Devathon’22 Problem Statements

Online Admission module for MTech students

Problem Code : PS01
Preferred Tech stack : Django + MySql

Roles : Student , Wsdc_admin

Student :

1. Create a registration form with basic details : email , password , name , department, etc.
2. An application ID should be generated, which will be unique for all and should be an identifier for the department as well .
3. After login , a form needs to be filled , where the student needs to fill the form details like :
Name , Birthdate, Aadhar, Address, Gender, Specialization(multiple) , Category, PWD , Passport pic + other documents (marksheets , caste cert, etc .) .

Important Note:

- Here, the student must be able to choose multiple specializations while applying. But it should be in the same department . Thus, if my department is X , I can only go with specializations XA , XB and not YA , YB .
- Extension Validation for file (Frontend + backend) .
- File size validation .
- Proper Authorization and access control must be present .

Admin :

- After the student has applied ,the admin should be able to review the application along with all the specializations he/she has applied to .

- Further , the admin should be able to accept or reject the applications .
- The student should receive the result of his/her application .
- If rejected , Students should be able to apply again .
- If pending , no action should be given to students .
- If accepted , they should get a registration number .

Note:

- Admins should be able to see all the applications as per status , and sort or filter them according to departments and specializations .
- This can be implemented in multiple ways like multiple tables , multiple pages , etc .
- Students are able to upload docs along with the application , which will also be available to the admin while reviewing the application .
- Download Option for the admin to download the table data of filtered students in excel format .

Brownie points:

- Students receive their application status update via mail service .
- Good UI and UX .
- Admin is able to filter applications on the basis of eligibility criteria like marks also, etc . OR an alert which does not allow ineligible students to fill the application based on the criteria set by admin ,
- Payment integration for student application submission .
- Any security features , which can be integrated .

Announcement And Registration Platform

Problem Code : PS02
Preferred Tech stack : Django/MERN

Problem Statement:

NITW has more than 30 clubs,associations. There will be club recruitments, events, workshops, quizzes etc going on very frequently. Also there can be important announcements from the student council.Currently there are 2 mediums of publicity which are social media and offline poster/ classroom publicity. However it may happen sometimes that students might miss important stuff because of spam or some other reason. Also there will be so many google forms going around for each and every event which causes so much redundancy.

Description:

There can be a centralized and monetized platform for our institute where all announcements will be coming. All clubs/associations will have their verified unique accounts using which they can publish posts. Students can check announcements and register/apply for events.

Detailed points:

- There will be one admin user to verify and provide users to authorized clubs/associations of NITW.
- Those clubs will get the option to post an announcement (may include video,audio,image,text,gif) along with deadline(only for registration events),date,time,venue,event type details.
- Students can register(single click*) using their institute mail on the platform and they can check daily announcements on the app.

- Students can search for events/clubs and also can apply filters based on event tag(type), date,registration deadline etc. many more.
- Students can register/apply for events and the organizer will get a list and data of registered participants.

Brownie points:

- Organizers can restrict branch,year etc. for events and only users satisfying criteria can register. (5 points)
- Use your creativity to apply more filters for the student side to find events of their interest and give more flexibility,features on the organizer side as well. (according to creativity and use case of idea)
- Each student will get a one time QR code after registration which will be scanned by the organizer during the event to get details of attendees. (10 points)

* Single click registration means all data of a student(user) will be stored prior and users can register for events by single button click and data of that user will be sent to organizers.

One Place Portal

Problem Code : PS03
Preferred Tech stack : Django/MERN/Android

Statement :

Carrying a Mess card has always been a headache, have you ever been in a line to make a new one or requested them to issue a new one upon loss of the old one?
This is a daily scenario of a mess ever thought of making it simple.

How about a portal where one can issue new cards online, upon payment will get equivalent credits. Instead of carrying a mess card one can scan a QR code some credit will be deducted from one account (given one has a sufficient amount of balance).

This might further extend to calculating the number of people currently dining in a mess at some particular time.
Portal Should have two sides: Admin for Mess authority and Registration for students

Students Portal Features :

- Registration of new card / renewal
- Upon QR code scanning amount should be deducted from the account.
- Display the success or failure message on the screen.
- Should display all options of mess available during the registration.
- Ensure that one mess card credit should not be valid in another. (For example, if you have 50 credits of IFC A, you can not be used for IFC B).
- Rating of particular day

Admin Portal Features:

- They Should be able to see a complete list of student and their info.
- Should be able to query based upon name, roll, the last date, visited today or not.
- Display the number of people who already dinned.
- Payment or total money they received so far, can filter also between given dates as well.

Brownie points:

- Payment integration.
- One Person can dine only once and even if one misses to attend the Mess some time, an amount of credit will be deducted. (which is half the amount of the regular Price)
- Display the number of people currently dining at a particular time. (Let's say if you are planning to go to the mess you will be able to see the number of people still there to check the rush. Few Possible approaches: Take some time assumption of average eating time let's say Te = 20 mins. Then you can number of QR scanned in the last 20 mins.
- Any other useful feature is a plus point.

Seminar Hall Allotment.

Problem Code : **PS04**
Preferred Tech stack : **Open**

Problem:

It's Difficult to keep track of the room allotment in the seminar halls complex on the campus. Currently, the process is all offline where you need HoD to sign a document and submit it to the registrar's office to get the room allotted. You have no choice but to go with the seminar hall allotted. You cannot guarantee if the seminar hall assigned has all the facilities needed for the event. On top of that sometimes the room allotment clashes and one has to switch to a different room on an urgent basis.

