

PROJECT REPORT

GENERAL ALLOCATION PORTAL

October 28, 2017

160050026

160050034

160050045

Software Systems Laboratory

INTRODUCTION

The aim of this project was to explore new things besides the course curriculum and learn to apply new tools to make a useful programs. Django is a powerful python web-framework which can be used to develop complex database models and we have tried to explore most of the aspects and features of it.

All institutes, from educational to corporate companies to government sector, are faced with problems where they have to allot some choices to the applicants based upon their preferences and other criteria through a portal. We need a shorthand method to quickly create such a portal. Through the General Allocation Portal, any institute can create a portal and allocate choices without making a separate one for itself.

MOTIVATION

Large data handling is a big deal nowadays. So a database project has helped us understand and critically evaluate features of competing big data technologies.

Since every institute needs an application portal to fill in vacancies while relying on the preferences of the candidates, our portal can easily be used by them. Our portal will definitely spare a great deal of time and resources for all of them. Our portal may be well used in our institute as well as our department.

Already many portals exist for these problems but no abstract portal for all institutes. This is what made this problem interesting. We have created an abstract portal applicable in various such matchmaking problems.

PROBLEM STATEMENT

Develop a generic portal for any institution to register and to allot their candidates their seats based on their preferences and the candidates' data given by the institution.

SOFTWARE AND HARDWARE REQUIREMENTS

- (1) Python3
- (2) Django
- (3) HTML5

- (4) CSS3
- (5) SQLite3
- (6) A server to base the web based portal, though we have used a virtual server on our laptop itself.
- (7) Git

IMPLEMENTATION

Algorithm Used

For our matchmaking problems, we have used a modification of the Gale Shapley algorithm to allocate the seats.

Gale Shapley Algorithm:

This algorithm is basically the solution to the stable marriage problem . There are n men and n women, which are unmarried. Each person has a preference list of the people of opposite sex. We have to find a stable matching which is a matching of men and women such that there is no pair of a man and a woman who both prefer each other above their partner in the matching.

This can be modified to solve the seat allocation problem. Here the men represent the candidates and women represent the choices available. Each option can accept at most a certain number of candidates and the preference order for each choice is based on the rank of candidates.

Web Application GUI

For all the web based interface, we have used HTML5 and CSS3. Our python code and the user interface have been interfaced using django. The User interface include:

1. Institute registration

- (a) New Institutes can register using a form.
- (b) The portal, once active, allows various institutes to use the matchmaking problem solver and so it can cater a wide variety of matchmaking problems.

Institute Registration

Home

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Sign up

Figure 1: Registration Page

2. Institute Login

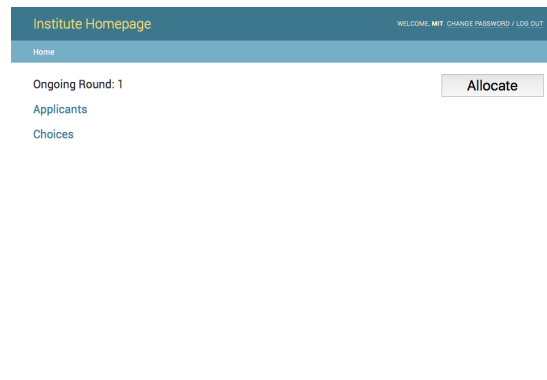
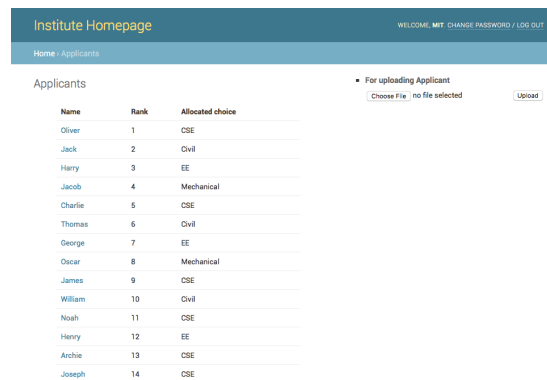
- The various institutes using the portal are admins with access to all the information of the applicants and the choices made available.
- The institute can create a large number of applicant users, through csv file input, to give their preferences and allocate them choices.

3. Applicant Login

- The various applicant users of the portal are also provided with a username and a password as given by their institute.
- They can use this to login and fill in their preferences.

