# INTRODUCTION TO PROJECT

The "Faculty Schedule Master" project is a web-based system designed to manage and streamline the scheduling of faculty members for academic course modules. This project leverages the power of Spring Boot for the backend, React for the frontend, and MySQL for database management to create a seamless and efficient scheduling experience.

"Faculty Schedule Master" enables the admin to search for available faculty members and assign them to specific courses based on the schedule, which includes details like time, date, and subject. The system allows the admin to easily select and assign faculty for various course modules, ensuring that the right faculty member is scheduled for the right course at the appropriate time.
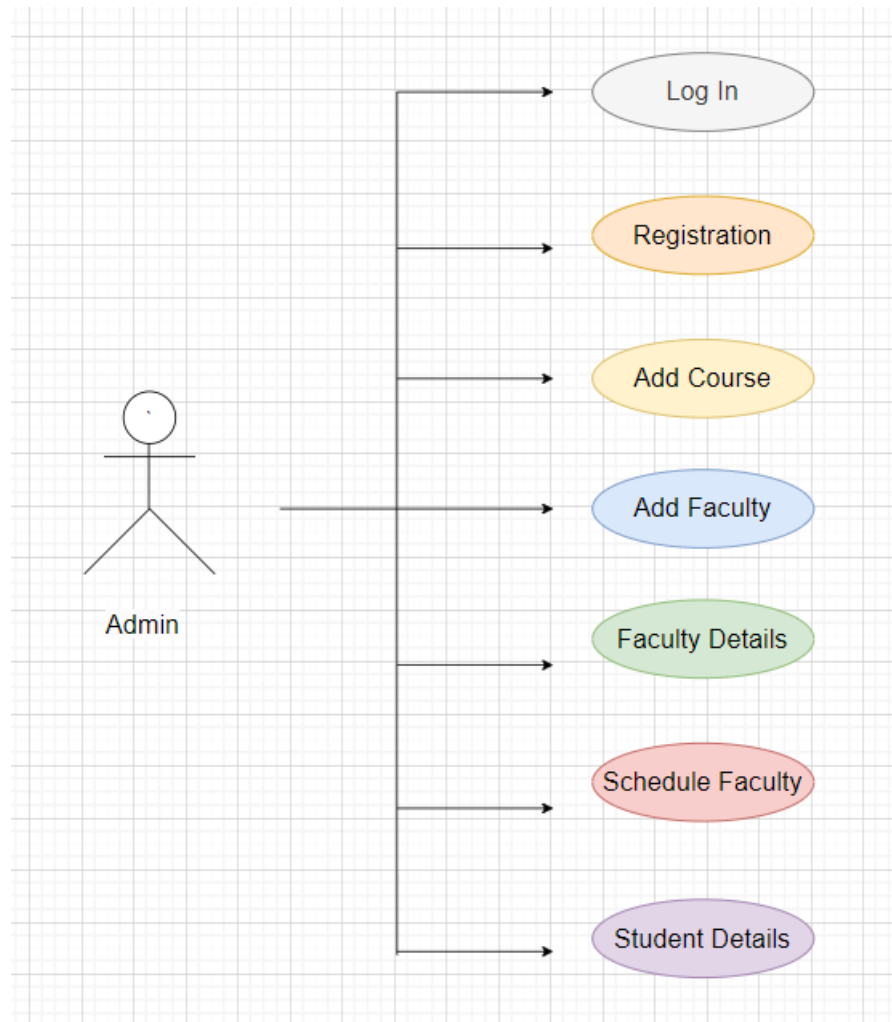
The project provides a user-friendly interface where the admin can quickly view available faculty members, their schedules, and assign them to courses without having to log in multiple times. Just as an airline reservation system checks for seat availability, "Faculty Schedule Master" checks the availability of faculty members for the desired time slot. If the faculty is available, the admin can proceed to schedule the course. Otherwise, the admin is prompted to select another faculty member or adjust the schedule.

