. Each user is in possession of unique credentials, being the email and password provided by IT services at the University of Malta. Important to note is that in this case, user credentials were seeded due to a development environment. Therefore, if the system were to go live, the targeted credentials would be implemented into the core system functionality. If credentials inputted are validated, and found to be correct, the system will sign the user and provide with access to page targeted towards the respective role held by the user.

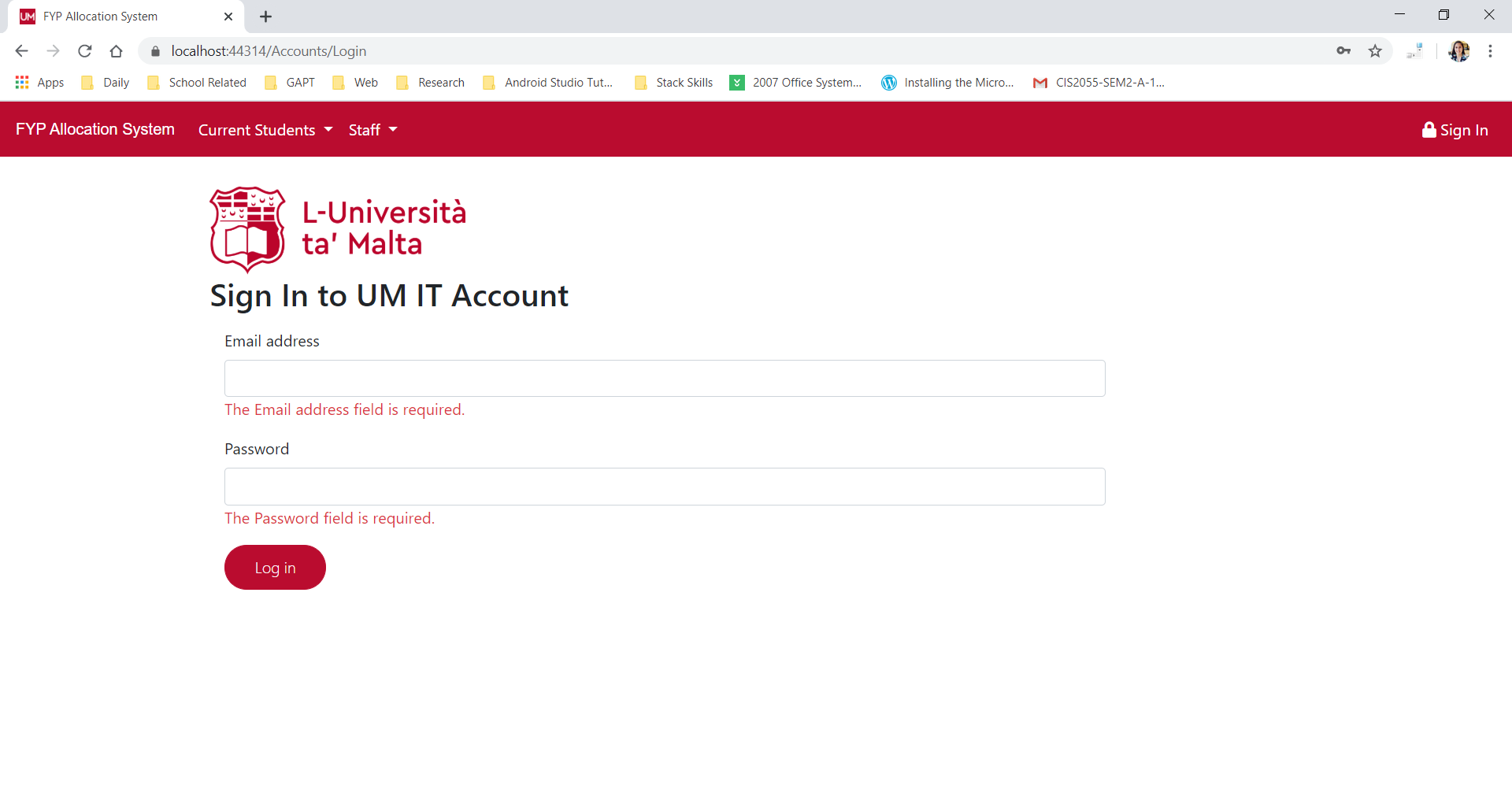


Figure : Login validation upon entering no data into each field

If the user does not input anything into these fields and tries to perform a login, the user will be prompted with the above prompts to make sure both fields are not empty before reattempting sign in.

