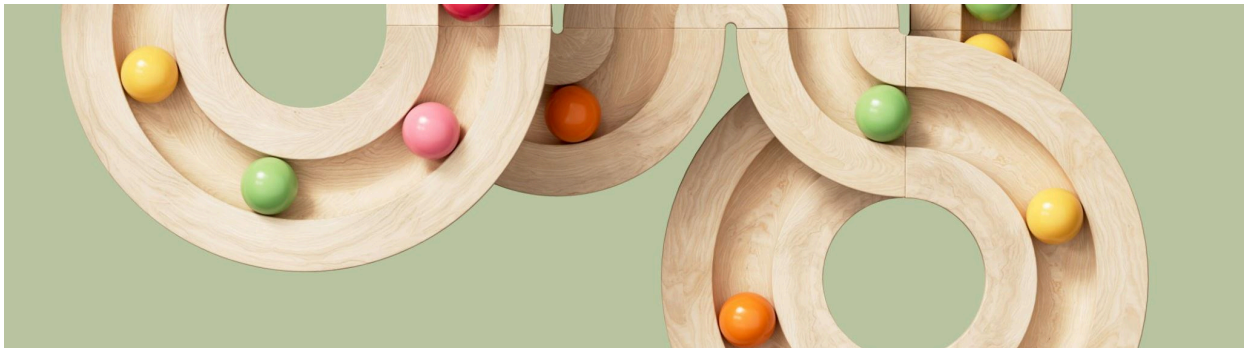


# Attendance management system

**Smart Attendance System**

Project Overview:

The Smart Attendance System is a modern, user-friendly desktop application designed to streamline attendance management for educational institutions. It enables efficient tracking of student attendance, management of student and subject details, and provides insights into attendance percentages. Built using the .NET Framework and a service-based SQL Server database, the application features a sleek, modern UI implemented as a Windows Form Application.

Objectives:

- To automate attendance tracking for students.
- To manage student profiles, subjects, and user access effectively.
- To record and calculate attendance percentages for each student.
- To provide a simple and intuitive interface for users.

**Key Features**

1. Student Management: Add, update, and delete student profiles (e.g., name, ID, class).
2. Subject Management: Manage subjects (e.g., subject name, code, assigned teacher).
3. User Management: Define user roles (e.g., admin, teacher) with appropriate access levels.
4. Attendance Marking: Record daily attendance for students per subject with a simple click-based system.

5. Attendance Percentage View: Display individual student attendance percentages based on recorded data.
6. Modern UI: A clean, visually appealing, and responsive interface for seamless user experience.

**Technologies Used:**

- Frontend: Windows Form Application (.NET Framework) with a modern UI design.
- Backend: .NET Framework for business logic and application functionality.
- Database: Service-based SQL Server for secure and scalable data storage.
- Programming Language: C#.

**Target Users:**

- Teachers: Mark and monitor student attendance.
- Admins: Manage students, subjects, and users.
- Educational Institutions: Streamline attendance processes.

**Project Scope:**

- Develop a desktop application for Windows OS.
- Support basic CRUD (Create, Read, Update, Delete) operations for students, subjects, and users.
- Implement attendance tracking and percentage calculation.
- Ensure data persistence using a service-based SQL Server database.

**Deliverables:**

- Fully functional Smart Attendance System application.
- Source code with inline documentation.
- User manual for operating the system.
- Database schema and setup instructions.

**Timeline:**

- Duration: 4-6 weeks (adjustable based on complexity).
- Milestones:
    1. Week 1: Requirement analysis and database design.
    2. Week 2: UI design and setup.
    3. Week 3-4: Core functionality development (student/subject/user management, attendance marking).
    4. Week 5: Attendance percentage calculation and testing.
    5. Week 6: Final UI polish, bug fixing, and deployment.

**Future Enhancements (Optional):**

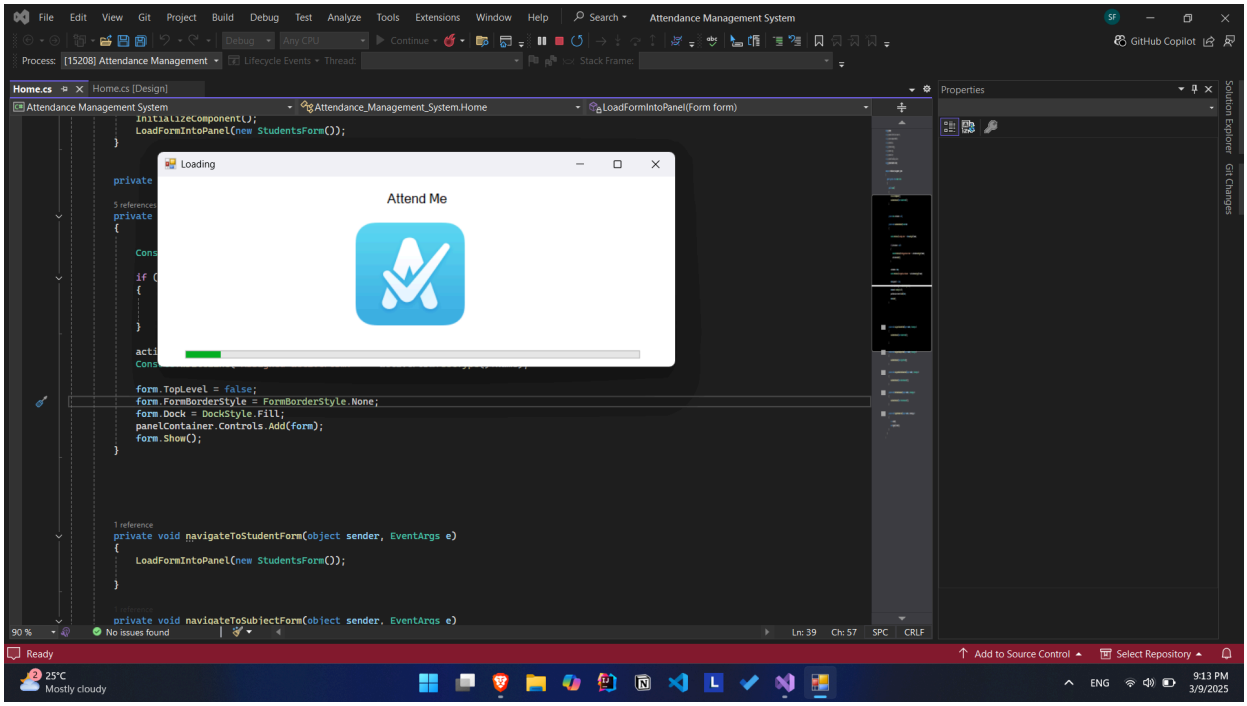
- Integration with biometric devices for automated attendance.
- Export attendance reports to PDF/Excel.
- Cloud-based database for remote access.