To finalize the schedule, the admin needs to enter specific details such as the faculty's name, course name, time, date, and subject. The system then updates the faculty's schedule and the overall course schedule database, ensuring that all data is accurate and up-to-date.

This system is designed to improve the efficiency and organization of scheduling tasks within an academic institution, making it easier for admins to manage faculty assignments and ensuring that all classes are properly staffed.
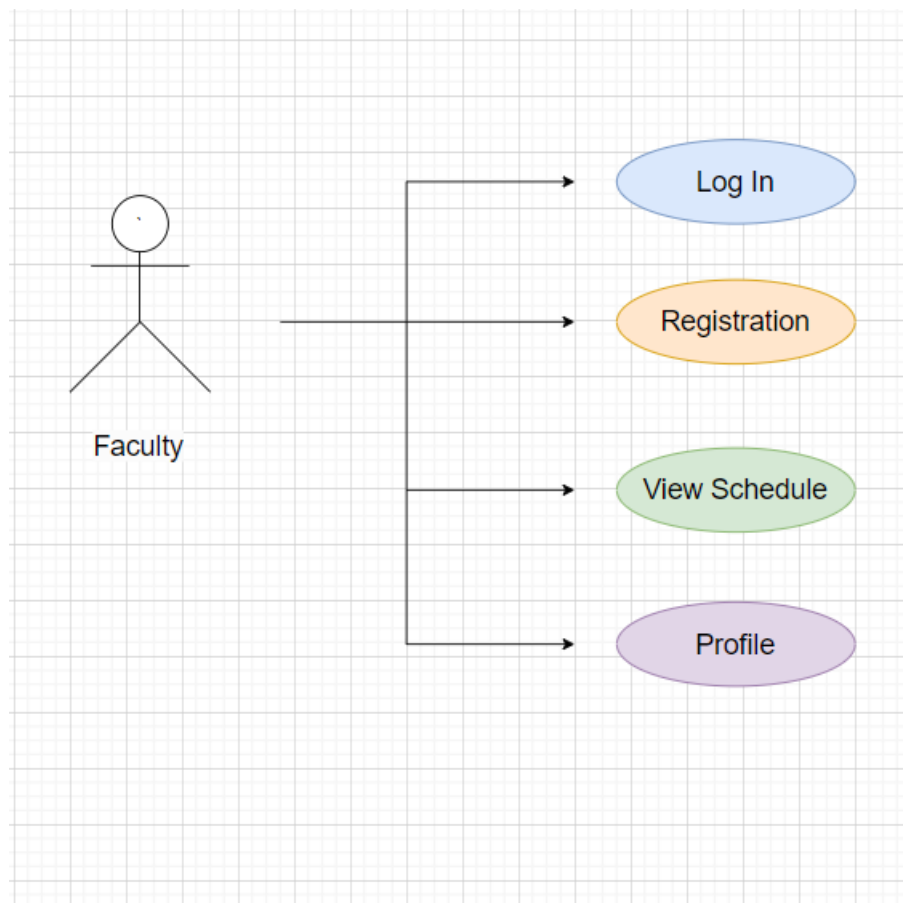
# REQUIREMENTS

## 1) Admin



In the "Faculty Schedule Master" project, the admin holds a pivotal role with extensive control over the scheduling and management of faculty and courses. Here are the core functionalities that the admin can perform:

1.  **Add Faculty**: The admin has the capability to add new faculty members to the system. This includes entering essential details such as the faculty's name, contact information, qualifications, and areas of expertise. Once added, these faculty members become available for assignment to various courses.
2.  **Add Courses**: The admin can create and manage course modules within the system. This involves specifying course details such as the course name, description, duration, and any prerequisites. These courses will later be used to assign faculty and schedule classes.
3.  **View Faculty Details**: The admin has access to a comprehensive view of all faculty members within the system. This includes detailed information such as their

qualifications, assigned courses, availability, and past schedules. This feature allows the admin to make informed decisions when scheduling faculty for specific courses.

4. **View Student Details**: The system also allows the admin to view details of students enrolled in various courses. This includes student names, contact information, enrolled courses, and academic performance. This feature is essential for maintaining a complete overview of the educational environment.

