International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Automatic Timetable Generator.

[1] Swara Mahadik, [2] Kashish Salunkhe

[1] Student, Diploma computer engineering, Vidyalankar Polytechnic Wadala (E)
[2] Student, Diploma computer engineering, Vidyalankar Polytechnic Wadala (E)

ABSTRACT:

Automatic Timetable Generator is a modern, intuitive platform designed to automatically generate class and exam timetables for colleges and universities. Built for simplicity and efficiency, Automatic Timetable Generator removes the hassle of manual scheduling and replaces it with an intelligent, user-friendly system.

With Automatic Timetable Generator, there are no complex spreadsheets or outdated tools. Teachers can easily input course details, rooms, and timings, while students can view their updated schedules with just a click. The platform supports dynamic editing, drag-and-drop adjustments, and separate views for both teachers and students — making it adaptable for every role.

Introduction

Today’s fast-paced academic environment, managing class and exam schedules manually is time-consuming, error-prone, and inefficient. That’s where Automatic Timetable Generator, the Automatic Timetable Generator, comes in.

Automatic Timetable Generator offers a simplified yet powerful platform to create, manage, and view academic timetables with ease. Built for both teachers and students, it eliminates the complexity of traditional scheduling tools. With just a few inputs, the system automatically generates optimized timetables — avoiding conflicts, managing rooms, and respecting preferences.

Unlike traditional systems, Automatic Timetable Generator prioritizes speed, privacy, and flexibility. Teachers can update schedules on the go, while students can instantly access their personalized timetables from any device. There’s no clutter, no confusion — just a smooth, intuitive experience.

With features like AI-powered scheduling, dynamic editing, and secure logins, Automatic Timetable Generator transforms timetable management into a seamless process — making academic planning smarter and more efficient for everyone.

Methodology

The Automatic Timetable Generator operates on a streamlined, intelligent workflow designed to simplify scheduling for educational institutions. Here's how it works:

1. **Data Collection**
   - Teachers and administrators input key details such as:
     - Subjects & Courses
     - Available Rooms
     - Teacher Availability
     - Preferred Time Slots
     - Course Batches and Departments

2. **Smart Scheduling Algorithm**
   - The system uses AI-assisted logic to automatically:
     - Generate conflict-free class and exam schedules
     - Assign time slots based on availability and priority
     - Allocate rooms efficiently based on capacity and use
     - Avoid overlapping sessions for shared resources or teachers

3. **Dynamic Timetable Creation**
   - Timetables are auto-generated for:
     - Each course/class

        - Individual teachers
        - Rooms and departments
    - Editable slots and drag-and-drop options allow for quick manual adjustments if needed.

4. **Role-Based Access**
    - **Teachers** have editable access to their subject slots and exam schedules.
    - **Students** can view dynamic, real-time schedules based on selected courses or departments.
    - **Admins** can oversee, update, and manage all schedules across the platform.

5. **Location & Optimization Features**
    - Geolocation can be used to factor in travel time between buildings.
    - Optimization routines ensure fairness in time distribution (no overload on a single day).

6. **Secure Storage & Sync**
    - All timetables are saved using local storage or cloud databases.
    - Optional integration with Google Calendar or mobile notifications keeps users updated.

1. Requirement Analysis

To develop an efficient, scalable, and user-friendly Automatic Timetable Generator, a thorough understanding of both the system needs and user expectations is essential. The following analysis breaks down the core functional and non-functional requirements of the platform.

2. System Design

- **ER Diagrams:**

□ **User**

- **user_id (PK)**
- **username**
- **password**
- **role (admin, teacher, student)**

□ **Course**

- **course_id (PK)**
- **course_name**
- **department**

□ **Subject**

- **subject_id (PK)**
- **subject_name**
- **course_id (FK)**
- **teacher_id (FK)**

□ **Teacher**

- **teacher_id (PK)**
- **name**
- **availability (e.g., JSON or slot string)**

□ **Room**

- **room_id (PK)**
- **room_name**
- **capacity**

□ **Timetable**

- **timetable_id (PK)**
- **course_id (FK)**
- **subject_id (FK)**
- **teacher_id (FK)**
- **room_id (FK)**
- **day**
- **time_slot**

□ **Exam**

- **exam_id (PK)**
- **course_id (FK)**
- **subject_id (FK)**

- **room_id (FK)**
- **date**
- **start_time**
- **end_time**

- **Design Systems & UI & UX:**

Minimal & Clean: Prioritize clarity and simplicity over clutter.

