

# Problem Statements

## Web Dev:

### W1

Design and implement an application which enables a student to attempt tests and see their scores for various courses.

#### REQUIREMENTS:

##### 1. Registration:

- Students will send a request to register by filling course details and personal details.
- The Faculty Advisor will check and accept the courses and generate a password and mail the credentials to the student. **Automating the process of sending mail is a plus.**
- Student has to first reset the password to continue to the profile.

##### 2. Exams:

- Pattern and questions are decided by the faculty handling the course
- Faculty uploads the questions for the exam and those are to be stored in the Database.
- Types of questions:
  1. Single Answer Correct in MCQ
  2. Multiple Answer Correct with partial marking in MCQ
  3. Multiple Answer Correct without partial marking in MCQ
  4. Integer answer
  5. Fill in the blanks
  6. True/False
  7. Descriptive answer – Two options:
    1. Type the answer
    2. Upload PDF
- There should be an option to go from any question to any question in the test.
- Test duration should be handled. Test duration and start time should be provided by faculty.
- In each section, the question should be in jumbled order for different students.

##### 3. Evaluation:

- MCQs will be evaluated automatically.

- Evaluation should be done only after the student submits the exam. No evaluation should be done while the student is attempting the test.
- As soon as the student submits the test, marks will be displayed for MCQs.
- Have to handle different types of questions.
- Final marks have to be updated when the faculty updates the marks for descriptive questions and be displayed in the results.
- **Be creative as to how the faculty can update the marks.**
- **Any automated evaluation method for descriptive answers is a plus.**

#### 4. Scores:

- The student should be able to see his/her courses in the scores tab.
- In each course the student can view the exams he/she attempted.
- On opening a specific exam, the student should have a provision to see all the questions and the attempted and correct answers along with the marks he/she scored in that exam.

#### NOTE:

- Don't have to create the complete database now. But there should be a database to test the working of the application.
- There should be a window of 5-10 mins after which the student can't begin the exam.
- Try to provide an instruction page after opening the exam and before beginning the exam.
- As long as the student is on the instructions page, time shouldn't be counted.
- The begin exam button should only be activated as per the exam details provided by the faculty.

---

## W2

Design a material management portal, where the faculty will upload all the important materials and the students should be able to access them. There will be mainly two roles, one is student and other is faculty.

### Objectives:

#### Faculty Side:

- Ability to login/signup

- Courses offered by the faculty can be predefined in the database or can be selected just after signup.
- Every subject the faculty is currently teaching must have the ability to upload files.
- Ability to delete the files which were added previously.

#### **Student Side:**

- Ability to login/signup
- Students should be able to select courses offered just after the signup.
- Students should be able to access all the files uploaded by the respective faculty of their registered courses. (He should not be able to access not registered courses' files)
- Students should be able to sort the files in each subject based on the name/upload date.

#### **Additional Points:**

- File upload and display should be done by using google drive APIs.
- You need to restrict the students to download/print any file shared in the portal.
- You are free to use any open source library that is built upon google drive APIs

**Ex:** [Django Google Drive Storage — Django Google Drive Storage 1.3.5 documentation](#)

### **W3**

#### **Office File Tracking System**

This system helps users to track all the daily activities that are involving documents/letters and approval processes in a digital platform.

There are two roles user, admin (**May be more roles**)

#### **User side:**

- Should be able to request for approval by uploading files.
- Should be able to choose the category under which we are requesting.  
Ex: (Request for leave, Request for permission to organise an event)
- Should be able to view the status of the request.

#### **Admin Side:**

- Should be able to filter/search for the application based on category/status
- Should be able to accept or reject the application
- Should be able to also ask for additional files/info if application is rejected.

#### **Additional Points:**

- System should have login/signup functionalities for both user and admin
- Users should get an email if the admin accepts/rejects the application.

## W4

### Mess Extras Management System

Every semester a lot of complaints are received about incorrect mess bills.

Develop a system that helps users to track and facilitate the transaction the mess extras coupon between students and the mess vendors.

#### Things to note:

- The system should be robust enough to prevent any fraud from either parties and should not leave any chance for a dispute. The system should also be fast as to avoid long waiting time at the counters.
- The user and the vendor both should have a real time record of the coupons issued as applicable.

This problem statement has been left ambiguous on purpose, participants are free to devise any method that they think will be feasible to tackle this issue.

**Note :** Extra points to the participants that try to choose this problem statement.

---

#### App Dev:

**A1** Create an **e-book library NITW** app to help students find e-books that are related to courses and reduce their efforts in finding books on web.

The app has two views:

>Student View:

Each student should login using their wsdc credentials. The course related e-books should be directly available after login page. From there they should also have an option to open library and access all books

>Librarian View:

Librarians should have a different login page and he/she can add books with all the details such as course, subject, name, author name etc.

---

**A2** Create a **Grievance Portal and Complaint Tracking System**. The app must contain the following features

A user can post his/her complaint along with photos/documents to support their claim.

He/she should also select the department the complaint should be assigned to. (Rather than the user selecting the department, look into classification techniques to assign the department).

After posting the complaint, the user can view the current status of the complaint.

Each department has a separate view of the Grievance portal where they can view the list of complaints. An Administrator assigns each of these complaints to a particular individual.

As soon as there is an update in status, the individual should be able to view it.