4. Allocation

- The institute has an Allocation button through which they can control the number of rounds and its frequency

**Figure 2: Dashboard****Figure 3: List of Choices****Figure 4: List of Applicants**

- (b) Sort the students in order of their rank and allocated as per the rank list.
- (c) Checking its correctness is tough. One needs to solve the test cases manually and then check with the program output.

5. Freeze, Float, Drop:

Institute Homepage WELCOME MIT CHANGE PASSWORD / LOG OUT

Home | Choices | CSE

Choice name: CSE
Capacity: 50
Seats Filled: 6

[Update](#) [Delete](#)

List of Applicants who have been allocated CSE choice:

Name	Rank
Oliver	1
Charlie	5
James	9
Noah	11
Archie	13
Joseph	14

Figure 5: Choice Detail as seen by institute admin

Applicant Portal WELCOME GEORGE CHANGE PASSWORD / LOG OUT

Name: George
Institute: MIT
Rank: 7
Ongoing Round: 1
Allocated choice:

Drop Float Freeze

Fill your choices and priorities:

- Choice: CSE
Priority: 1
- Choice: EE
Priority: 2
- Choice: Civil
Priority: 3
- Choice:
Priority:
- Choice:
Priority:

[Submit](#)

Figure 6: Applicant Interface

After allocating once, all the applicants can choose to either freeze, float or drop their current branch.

- Freeze: Accept the current allocated choice and be removed from further rounds of allocation.
- Float: Be included in further rounds of allocation thus risking the current allocated seat in hope for a better allocation.
- Drop: To drop the current seat and not be included in further rounds of allocation.

Database Modelling:

To store all the data efficiently, multiple data tables have been used and linked together by the use of foreign keys. The tables created and the information stored are given as follows:

1. Institutes:

- (a) name/username
- (b) Round number currently organising

Each Institute has a username and a password and a user is created every time an institute registers so as to login again.

2. Applicants:

- (a) name/username
- (b) Rank
- (c) Is_float- to store if the applicant has accepted his allotted choice or not

3. Choices:

- (a) This stores the choices made available by the institutes
- (b) Name of the choice
- (c) capacity - maximum number of students allowed to be given the choice

4. Application: This table relates an applicant to the choice he prefers and also stores the priority set by the applicant for that choice.

5. Allocation: This table relates an applicant to the choice he has been allocated.

DELIVERABLES PROMISED

1. Login Page - Verification through data set
2. Update Profile Info
3. Fill Choices
4. E-mail notification after each round
5. Seat Confirmation after each round
6. Additional implementation('if time permits ') - Forum to clear doubts and queries

FEATURES DELIVERED

1. Abstract portal for any university to register and handle data for all of them separately.
2. Registration Page to register any university and create as many applicants as it allows.
3. Login Page, separate for Institutes and Applicants
4. Input Page for Applicants to fill in their choices and display their currently allocated choice
5. Ask for seat confirmation after each round which includes Freeze, Float and Drop
6. Update Profile Info - only the admin and the institute is allowed to do so

DEVIATIONS AND REASONS

1. We have created a general portal for any institute rather than a portal to cater only one institute as promised earlier.
2. We could not implement the forum (additional task) for people to ask their queries due to time constraints. Also earlier it was meant for an institute to answer the queries but for a generic portal, it seemed futile.
3. We could not implement the email notification for all applicants but it required a domain from the SMTP server which are either paid or are not allowed anymore because of security constraints.

BUGS AND LIMITATIONS

1. Applicants and choices can be added by the institute only in the .csv file. We have not implemented our version for .xls files.
2. Once all the choices are allotted to the applicants of the institute there is no option to undo the allotted choices.
3. Institute cannot add applicant's post-round, so all the applicants need to be provided before the first allocation action is performed.
4. Institute cannot allot an applicant a choice manually.

REFERENCES

- [1] Django documentation, <https://docs.djangoproject.com/en/1.11/intro/>
- [2] Jon Kleinberg, Éva Tardos. *Algorithm Design*, 1st Edition.
<http://blog.narenarya.in/right-way-django-authentication.html>
- [3] <https://www.tutorialspoint.com/django/>
- [4] Stack Overflow, www.stackoverflow.com
- [5] https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website