NAME:	NIVETHITHA B
ROLL NUMBER:	7376221EC253
SEAT NUMBER:	138
PROJECT ID:	23
PROBLEM STATEMENT:	TIMETABLE

1. INTRODUCTION:

Creating a responsive timetable for classes using PYTHON Stack. This PYTHON stack is ideal for developing dynamic, high-performance web applications.This Python stack is a comprehensive and versatile solution for web development. The front end comprises HTML, CSS, and JavaScript.On the back end, Python and the Django framework offer robust, scalable, and secure server-side logic. For database management, PostgreSQL and MySQL are employed. APIs, including OpenAPI, SOAP APIs, and RESTful APIs, facilitate seamless communication between different software components, enhancing interoperability and functionality.This website comprises of two modules (Admin log page,Academic Selection Page(for both students and BIT committee).Login Page for logging in the page using student's individual bitsathy email id, Academic Selection Page for viewing the timetable of respective year.

PURPOSE:

This project is a full-fledged web application that simplifies the process of generating and managing timetables for educational institutions. By leveraging a Python-based server with Django, a robust front-end technology stack, and reliable database management systems, the application provides a seamless and efficient user experience. The comprehensive error handling, validation mechanisms, and API integrations ensure the system's reliability and scalability, making it a valuable tool for administrators, faculty, and students alike.Through this document, we can understand the purpose and novelty of the project.

1.1 REQUIREMENTS:

1) Django(Python Web)
2) MySQL Database Server
3) HTML,CSS, Javascript(Visual Studio Code/Sublime Code Editor)

1.3 SCOPE OF THE PROJECT:

- It is an open portal for students to view timetable for classes according to their year of study, department and semester.
- Support for multiple departments with distinct requirements.
- Excluding clashes like same venue and faculty scheduling conflicts, overlapping class times for students. Timing clashes also occur when classes are scheduled too closely or during break periods.

1.4 SYSTEM OVERVIEW:

Users:

1. Students / Faculty:

Students have to login the page using their bit-sathy email address.Once they logged in they are asked to choose the academic year,semester,department.According to the details the timetable is shown.Faculty have to login through their bit-sathy email address.Once they logged in they need to fill their faculty id.According to the faculty id respective timetable for the id is shown.

2. Admin:

Administrators have the power to enter the details required for the timetable generation.Once login is done it will direct to the timetable generation modulator.Admin should provide the details for faculty timetable and students timetable.For faculty timetable generation faculty id,venue,subject code and name,number of students should be given by the admin.For every faculty their id and handling subject details should be given.For students timetable venue,subject code and name,faculty name,time should be given for every department and every year.

1.2 PROJECT FLOW:

- This Project opens with a user authentication (Login credential).
- For the admin if it pass user authentication (i.e. If the given mail id and password are correct),then the login page will automatically direct us into the timetable generation module in which the details for the timetable generation is given and the timetable should be generated automatically.
- For students if it pass user authentication (i.e. If the given mail id and password are correct) , then the login page will automatically direct us into the academic selection page where we can choose the department , semester and year of study. After providing the details the timetable for the classes is shown.

- For faculty if it pass user authentication (i.e. If the given mail id and password are correct),then the login page will automatically direct us to enter their faculty id.Then the timetable for the faculty is shown.
- If the user authentication fails (i.e. If the given mail id or password is wrong), then an error message pops at the login page (“Invalid username or password”).
- If the admin login is successfully completed the admin should provide the details about subjects,faculties,lab subjects,venue details.
- It will automatically generate the timetable and it can be viewed by faculties and students.

3.1 WORKING MECHANISM:

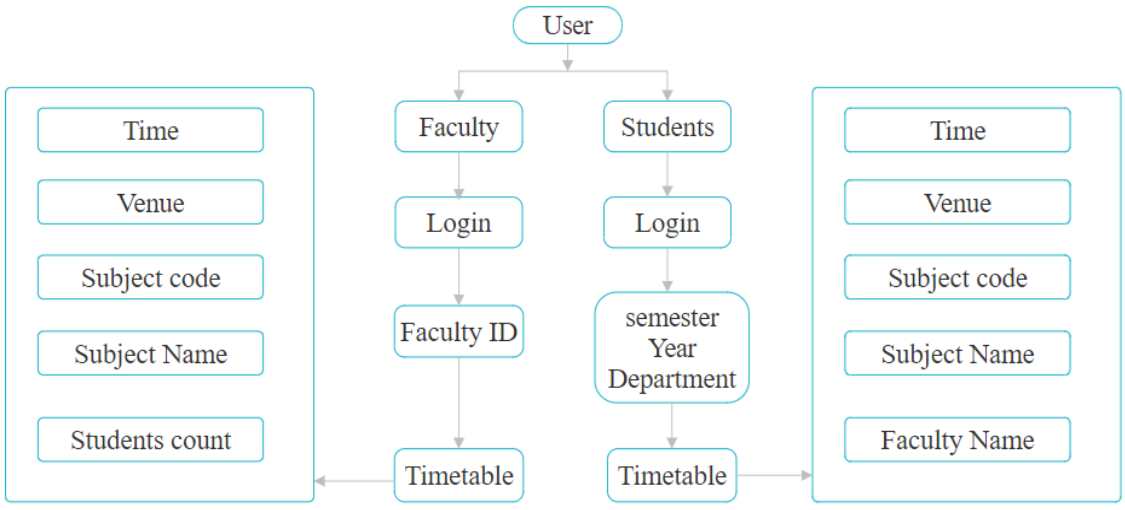
The timetable generation project involves creating a web application that allows an admin to manage and generate timetables for faculty and students. Hosted on a Python-based server using Django for web development, the application utilizes HTML, CSS, and JavaScript for the front end, and PostgreSQL and MySQL for the database.