Modern & Responsive: Works across all devices – desktop, tablet, and mobile.

User-Centric: Tailored views and actions for each role (teacher, student, admin).

Quick Actions: Easy-to-use buttons and dynamic controls like drag-and-drop, toggles, and modals.

- **Technology Stack Selection:**
  - **Frontend: React. js / HTML, CSS, JavaScript, php for interactive user interface.**
  - **Backend: Node. js / Django to process the requests.**
  - **DatabaseMySQL / MongoDB to store user data, listings and messages**

3. Development Phase

1. Setup & Environment Preparation

Actions:

- Install necessary tools: VS Code, XAMPP/LAMP/WAMP, Git
- Set up project folder structure
- Initialize Git repository for version control
- Configure database (MySQL) and backend (PHP/Node.js)
- Create .env or config files for database connections

2. Frontend Development

UI Implementation:

- Build **Login/Register** interface (role-based: Admin, Teacher, Student)
- Create **Sidebar Navigation** with links to:
  - Dashboard
  - Timetable
  - Courses
  - Exams
  - Messages
- Develop **Timetable UI**:
  - Days on top (Mon–Sat)
  - Time slots on the side
  - Editable cells with dropdowns for subject, teacher, and room
  - Color-coded subjects for clarity
- Add **Dynamic Row Controls**:
  - Add/Delete time slot rows
  - Drag-and-drop functionality (via libraries like SortableJS)

3. Backend Development

Functional Modules:

- **Authentication System**:
  - Secure login/logout for users
  - Session management with roles (admin, teacher, student)
- **Course & Subject Management**:
  - Add/Edit/Delete courses and subjects
  - Assign subjects to teachers
- **Teacher Availability**:
  - Set available time slots
  - Conflict checking when generating schedules
- **Timetable Logic**:
  - Auto-generate or manually create timetables
  - Store timetable data in MySQL
  - Fetch and display based on user role and course

4. Exam Timetable Feature

Exam Scheduling:

- Input exams by course, subject, room, date, time
- Conflict checking for teachers/rooms
- Dynamic display for students

5. Messaging System (Optional Initial Version)

Basic Messaging:

- Teacher ↔ Student
- Chat-style interface using PHP & MySQL or Firebase Realtime DB

6. AI & Smart Feature Integration (Post-MVP)

Add in later stages after core development

- AI-based timetable suggestions (based on availability)
- Study schedule generator before exams
- Exam countdown timer
- Notifications (class reminders, exam alerts)

7. Testing & Debugging

Tasks:

- Test all roles: Admin, Teacher, Student
- Check timetable conflicts
- Validate exam scheduling
- Cross-browser and mobile testing

- Bug fixes and performance improvements

8. Optimization & Deployment

Final Touches:

- Optimize code for performance and loading speed
- Secure user inputs to avoid SQL injection or XSS
- Host backend on cPanel or Node server
- Host frontend on Netlify, Vercel, or same server
- Connect database and finalize deployment

4. Testing & Debugging

1. Unit Testing (Function-Level Tests)

Objective:

Test individual backend functions/modules in isolation.

Examples:

- Validate login logic for each role (admin, teacher, student)
- Test timetable slot conflict-checker
- Ensure subject/room/course CRUD operations work correctly
- Check exam clash detection function

Tools:

- PHP Unit (for PHP backend)
- Jest (if using JavaScript/Node.js)
- Postman for API route testing

2. Integration Testing

Objective:

Ensure that connected modules communicate and work correctly.