5. **Schedule Faculty for Courses**: The most critical function of the system is the ability for the admin to schedule faculty for specific course modules. The admin can allocate time slots and dates for each course, ensuring that the right faculty is assigned to the right course at the appropriate time. This scheduling process is crucial for maintaining an organized and efficient academic calendar.

## 2) Faculty



In the **Faculty Schedule Master** project, faculty members have access to a dedicated dashboard where they can manage and interact with their assigned schedules. Here are the key functionalities available to faculty:

## 1. View Schedule:

- **Detailed Schedule Display:** Faculty can view their entire schedule, including the specific courses they are assigned to, along with the corresponding time, date, and

subject details. This helps faculty members to keep track of their responsibilities and plan their activities accordingly.

- **Calendar View:** The schedule can be displayed in a calendar format, making it easy for faculty to visualize their commitments over weeks or months.
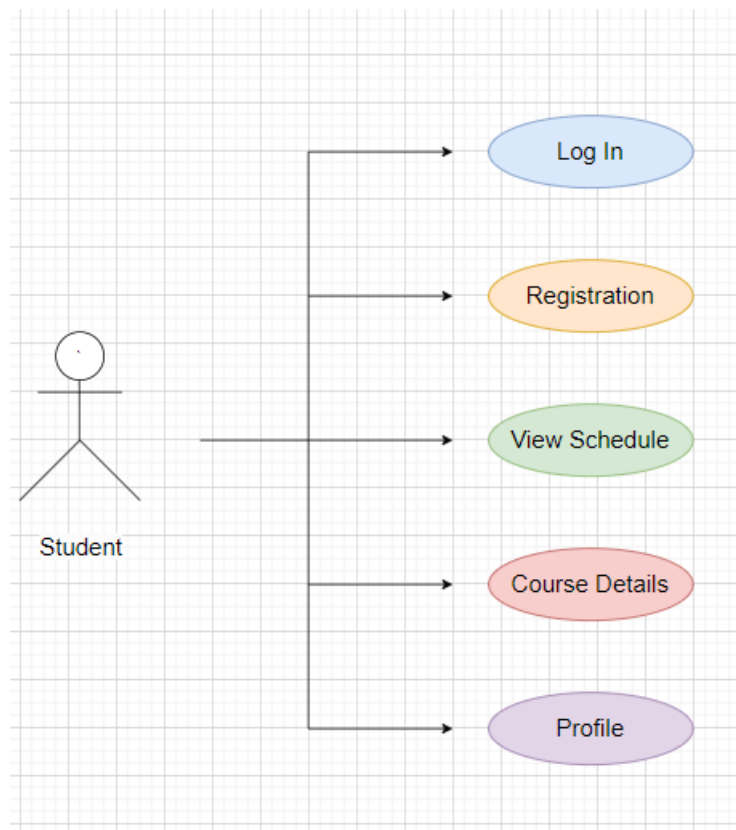
## 2. Edit Profile:

- **Personal Information:** Faculty members can update their personal information, such as contact details, profile picture, and other relevant data. This ensures that the admin and students have up-to-date information about the faculty.
- **Qualifications and Expertise:** Faculty can also update their qualifications and areas of expertise, which can be useful for the admin when scheduling courses and for students to know more about their instructors.

## 3. Access to Course Details:

- Faculty can access detailed information about the courses they are teaching, including the course syllabus, enrolled students, and any specific instructions or materials provided by the admin or the institution.

## 3) Student



In the "Faculty Schedule Master" project, students have the ability to explore various courses such as DAC, DMC, and DBDA. Each course is composed of specific modules—for example, the DAC course includes modules like Core Java, Advanced Java, React, MySQL,

etc. Once a student selects a course, they can view the daily schedule related to that course. Additionally, students are provided with the functionality to edit their personal profiles, ensuring that they can manage and update their information as needed.