1. Users access the website by entering the URL in their browser, which the web server then forwards to the Django application.The first interaction users have is with the login system, where they must authenticate themselves using their email and password. Django's built-in authentication system handles this process, ensuring that only authorized users can access the system.
2. After logging in(sign in with our bit-sathy email id), and based on their credentials, they are directed to their respective dashboards.When the given Email Id and Password are correct then the PHP running in the login site will automatically direct into the dashboard.
3. If the user email or password is wrong, then the error message will pops at the login site which is triggered by the JavaScript.
4. If the login credentials are correct, users are redirected to their respective dashboards based on their roles (admin, faculty, or student).
5. The admin dashboard is the control center for the timetable generation process. Upon successful login, the admin is presented with various options for managing timetables. This includes inputting or uploading details for both faculty and students.
6. Faculty details may include time,venue and subjects taught, while student details might encompass subject name,time,venue and faculty name. This information is entered through forms and stored in the PostgreSQL database.
7. The core functionality of the project revolves around the timetable generation algorithm. Once the necessary details are input, the algorithm processes the data, taking into account various constraints such as faculty availability and classroom schedules to generate conflict-free timetables.

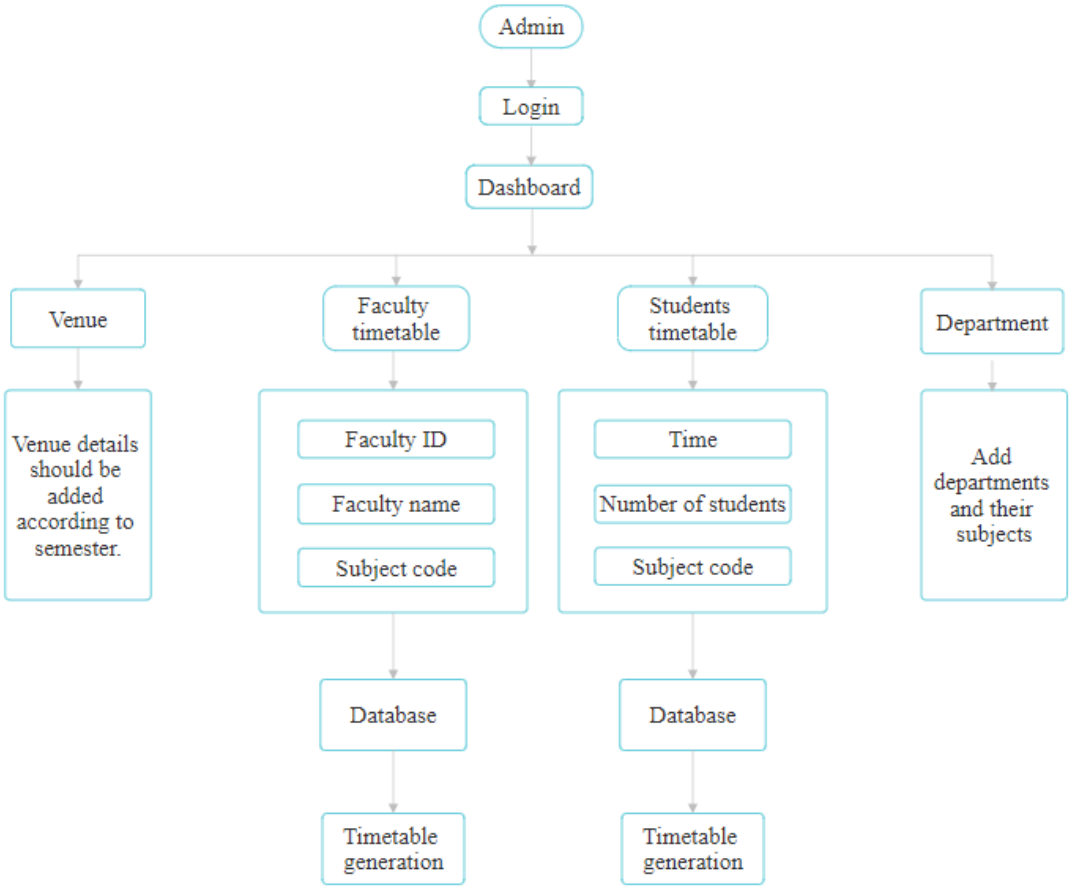
8. For faculty, the algorithm ensures there are no overlapping classes and that their availability is respected. For students, it ensures that their courses do not clash in terms of timing.
9. After the timetables are generated, they are saved in the database and made accessible to faculty and students through their respective dashboards.
10. To keep users informed, the system employs a notification mechanism that sends alerts using RESTful APIs. This ensures that faculty and students are promptly notified when their timetables are available or if there are any changes.
11. The project utilizes PostgreSQL for primary data storage, managing user information, timetable details, and configuration settings through Django's Object-Relational Mapping (ORM) system.
12. MySQL can be employed for storing additional data such as logs, historical records, and audit trails, providing a reliable and efficient database management solution.
13. For seamless integration and communication between different system components, the project uses OpenAPI specifications to define and document RESTful APIs. These APIs facilitate interactions such as sending notifications, fetching timetable data, and integrating with other systems.
14. Additionally, SOAP APIs are available for legacy system integration, ensuring compatibility with a wide range of external services.