Focus Areas:

- Timetable slot update reflects in database and frontend
- Subject addition appears in dropdowns across all views
- Student selection of course loads correct timetable
- Teacher timetable updates correctly when subjects are edited

3. Functional Testing (Role-Based)

Admin Tests:

- Can log in, add/edit/delete users, courses, rooms
- Can view/edit global timetable
- Can schedule and reschedule exams

Teacher Tests:

- Can view/edit their own timetable
- Can add subjects to available slots

- Receives clash alerts for overlaps

Student Tests:

- Can select course and view class/exam timetable
- Can see room & time details
- Cannot edit or access admin/teacher tools

4. UI/UX Testing

Objective:
Ensure usability, responsiveness, and clarity across devices.
Checklist:

- Layout adjusts properly on mobile, tablet, and desktop
- Hover, click, and drag events work smoothly
- Modals and pop-ups open/close as expected
- Color-coded timetables are readable and intuitive

5. Database Testing

Objective:
Ensure data is stored, retrieved, and updated correctly.
Includes:

- Validate database schema relationships (e.g., teacher–subject–course)
- Test data integrity when deleting subjects or users
- Ensure cascade behavior (e.g., delete subject → remove from timetable)

◆ *6. Security Testing*

Key Focus:

- Prevent SQL injection in forms
- Protect against XSS in timetable entry inputs
- Secure login and session handling
- Restrict access based on user roles

Conclusion

The **Automatic Timetable Generator** is a robust and dynamic solution that streamlines the process of creating, managing, and viewing timetables for both students and teachers. With a user-friendly interface and powerful features, it simplifies timetable management while offering flexibility and scalability for educational institutions.
Key features such as **editable time slots**, **personalized schedules**, and **drag-and-drop functionality** provide a seamless experience for teachers to manage their schedules, courses, and rooms. On the student side, the platform offers a **view-only dynamic timetable** based on selected courses, ensuring clarity and ease of access.

Practical Applications

- **Teacher Scheduling:**
    - **Teachers can easily create, manage, and update their schedules based on subject assignments, available rooms, and time slots.**
    - **Prevents scheduling conflicts by automatically identifying overlaps or unavailable rooms.**
    - **Provides a streamlined way to manage workloads, ensuring teachers are not overburdened.**

- **Student Timetable Management:**
    - **Students can view their class schedules dynamically by selecting their course and year.**
    - **Provides clear visibility of course timings, locations, and instructor details, improving time management.**
    - **Allows students to plan their day by getting reminders before class starts.**
- **Exam Timetable Management:**
    - **Automatically generates and updates exam schedules, including room allocations.**
    - **Alerts students and teachers about exam dates, timings, and locations, ensuring no one misses their exams.**
    - **Conflict resolution feature ensures that there are no overlapping exam schedules.**

2. Administrative Efficiency

- **Automated Class Allocation:**
    - **Administrators can use the timetable generator to automatically assign teachers, classrooms, and resources to specific time slots, based on availability.**
    - **Reduces administrative overhead and the chances of human error when scheduling classes and rooms.**
- **Room Utilization Optimization:**
    - **The system tracks classroom usage and suggests optimal times for classes based on occupancy.**
    - **Administrators can better manage the allocation of resources by minimizing room downtime or overcrowding.**
- **Real-Time Updates and Notifications:**
    - **The system can notify administrators, teachers, and students about any last-minute changes, such as class cancellations, room changes, or exam reschedules.**
    - **Helps to keep all stakeholders updated in real-time, reducing confusion and missed information.**

Special Features

1. Automatic Timetable Management

- **Conflict Resolution: Automatically detects conflicts such as overlapping classes or double-booked rooms, and suggests the best alternative schedules.**
- **Personalized Scheduling: Suggests personalized study schedules or teaching hours based on past preferences, workload, and upcoming exams.**

3. Real-Time Notifications and Reminders

- **Class Reminders: Sends automated notifications and reminders to both students and teachers before a class begins, ensuring that no one forgets their schedule.**
- **Exam Alerts: Personalized exam reminders, including room allocations, exam timings, and countdowns for upcoming exams.**
- **Last-Minute Updates: Real-time notifications regarding schedule changes, such as cancellations or location shifts, ensuring users are always in the loop.**

Pros & Cons

Pros

 Efficiency and Time-Saving

- **Automated Scheduling**: The system automatically generates schedules, saving time for both administrators and teachers, reducing manual errors and the complexity of scheduling tasks.
- **Drag-and-Drop Functionality**: Allows teachers to quickly adjust their schedules without the need for complex procedures, making timetable management much faster.