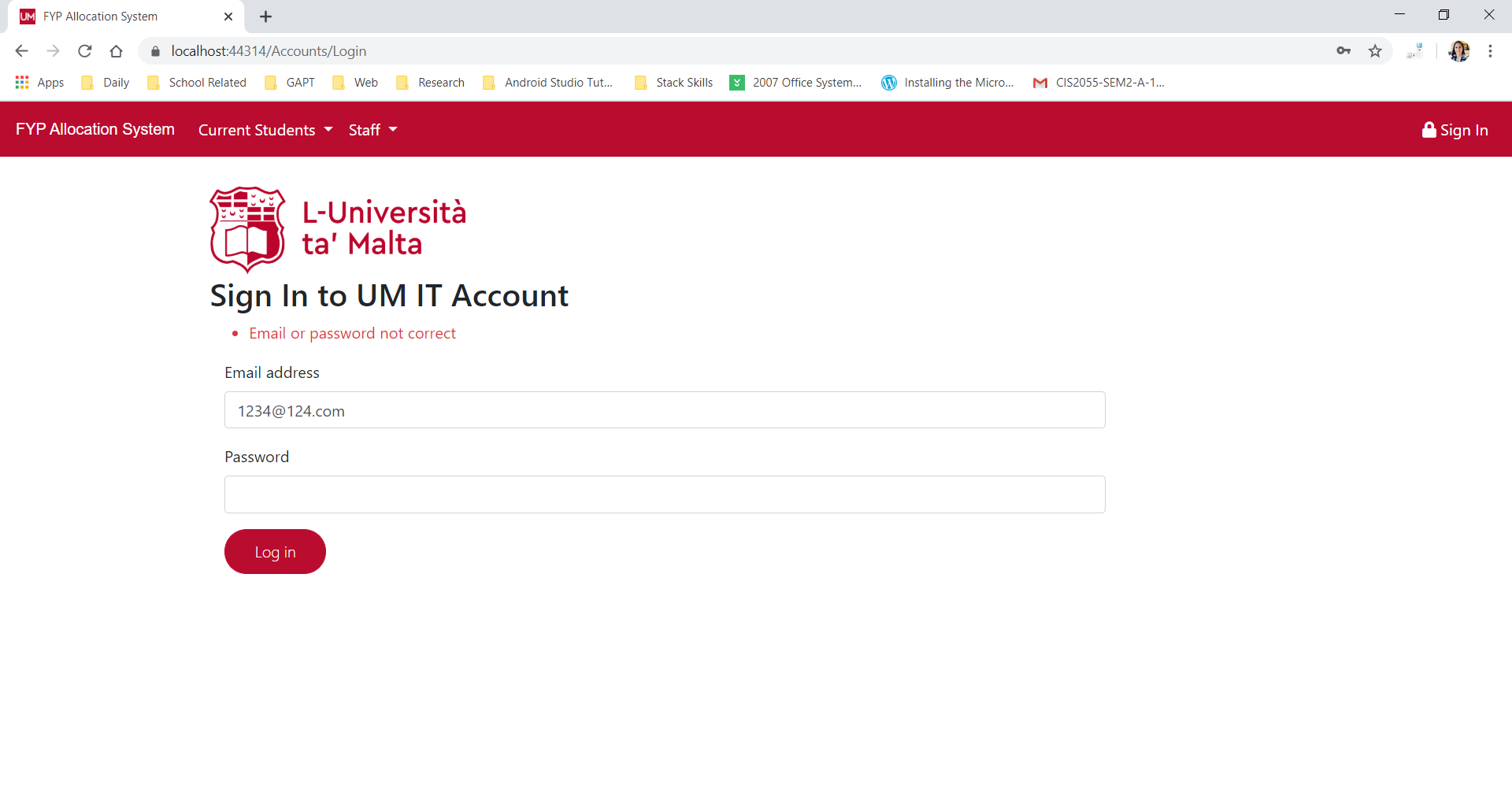


Figure : Error message upon entering incorrect credentials

Also, users will be prompted with an error message if either of the fields for sign in are incorrect. This message will inform users that another attempt must be made, this time using correct credentials. Once this message is prompted, the password field will be automatically cleared to ensure proper verification.

