Shri Govindram Seksaria Institute of Technology and Science

Computer science engineering department

TimeTable Generator

Submitted by:-
Atharva Maddhav
3rd Year
Enrollment no.:- 0801CS221037
Section:- A

Submitted to:-
Prof. Surendra Gupta
Mrs. Ashwini Sharma

1. Introduction

1.1 Objective

The purpose of this document is to specify the software requirements for the Faculty Timetable Management System. The document outlines the functional and non-functional requirements, system features, and performance goals to ensure that the project is developed according to the expectations of the stakeholders.

1.2 Scope

The Faculty Timetable Management System is designed to automate the process of generating conflict-free timetables for faculty members, taking into account their availability and preferences. The system will include modules for inputting faculty data, generating the timetable, and viewing or exporting it. The system will be scalable, user-friendly, and secure.

1.3 Definitions, Acronyms, and Abbreviations

- **FTMS**: Faculty Timetable Management System
- **UAT**: User Acceptance Testing
- **UI**: User Interface

2. System Overview

The Faculty Timetable Management System will be used by educational institutions to generate timetables for faculty members. It will offer the following functionality:

- Input module for faculty information.
- Timetable generation algorithm ensuring no scheduling conflicts.
- Ability for administrators to manually adjust the timetable.
- Notifications to faculty members regarding their schedules.
- Exporting functionality for the timetable in PDF or Excel formats.

3. Functional Requirements

3.1 User Authentication and Roles

- **Description**: Secure login and role-based access for users.
- **Requirements**:
    - Admin users should be able to log in securely using a username and password.

- Unauthorized users should be restricted from accessing or modifying the timetable.
- The system should include role-based access control (admin and faculty).

3.2 Faculty Input Module

- **Description**: Interface to input faculty and timetable data.
- **Requirements**:
  - The system should allow administrators to input the number of teachers, working hours per day, and working days per week.
  - The system should collect detailed information about each faculty member (e.g., name, subject, preferred working hours, availability).

3.3 Timetable Generation

- **Description**: Algorithm for generating the timetable.
- **Requirements**:
  - The system should automatically generate a timetable based on the provided input.
  - The system should ensure no overlap of classes for any faculty member.
  - The system should distribute teaching hours evenly throughout the week.

3.4 Timetable Management

- **Description**: View, export, and manually adjust timetables.
- **Requirements**:
  - Users should be able to view the generated timetable in a user-friendly format.
  - Users should be able to export the timetable in PDF or Excel formats.
  - Administrators should have the option to manually adjust the timetable.

3.5 Notifications

- **Description**: Send notifications about timetable updates.
- **Requirements**:
  - The system should notify faculty members about their assigned schedules.

- Administrators should receive alerts if any conflicts occur during timetable generation.

4. System Design and Architecture

4.1 System Architecture

- The system will follow a client-server architecture with a web-based front-end and a back-end server handling business logic and database interactions.

4.2 Modules Overview

- **Input Module**: Collects teacher information, working hours, and availability.
- **Timetable Generation Module**: Uses an algorithm to generate a conflict-free timetable.
- **Timetable Management Module**: Allows users to view, edit, export, and manage timetables.

4.3 Data Flow

- Users input faculty data via the input module.
- Data is processed by the timetable generation module.
- The generated timetable is stored in the database and can be viewed or exported through the user interface.

5. Software Requirements

5.1 Software Requirements

- **Programming Languages**: JavaScript (for the back-end), ReactJS (for the front-end)
- **Frameworks**:
    - **Front-End**: React.js/Vue.js
- **Operating Systems**: Linux/Windows
- **Version Control**: Git/GitHub
- **Web Servers**: Apache/Nginx

6. Milestones

1. **Project Initiation**:
    - Gather requirements, finalize the project team.

2. **Design Phase**:
    - Create system architecture and UI design.

3. **Development Phase**:
    - Develop the database, input module, and timetable generation algorithm.
    - Build the user interface and admin panel.
4. **Testing Phase**:
    - Perform unit, integration, and UAT testing.
5. **Deployment**:
    - Deploy the system on institutional servers.
6. **Post-Deployment**:
    - Provide maintenance, updates, and support.

7. Acceptance Criteria

- The system should meet the functional and non-functional requirements outlined above.
- The timetable should be generated with no conflicts and should be adjustable by the administrator.
- Faculty members should be notified of their schedules, and all functionalities should perform within the specified performance requirements.

8. Conclusion

The Faculty Timetable Management System will streamline the process of generating, managing, and adjusting faculty schedules. By automating this process, institutions will save time, reduce errors, and improve overall efficiency. The system is designed with scalability, usability, and security in mind to ensure long-term effectiveness.

9. References

- React tutorial by freecodeCamp.org , Lantiv studio