□ **Real-Time Updates**

- **Instant Notifications**: Provides real-time updates about schedule changes, class cancellations, or new events, keeping everyone informed without delays.
- **Conflict Detection**: Automatically detects and resolves scheduling conflicts (e.g., double-booked rooms or overlapping classes), minimizing human errors.

□ **Personalized Experience**

- **Customized Timetables**: Allows students to view their personalized schedules based on their course selections, making it easier for them to stay organized.

□ **Role-Based Access Control**

- **Custom Permissions**: Different user roles (students, teachers, admins) have different levels of access, ensuring that users can only modify what they are authorized to.
- **Secure and Controlled Access**: Sensitive data is only accessible to authorized users, reducing the risk of unauthorized access or tampering.

□ **Cross-Platform Compatibility**

- **Mobile and Desktop Support**: The timetable generator is accessible from both desktop and mobile devices, ensuring that students and teachers can check their schedules anywhere and anytime.
- **Cloud Integration**: The platform's cloud-based nature allows seamless access from multiple devices, offering a unified experience for users.

□ **Advanced Analytics**

- **Timetable Analytics**: Administrators can gain insights into room usage, course popularity, and teacher workload, helping optimize scheduling and resource allocation.
- **Attendance Integration**: Directly integrates with attendance systems, reducing the administrative workload of manually tracking attendance.

□ **Security Features**

- **Encryption & Authentication**: Ensures user data is protected with robust encryption, making the platform secure against potential breaches.
- **Backup and Recovery**: Regular cloud-based backups ensure that timetable data is safe and recoverable in case of system failure.

Cons

□ **Initial Setup and Configuration Complexity**

- **Time-Consuming Setup**: Setting up the platform may be time-consuming, especially for institutions with complex course structures, many teachers, and diverse room schedules.
- **Customization Challenges**: While the platform offers flexibility, customizing the system to meet specific needs (e.g., institution-specific rules or unique teaching schedules) can require technical expertise.

□ **Dependence on Internet and Cloud**

- **Internet Dependency**: Cloud-based systems require a stable internet connection for access, which could be problematic in areas with unreliable internet service.
- **Data Privacy Concerns**: Storing sensitive student and teacher information in the cloud may raise concerns regarding data security and privacy, especially for educational institutions handling personal or sensitive information.

□ **User Learning Curve**

- **Complex Interface for New Users**: The platform may have a steep learning curve for new users, particularly teachers or students who are not familiar with digital timetable systems.
- **Training Requirement**: Users may need training on how to fully utilize the features, especially advanced ones like conflict resolution, AI-based recommendations, and drag-and-drop scheduling.

□ **Over-Reliance on Automation**

- **Lack of Human Touch**: Relying too heavily on automation may lead to unexpected issues that only human intervention can resolve, such as complex scheduling conflicts or unique classroom requirements.
- **System Limitations**: While automation is useful, it may struggle with handling highly personalized or rare scheduling requests that don’t follow typical patterns.