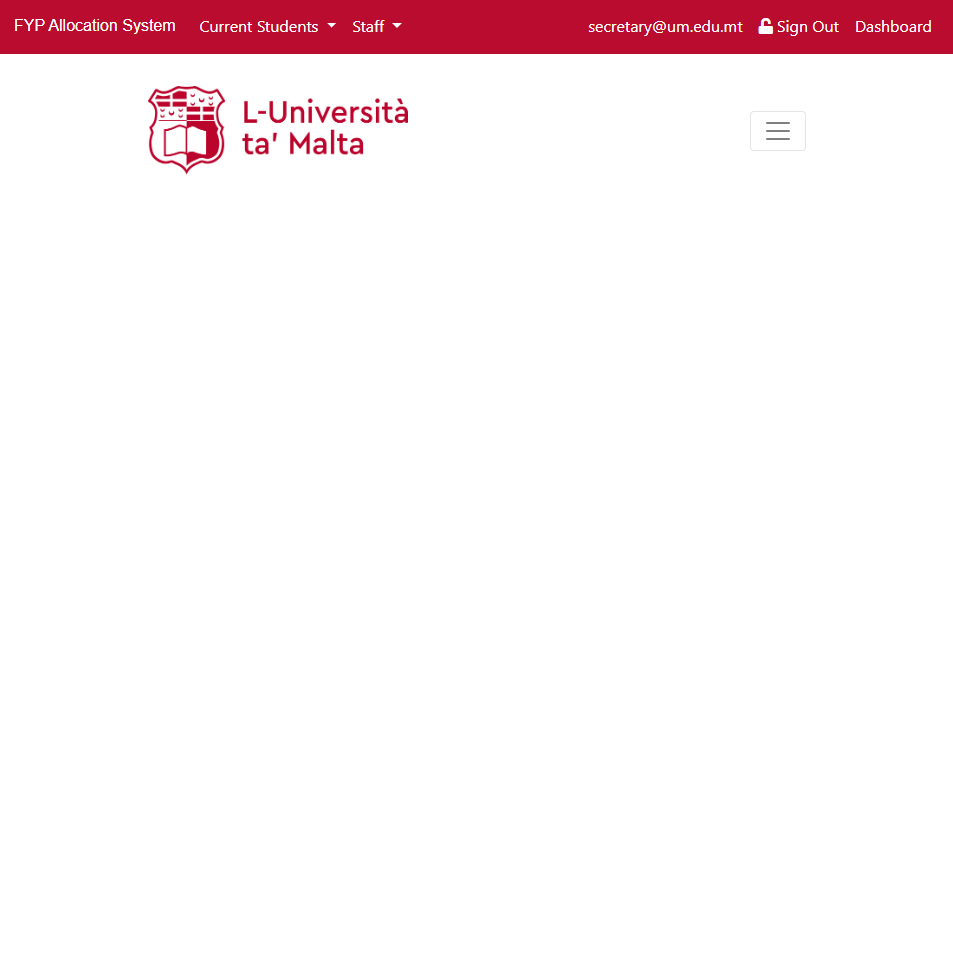


Figure : Altered header to show administrator details and options relating to signed in administrators



Figure Altered header to show supervisor details and options relating to signed in administrators



Figure Altered header to show student details

This is the display for each role after login. Email is displayed in the top navigation bar on the right. With an additional dashboard or profile option for administrators and supervisors, respectively.