1.2 WORKFLOW CHART:

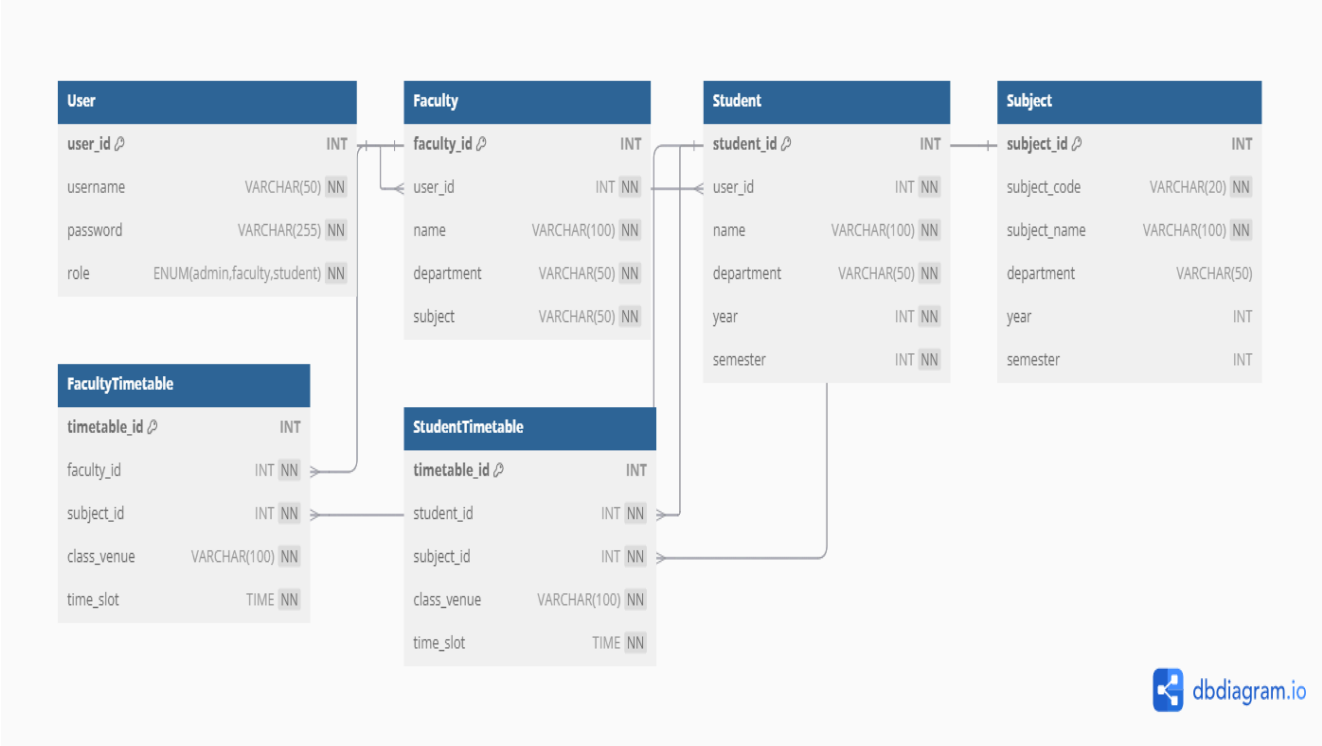
User Interface:



Admin's Interface:



ER DIAGRAM:


User
user_id INT
username VARCHAR(50) NN
password VARCHAR(255) NN
role ENUM(admin,faculty,student) NN
Faculty
faculty_id INT
user_id INT NN
name VARCHAR(100) NN
department VARCHAR(50) NN
subject VARCHAR(50) NN
Student
student_id INT
user_id INT NN
name VARCHAR(100) NN
department VARCHAR(50) NN
year INT NN
semester INT NN
Subject
subject_id INT
subject_code VARCHAR(20) NN
subject_name VARCHAR(100) NN
department VARCHAR(50)
year INT
semester INT
FacultyTimetable
timetable_id INT
faculty_id INT NN
subject_id INT NN
class_venue VARCHAR(100) NN
time_slot TIME NN
StudentTimetable
timetable_id INT
student_id INT NN
subject_id INT NN
class_venue VARCHAR(100) NN
time_slot TIME NN
dbdiagram.io