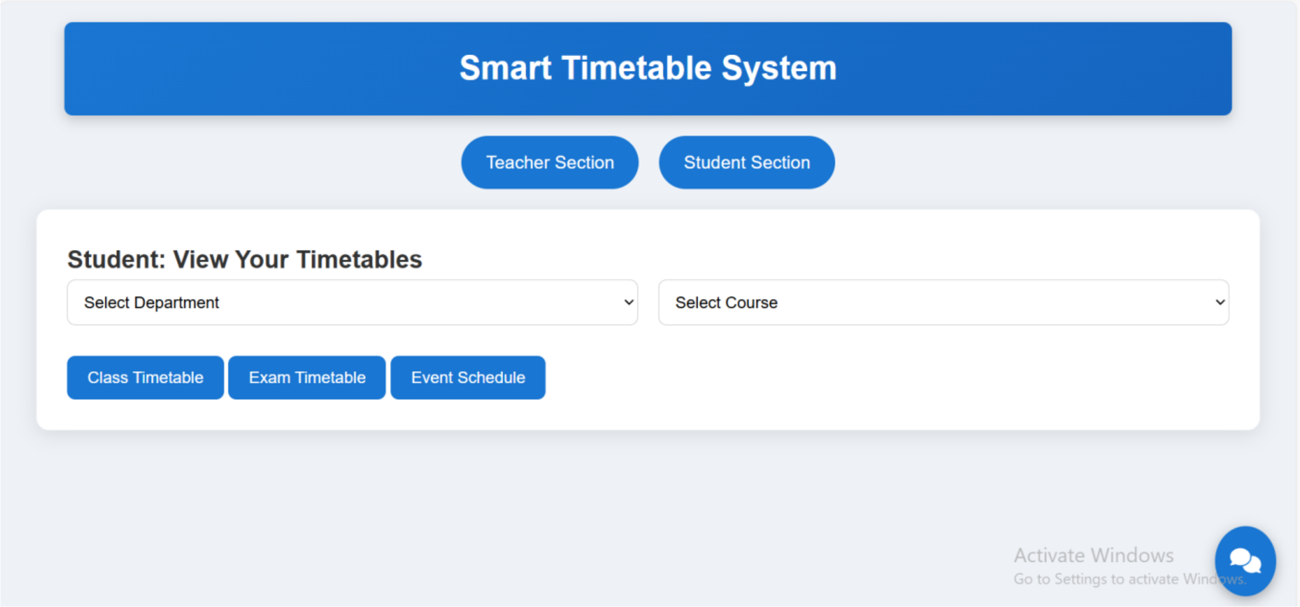
□ **Maintenance and Updates**

- **Ongoing Maintenance**: Cloud-hosted systems require ongoing maintenance and updates to keep them running smoothly, which may require dedicated technical support.
- **Feature Overload**: With continuous feature additions (like AI suggestions or analytics), the platform may become overly complex, making it difficult for some users to navigate or focus on the core functionalities.

Conclusions

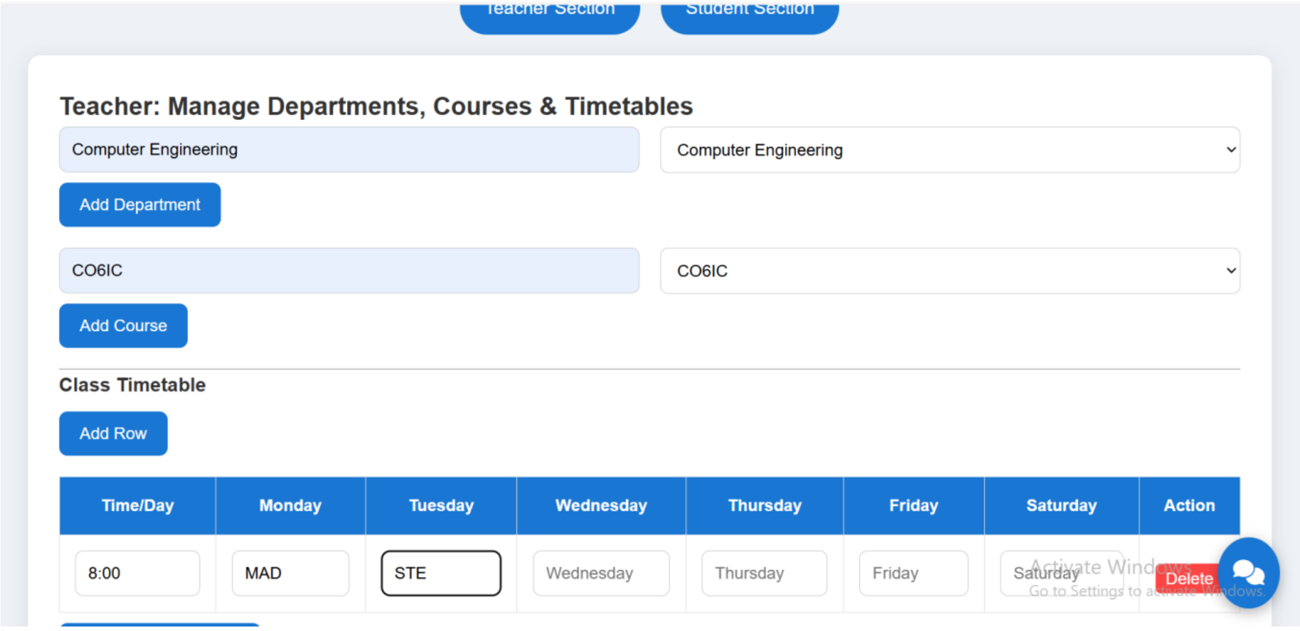
The college timetable generator is a powerful tool that simplifies and automates the scheduling process for both teachers and students. By efficiently managing classes, subjects, rooms, and exams, it ensures a well-organized and conflict-free timetable. With its user-friendly interface and dynamic features, it saves time, reduces errors, and improves overall academic planning and coordination.