**8.4.2 Administrator Role**

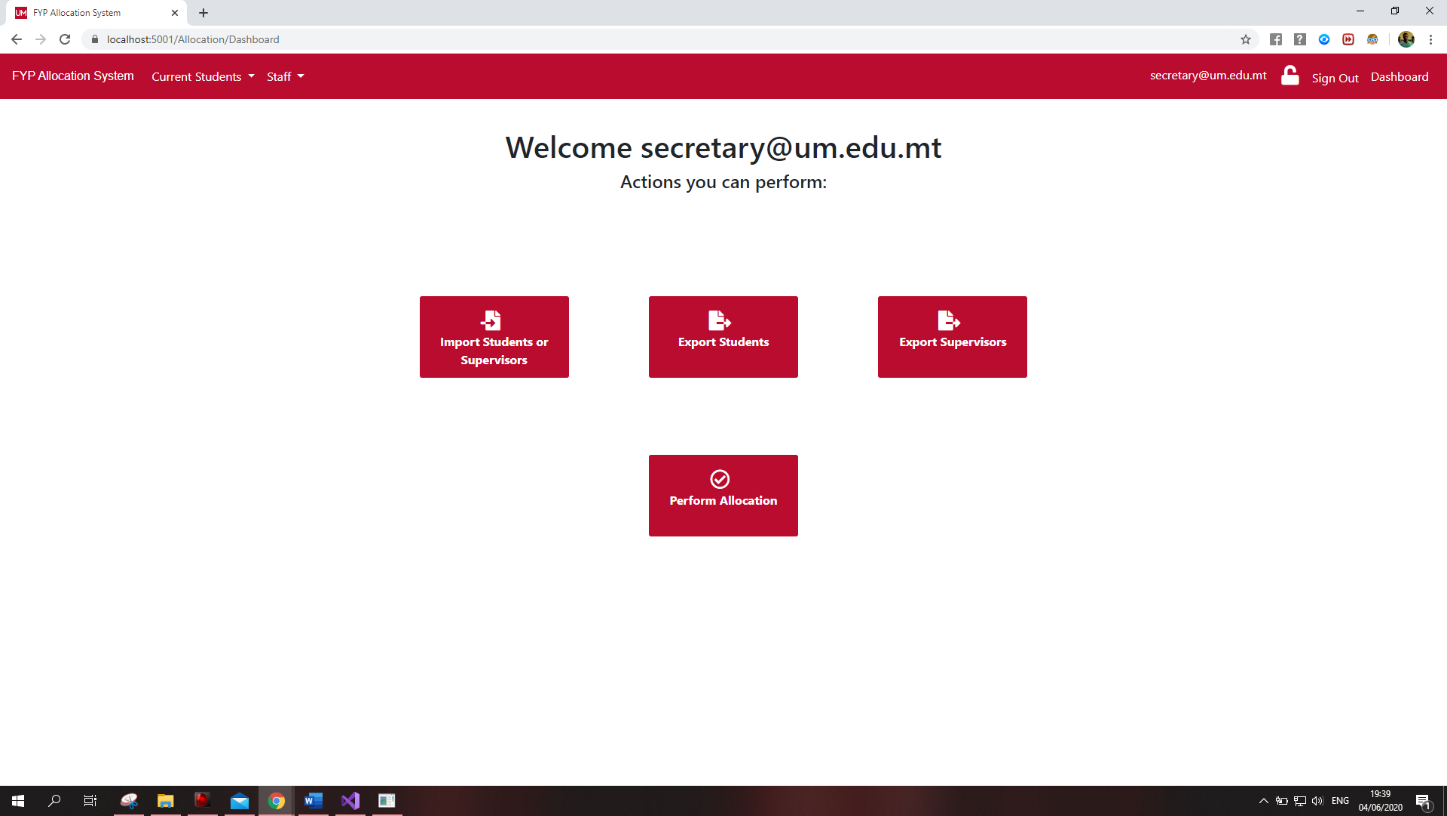


Figure 7: Administrator Dashboard

When the administrator is logged into the system, the dashboard area must be accessed, containing the main actions pertaining to this role. In the dashboard area, the administrator can import a list of enrolled students or employed supervisors, together with the exportation of both supervisors and students to ensure correctness of data. Also, the ‘perform allocation’ option may be selected to access all functionality relation to the performance of allocations

