Objective of the project:

The main objective of the project is to make this entire process automated with limited admin intervention.

Features:

• Initially, admin adds topics by logging into the portal which we store in our database and submits them which sends emails to students registered to that module that topics are added and preference needs to be submitted.

• Students then login into the portal and submits topics (let’s say as 5 topics) as their preference. These topics are stored in our database.

• After all the students submitting their preferences, admin checks them through a new page where all student’s responses are displayed, and then he runs a matching algorithm which allocates topics and groups for students.

• In our case, matching algorithm is a program which added weights to the topics by their priority (example: first preferred topic is assigned with a highest weight and lowest to the least preferred and calculate some kind of algorithm which can make us allocate topics accordingly).

• After running matching algorithm admin can view the allocated topics and groups, and if he is fine with the allocation, he/she can publish the allocation where students can check them by logging into the portal or by sending a mail with an allocated topic and group members. we are using bipartite weighted matching algorithm where every preference is allocated with some weight. This is a common approach which suits for many problems but the tricky part is we must allocate 1 topic to maximum of 5 students, if it is just assigning only 1 student a topic we can directly use this algorithm, but In our case we need some more logic where we must allocate accordingly.

Requirements:

Essential:

• We have 2 main users, admin and users. Here student needs two views, one for submitting the preferences and for viewing the allocation.

• Admin needs 3 views, one for adding the topics, one for viewing preferences of each student and final view which displays the allocation (it’s the same view which is shown for students too).

• We need a database for storing login details, topics, preferences and allocations.

• Login system for students and admin.

• API’s also needs to be designed to submit the topics initially and store them into the database and also for submitting the preferences of students.

• One of the main requirements of this project is implementing matching algorithm.

Recommended:

Admin must able to do all the actions of students and supervisors.

Admin must able to submit a preference on behalf of a student

Admin must able to submit a supervising topic on behalf of a supervisor

Admin must able to edit the preferences of student and supervisor

Admin must able to edit allocated groups.

Make the web application such a way that it meets WCAG requirements.

Making sure allocation is done properly when most of them choose same topic as their first preference.

Students can make changes to their preferences even after submitting for a while.

Optional:

Asking students their preferred group members.

Linking this to university website so that all the users can log in with their university IT accounts.

Making application more secure

Making application responsive to use them on any device, Currently, we are displaying the values of all the student names, allocations, groups as tables for which we cannot make them responsive.

More plugins and libraries to make better UI/UX

Can be able to merge with UOL application

Sending email to students before deadline, who are registered to this module, when they didn’t send their response (which is currently done manually)

Taking users preference first and recommend topics according to their preferences.