

SRS for Automatic Timetable Scheduler Website

Abid Mir - CS22B1002
Aditya Pratap Singh - CS22B1004
Shubham Soni - CS22B1053
Manvendra Singh - CS22B1054



Contents:

- Introduction
- Functional Requirements
- Non-Functional Requirements
- System Features
- Data Requirements
- Testing and Quality Assurance

Introduction:

In today's fast-paced educational institutions and organizations, efficient time management is crucial for maintaining smooth operations. One of the key challenges faced by educational institutions is the creation of timetables that optimize resource utilization, minimize conflicts, and meet the needs of students and faculty. To address this challenge, an Automatic Timetable Scheduler Website is proposed. This web-based application aims to simplify the process of generating timetables by taking user inputs for batch time slots, faculty availability, and course information, ultimately producing an optimized timetable that satisfies the constraints and preferences.

User Requirements

1. User-friendly Interface:

An intuitive and easy-to-use interface is crucial to ensure that users, whether administrators or faculty, can navigate the system effortlessly.

2. Batch Input:

Users should input batch-specific time slots, specifying days and timings for classes, enabling customization for each group.

3. Faculty Availability:

Faculty members must provide their availability, preferred time slots, and constraints, allowing the system to allocate classes effectively.

4. Course Information:

Users need to input comprehensive course details, including course names and credit scores, to inform allocation decisions.

5. Automated Scheduling:

The system's core functionality is its ability to automatically generate feasible timetables, saving time and effort.

6. Customization:

Users should retain the flexibility to manually adjust the generated timetable to meet unique requirements or preferences.

7. Conflict Resolution:

Effective conflict resolution is essential; the system should handle issues such as overlapping class schedules and faculty unavailability.

Functional Requirements:

1. Timetable Generation:

An intelligent algorithm is required to generate optimized timetables that fulfill all user-defined criteria while minimizing conflicts.

2. Batch Allocation:

Courses should be allocated to batches fairly, considering credit scores and adhering to specific batch constraints.

3. Faculty Allocation:

Faculty should be assigned classes based on their availability and preferences, optimizing their work schedules.

4. Conflict Detection:

The system should swiftly identify and highlight scheduling conflicts, enabling timely resolution.

5. Manual Adjustments:

Users should have the option to make manual adjustments to the timetable when necessary, ensuring ultimate flexibility.

Non-Functional Requirements:

1. Performance:

Timetable generation must be efficient, even when handling numerous batches and faculty members, to maintain user satisfaction.

2. Scalability:

As institutions grow, the system should seamlessly accommodate more users, batches, and courses without a drop in performance.

3. Availability:

The website must remain accessible 24/7 with minimal downtime to meet the needs of users around the clock.

4. Security:

Ensuring the privacy and security of user data is paramount; stringent measures should safeguard against unauthorized access.

5. User Training:

Providing comprehensive training and support to users is essential for maximizing the system's benefits and ease of use.

System Features:

1. Timetable Visualization:

An interactive and user-friendly timetable visualization feature allows users to easily comprehend and share schedules.

2. Conflict Resolution:

The system should not only detect conflicts but also provide viable solutions and alternatives to streamline decision-making.

3. Reporting:

Generating detailed reports, such as faculty workload analysis, aids in informed decision-making and resource allocation.

4. Notifications:

Automated notifications are essential to inform users when timetables are ready for review or when conflicts require attention.

Data Requirements:

1. Batch Information:

Comprehensive batch data, including names, sizes, and time slot preferences, forms the foundation for timetable generation.

2. Faculty Data:

Faculty profiles with names, availability, and preferences are essential for effective allocation.

3. Course Information:

Accurate course data, including names and credit scores, informs intelligent allocation decisions.

4. Constraints:

Any specific rules or constraints regarding batch scheduling must be clearly defined and incorporated into the system.

5. Availability of rooms

Number of rooms is necessary to concurrently schedule multiple classes without any clashes.

Testing and Quality Assurance:

1. Testing:

Rigorous testing, including unit, integration, and user acceptance testing, is crucial to validate the system's functionality and reliability.

2. Quality Assurance:

Continuous feedback collection from users helps enhance usability and performance, ensuring the website meets user expectations.

In conclusion, the Automatic Timetable Scheduler Website addresses the critical need for efficient timetable management in educational institutions. By carefully considering user requirements, functional and non-functional requirements, system features, data needs, security measures, and quality assurance, this website promises to revolutionize scheduling, benefiting both students and faculty alike.