Importing of Students and Supervisors

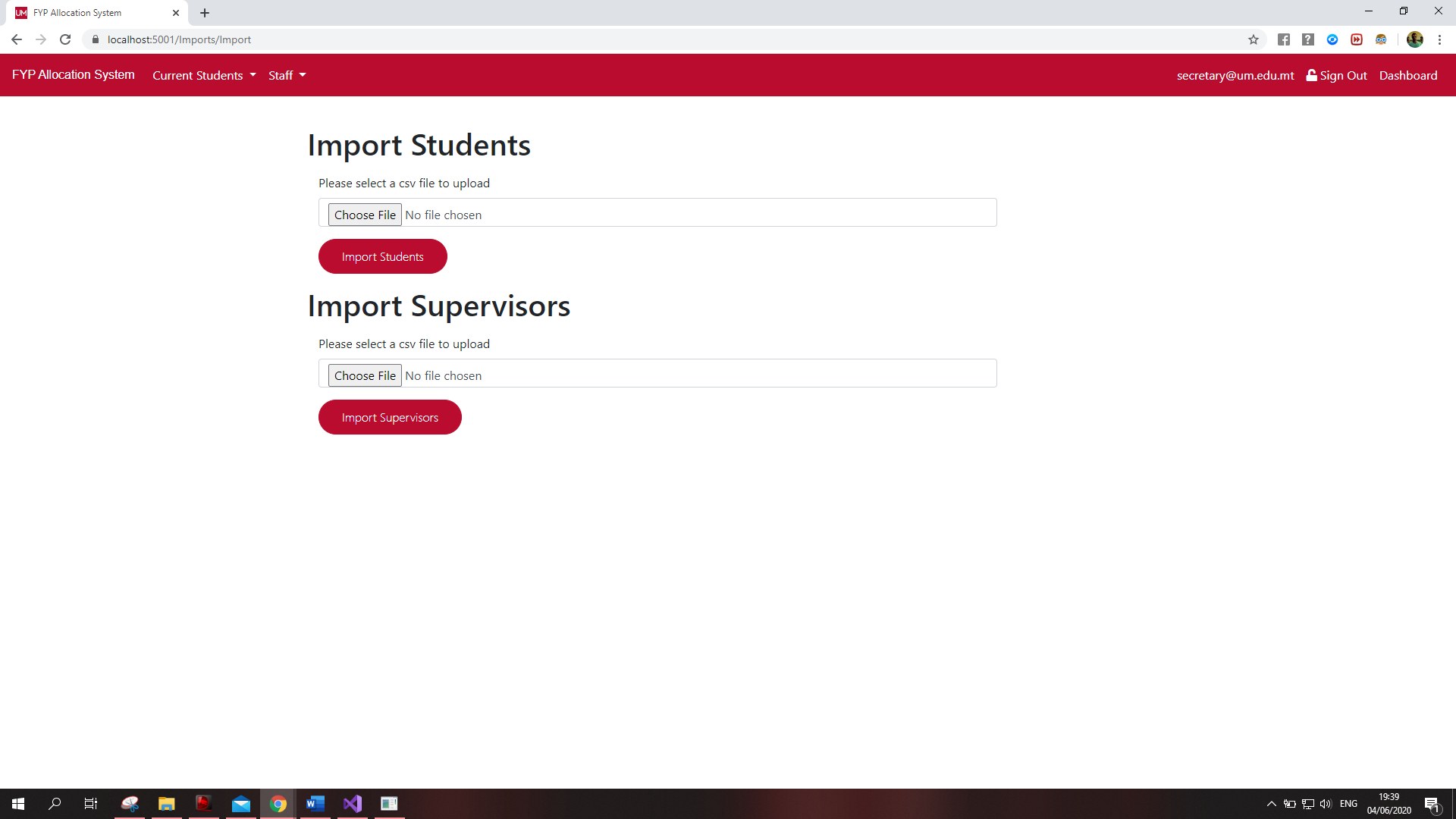


Figure : Importation of enrolled students or employed supervisors

In the ‘import students or supervisors’ option, the administrator is able to import the students or supervisors, in order to enable system functionality. To be able to import data, .csv (comma separate value) files are required as means of correctly storing data in the database. Also, it is important to mention that due to the nature of storage, files must not contain column header as to prevent any unprecedented faults. Appropriate validation has been added for the instance of headers existing withing the .csv files.

Exporting of Students and Supervisors

With regards to the options for Exportation of students or supervisors, users are given the ability to be able to download all the content of the students or supervisors, with each respective option available. This allows for assurance of correctness in order to prevent the case of students or supervisors being omitted or entered incorrectly into the system.