Output:



Smart Timetable System

Student: View Your Timetables



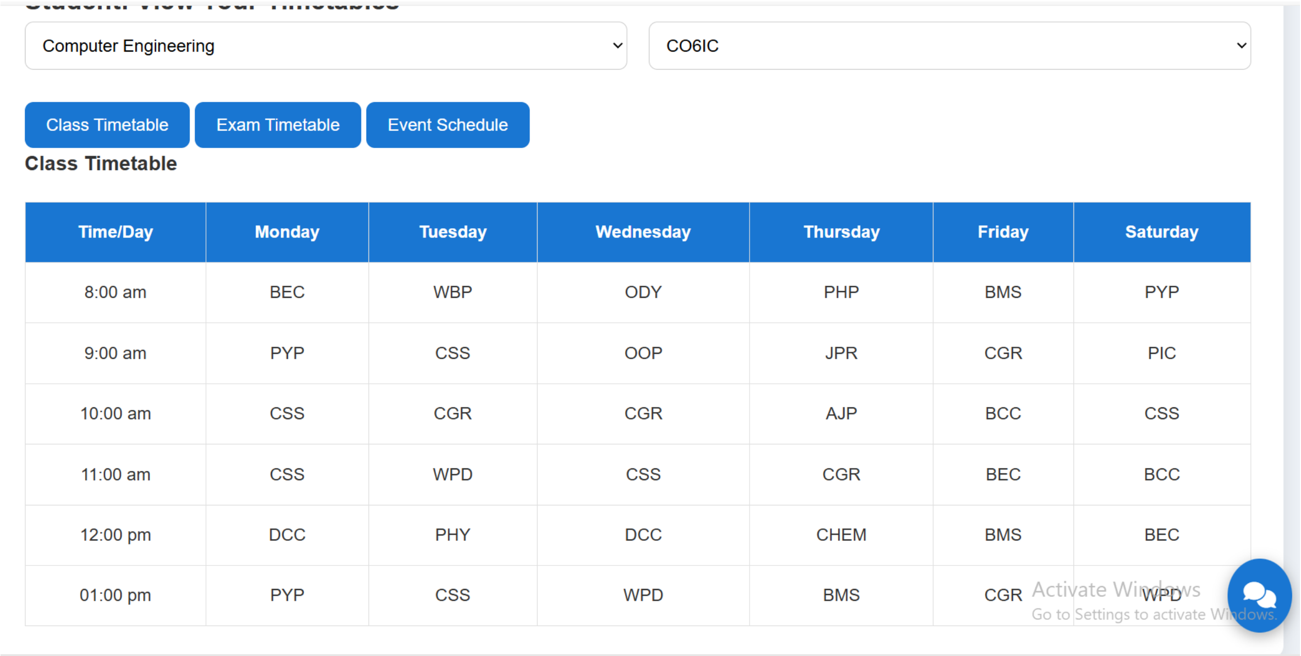
Teacher: Manage Departments, Courses & Timetables

Computer Engineering | Computer Engineering

CO6IC | CO6IC

Class Timetable

Time/Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Action
8:00	MAD	STE	Wednesday	Thursday	Friday	Saturday	Delete



Student: View Your Timetables

Class Timetable

Time/Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
8:00 am	BEC	WBP	ODY	PHP	BMS	PYP
9:00 am	PYP	CSS	OOP	JPR	CGR	PIC
10:00 am	CSS	CGR	CGR	AJP	BCC	CSS
11:00 am	CSS	WPD	CSS	CGR	BEC	BCC
12:00 pm	DCC	PHY	DCC	CHEM	BMS	BEC
01:00 pm	PYP	CSS	WPD	BMS	CGR	WPD