# DATABASE DESIGN

## 1) USER

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | bigint | NO | PRI | NULL | auto_increment |
| created_on | datetime(6) | YES | | NULL | |
| updated_on | datetime(6) | YES | | NULL | |
| fist_name | varchar(255) | YES | | NULL | |
| last_name | varchar(255) | YES | | NULL | |
| password | varchar(18) | NO | | NULL | |
| role | varchar(255) | YES | | NULL | |
| status | bit(1) | YES | | NULL | |
| username | varchar(30) | NO | UNI | NULL | |

## 2) Courses

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | bigint | NO | PRI | NULL | auto_increment |
| created_on | datetime(6) | YES | | NULL | |
| updated_on | datetime(6) | YES | | NULL | |
| capacity | int | YES | | NULL | |
| name | varchar(50) | NO | | NULL | |
| description | varchar(255) | YES | | NULL | |
| end_date | date | YES | | NULL | |
| fees | double | NO | | NULL | |
| start_date | date | YES | | NULL | |

## 3) Modules

| id | bigint | NO | PRI | | auto_increment |
|---|---|---|---|---|---|
| created_on | datetime(6) | YES | | | |
| updated_on | datetime(6) | YES | | | |
| hours | int | NO | | | |
| name | varchar(50) | NO | UNI | | |

## 4) Faculty

| id | bigint | NO | PRI | auto_increment |
|----|--------|-----|-----|----------------|
| created_on | datetime(6) | YES | | |
| updated_on | datetime(6) | YES | | |
| address | varchar(255) | YES | | |
| current_age | int | YES | | |
| dob | date | NO | | |
| mobile_no | varchar(255) | YES | UNI | |
| pan_no | varchar(30) | NO | UNI | |
| user_id | bigint | YES | MUL | |

## 5) Students

| id | bigint | NO | PRI | auto_increment |
|----|--------|-----|-----|----------------|
| created_on | datetime(6) | YES | | |
| updated_on | datetime(6) | YES | | |
| address | varchar(255) | YES | | |
| city | varchar(50) | YES | | |
| enrolled_date | date | YES | | |
| mobile_no | varchar(15) | YES | UNI | |
| state | varchar(50) | YES | | |
| course_id | bigint | YES | MUL | |
| user_id | bigint | YES | MUL | |

## 6) Schedules

| id | bigint | NO | PRI | auto_increment |
|----|--------|-----|-----|----------------|
| created_on | datetime(6) | YES | | |
| updated_on | datetime(6) | YES | | |
| end_date | date | YES | | |
| end_time | time | YES | | |
| start_date | date | YES | | |
| start_time | time | YES | | |
| course_id | bigint | YES | MUL | |
| faculty_id | bigint | YES | MUL | |
| module_id | bigint | YES | MUL | |

# CODING STANDARDS IMPLEMENTED

| Identifier | Case | Additional Notes |
|---|---|---|
| Class | Pascal | Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class. |
| Method | Camel | Methods should use **verbs** or verb phrases. |
| Parameter | Camel | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios. |
| Interface | Pascal with "I" prefix | Do not use the '_' sign |
| Property | Pascal | Use a noun or noun phrase to name properties. |
| Associated private member variable | _camelCase | Use underscore camel casing for the private member variables |
| Exception Class | Pascal with "Exception" suffix | |

# TEST REPORT

| SR-NO | TEST CASE | EXPECTED RESULT | ACTUAL RESULT | ERROR MESSAGE |
|---|---|---|---|---|
| 1 | Register Page | Redirected to Next page | OK | Nothing |
| 2 | Login Page | Pop-up will come | Ok | Please enter username and password again . |
| 3 | Reset login | Only users password will be reseted | Ok | Nothing |
| 4 | Add Faculty | Only Admin can do this | Ok | Nothing |
| 5 | Add Courses | Only Admin can do this | Ok | Nothing |
| 6 | Add Modules | Only Admin can do this | Ok | Nothing |
| 7 | Schedule Faculty | Only Admin can do this | Ok | Nothing |
| 8 | Students Details | Only Admin can do this | Ok | Nothing |