Performing the Allocation **Change Image to proper one during testing**

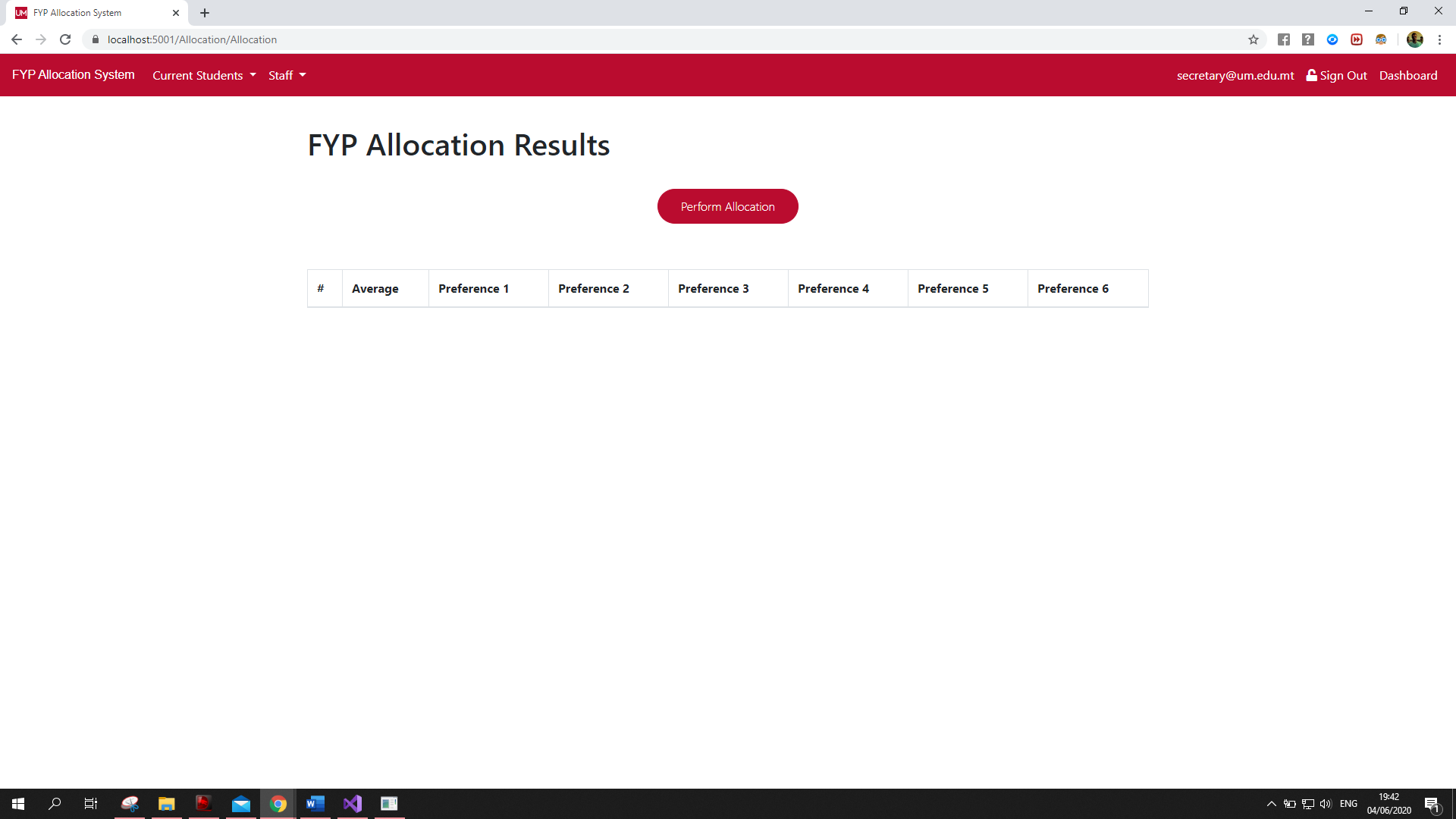


Figure 9: Perform Allocation Page with no allocation performed yet

**NEED:**

**Image before all preference are entered and after all preferences are entered**

The above image is the page within which allocation is performed. All students and supervisors must have been imported, along with the existence of preferences submission for each student, in order the ‘perform allocation’ button to become available.