Description :

User View:

- A web-based solution where a student/club / association can request a seminar hall.
- The request would contain the room required, the reason, and other important parameters, such as documents (Posters or Additional Permission Letter).
- One should be able to look at the current allotment situations of the rooms for the particular time frame.
- One should be able to filter the halls based on the requirements
  a) Hall capacity
  b) Functionality available e.g. projector, speakers, microphone, camera, etc.
  c) Location (CSE seminar, seminar halls complex or ALC)
- If the requested room is not available,, the app should be able to suggest the best time to do the event (recommended)

- He should also be able to cancel an event request/event and thus make the room available for others to use.

- Upon completion of the event, the student can enter the post-event information and thus provide feedback for review.

<u>Admins</u>

- He should see the number of pending requests.
- There should be a filter option to list the requests for rooms. For ex. Day of request, purpose and student.
- Here the incharge can approve or decline the student's request.
- The admin can also lock rooms from requests.
- The admin should be able to edit room information like the equipment available etc.
- If the hall is misused or not used the admin should be able to block the student from requesting further rooms.

<u>Brownie Points:</u>

- The admin side, the history of room allotment can be generated in pdf format wherein we can look at the allotment statistic of the room with details of the allotment.

  The statistic could show
  - The occupancy percentage.
  - No requests were declined/accepted.
  - Most preferred room

- The security of the application would be thoroughly tested, follow best security tactics for development. For ex. All the inputs should be cleaned, file uploads should be checked etc.

Hostel Issue Resolver

Problem Code : **PS05**
Preferred Tech stack : **Open**

Statement:

There are so many hostel related issues raised daily which go unattended. Maybe the carpenter was busy solving some other issue or maybe the electrician's work hours were over, but the students' time is too valuable for regularly reminding them. Maybe some management and tracking would better the situation. What can you as a genius software developer do to solve this problem?

Description:

Design and develop an issue tracking system to raise and track hostel related issues.

Students:

- Raise issues in the form of <description, who does the issue concern/type of issue (carpenter, electrician, data center, etc), issue status, timestamp, student information, etc all the issue related info you think might be helpful>
- Upvote/++ issues if multiple students are having similar issue

Resolver:

- View list of issues which can be sorted by <timestamp, upvotes, etc and/or filtered by floor, block of hostel, etc>
- You may show a resolver only the issues that concern them (resolver type)
- View and update status of issue <pending, resolving, resolved>
- Commenting under each issue for specific issue related communication

Brownie Points:

- Develop the Work distribution system (which may distribute work among resolvers by looking at their specific skills, availability, etc or in any other creative and efficient way)
- Analysis related to the issues(pending issues, resolved issues)
- If developing all this is easy, you can go beyond and flaunt your skills ;)
- Automatic allotment of the request to the resolver would be rewarded.

The Elevator Chaos

Problem Code : PS06
Preferred Tech stack : Open

We have all definitely been in this place, when in a hurry, where the lift mechanisms just added more frustrations onto our day. This is a common problem for the lift of 1k and other hostels, the long waiting time can be stressful especially when one is late for a class or an important meeting. The design of the elevators is even more troubling as they are coded to sync with each other, therefore on requesting once, all lifts shall transport to your floor. This is not optimal as we need to face the combined waiting time of all lifts, nor is it efficient in terms of energy consumption. The order of stopping with pressed buttons is also not optimal, with it stopping at the topmost floors first by default due to first come first serve algo, making passengers that requested for floors below that needing to wait an additional extra time. This becomes an even more critical problem when we consider skyscrapers with dozens of floors and passengers. Thus in order to resolve these issues the need for a new efficient and optimal design is evident.

You have to design, and provide a simulation* of an algorithm which operates N elevators on a building of M floors. Try to design an algorithm which is as close to an ideal algorithm.

Here ideal algorithm means it should have below properties:

- Energy efficient - consume energy efficiently
- Low Response Time - minimum time to serve the elevator request
- High availability - It should be available to all the floors and requests from any particular floor should not be on hold for a long time.

For getting better idea about problem statement watch this video - [The Science Behind Elevators]

Note:

- N and M should be configurable. (1 <= N, M <= 100)
- There should be 1 unit time taken by elevators to move from one floor to another, and unit time should also be configurable.
- For simulation, you can take elevator requests as input either from the GUI of your application, the console of your program, any input file or any other way, but it must not be hard coded in the code.
- Mention all the cases that you are handling in a document.
- It is not necessary to implement the solution mentioned in the above youtube video(if any).
- It’s better to have an understanding of time complexity for average case and worst case scenarios of your proposed solution.
- If you are designing an algorithm which is not a deterministic algorithm, it’s better to provide analysis for correctness and efficiency of your algorithm.

*Simulation - visualization of movements of elevators, It can be web app, mobile app or computer application.

JUDGING CRITERIA:

Evaluation would include but not limited to :

- No of problem statement point honored. Remember that some of the listed points could carry more point than others
- Preferred TechStack.
- Team contribution, distribution of the work.
- UI of the application
- Code structure, comments and functionality explanation.
- Presentation and Explanation. Quality of responses in the Q and A session.
- Brownie points covered.

Plagiarism would be strictly penalized.

There would be additional interviews on top presentation for students opting for WSDC team.