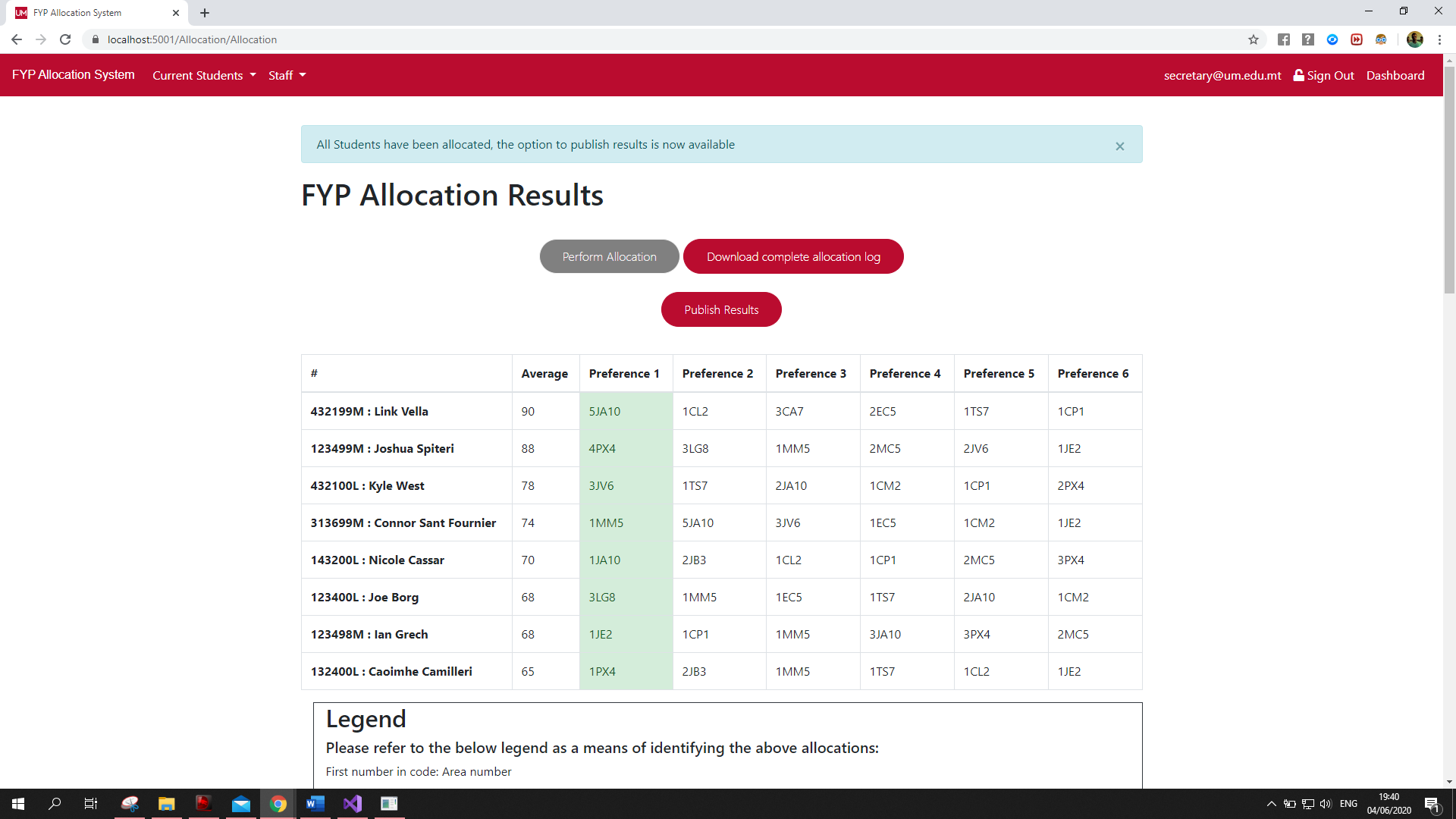


Figure : Allocation page after all students are successfully allocated

**8.4.3 Supervisor Role**

**8.4.4 Student Role**

**8.4.5 Important Validations**

Talk about, students FormA, Supervisor Preferences, And the conditions for allocation to be performed, all students have to submit preference etc…

**8.5 Non-functional properties**

**8.5.1 Usability considerations**

When taking usability into consideration, although functionality may be optimal, priority is to be appointed to the aesthetic value of the interface. This is due to the fact that, although the system may perform all desired functions, any given user may still be unable to properly navigate the available features due to a poorly designed UI. The main goal therefore was to have a system that boasts ease of use and understandability. As a result, such a system may be correctly utilised a varying range of demographics, as the system would subsequently pertains to an appropriate construction, not requiring any prior knowledge to be properly utilised.

An important notion to be pushed forth with regards to colour scheme, is that apart from the evident similarity to the University of Malta colour scheme, the UI is kept consistent throughout. This ensures that styling employed does not result in any long period strain and is therefore sufficient for visual comfort. This results in the user focusing more on performing tasks, rather than being distracted by inconveniences.

Whenever the completion of a task is to be accompanied with redirection, the system is appropriately designed to perform such traversals. This is to avoid users being rendered confused as to a lack of changing data following any tasks performed.

By potentially logging in using the University of Malta credentials, in this instance being developmental seeds made to mimic such credentials. Data on the respective user can be provided, without the need for registration, due to each user already having an account by default upon enrolling at the University of Malta. Another important feature which is essential in the case of user entry validation, is that of correctly pointing out the mistake made during user entry. To establish such prompting of mistakes, a red mark is made visible near the field and a comment is added to notify the user of what change needs to take place. This validation drastically reduces the chances of potential errors occurring during allocation.

Wherever possible, text fields were replaced with dropdown menus or fields were automatically filled, as to reduce user input, and therefore eliminate trivial errors such as spelling mistakes. It is also to be noted, that the entry of date submitted is performed entirely by the backend, as a means of ensuring no malicious entry of incorrect dates as to manipulate the allocation system in the favour of a particular user.

**8.5.2 Performance considerations**

Performance of the system proves to be significant enough to be a preliminary requirement. In order to build an efficient system to allocate students to supervisors, performance is vital. Having a system that performs FYP Allocation after a significantly large timespan is far from optimal. As per previous discussions, performance is to be defined by execution and allocation occurring with seconds. This timespan has been ensured through testing that will be covered in a later section.

The performance of allocations must return consistent results. The main aim for the system was to deliver allocations in a more accurate manner, with less time spent than in the case of allocating manually. The definition of accurate in this case, would be a fair allocation that results in most students being assigned to top preferences. Another important performance factor would be that of having an algorithm that is deterministic, where each time the allocation is repeated with the same parameters, the result is always the same. Such consistency inspires confidence to all those affected by the system. Having many results for the same input would defeat the purpose and thus the allocation system would be rendered impractical for use.

**8.5.3 Security considerations**

Upon utilisation of the allocation system, all users must be ensured of security, especially due to the nature of the website being a university organization. Security was highly prioritized, taking the implantation of proper security into consideration at every stage of development. The main security issues to be considered were those that that any users might encounter when dealing with logging in and when preserving the data within the system following user entry. Any oversights may easily result in malicious use by users wishing to render the system irreparable.

During the login stage, security must be factored in when assigning users to roles in the system by by means of email addresses. It was to be ensured that users are not able to switch between roles and role type is strictly dependent on database values. Not developing with this factor in mind, may result in an unauthorised user breaching an unassigned role, the malicious user may then manipulate allocation data and possibly, delete areas from supervisor profiles. As mentioned, this was prevented by means of ensuring that users have only one role that is defined from within the database.

As a measure of security, administrators are solely responsible for performing allocations. Student or lecturers are unable to access such features in order to maintain the integrity of results produced and prevent any malicious manipulation of data. Another measure was also implemented. Permanently disabling the ‘Perform Allocations’ button upon performing the initial allocation. This was to prevent any inaccurate data being produced, as following the primary allocation, quotas are updated to reflect new availabilities for any students who are yet to be submitted.

With regards to the student preferences submission form, being ‘FormA’, a security measure was implemented in the form of validation. This measure employed cascading stylesheets as to prevent students from matching supervisors with areas that are not appropriated towards them as the primary supervisors. This therefore, ensure security through the integrity of data sent to the database in order to achieve performance of accurate allocations