**Group**

| Reg.No | Name | Role (position) |
|---|---|---|
| 2021/ICTS/18 | Buwanajith Anupa | Regex and Error Handling |
| 2021/ICTS/43 | Kanushka Thejan | Database |
| 2021/ICTS/80 | Ruchika Harshajith | User Interface |
| 2021/ICTS/86 | Sasindu Lakshan | User Interface |
| 2021/ICTS/98 | Sandun Bandara | Developer |
| 2021/ICTS/153 | Yashen Fernando | Developer |

## Project Overview

### 01. Loading Screen

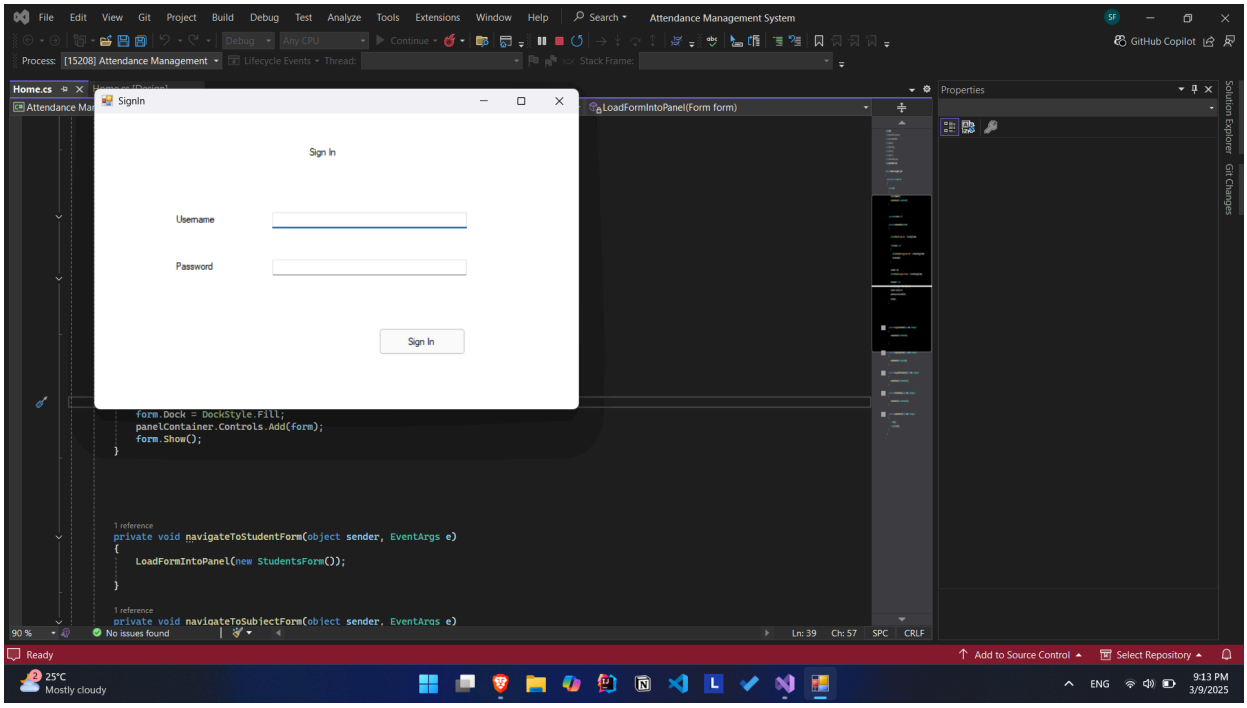



## 02. Sign In Screen

The Sign-In Form serves as the entry point to the Smart Attendance System, ensuring secure access for authorized users. It features a clean and minimalistic design with the following components:

1. **Username Text Box**:
    - A text field where users enter their assigned username.
    - Placeholder text: "Enter Username".
    - Validates input to ensure it is not left empty.
2. **Password Text Box**:
    - A secure text field where users input their password.
    - Placeholder text: "Enter Password".
    - Masks the input with dots (••••) for privacy.
    - Ensures the password is not left blank.
3. **Sign-In Button**:
    - A button labeled "Sign In" that initiates the login process.
    - Upon clicking, the system verifies the username and password against the SQL Server database.

- If credentials are valid, the user is granted access to the main application; otherwise, an error message is displayed (e.g., "Invalid username or password").




## 03. Home Screen / Menu

The Home Screen is the central hub of the Smart Attendance System, providing easy navigation and access to key features. It features a modern and intuitive layout with the following components:

1. Menu Buttons:
   - A set of clearly labeled buttons aligned vertically or horizontally (e.g., in a sidebar or top menu).

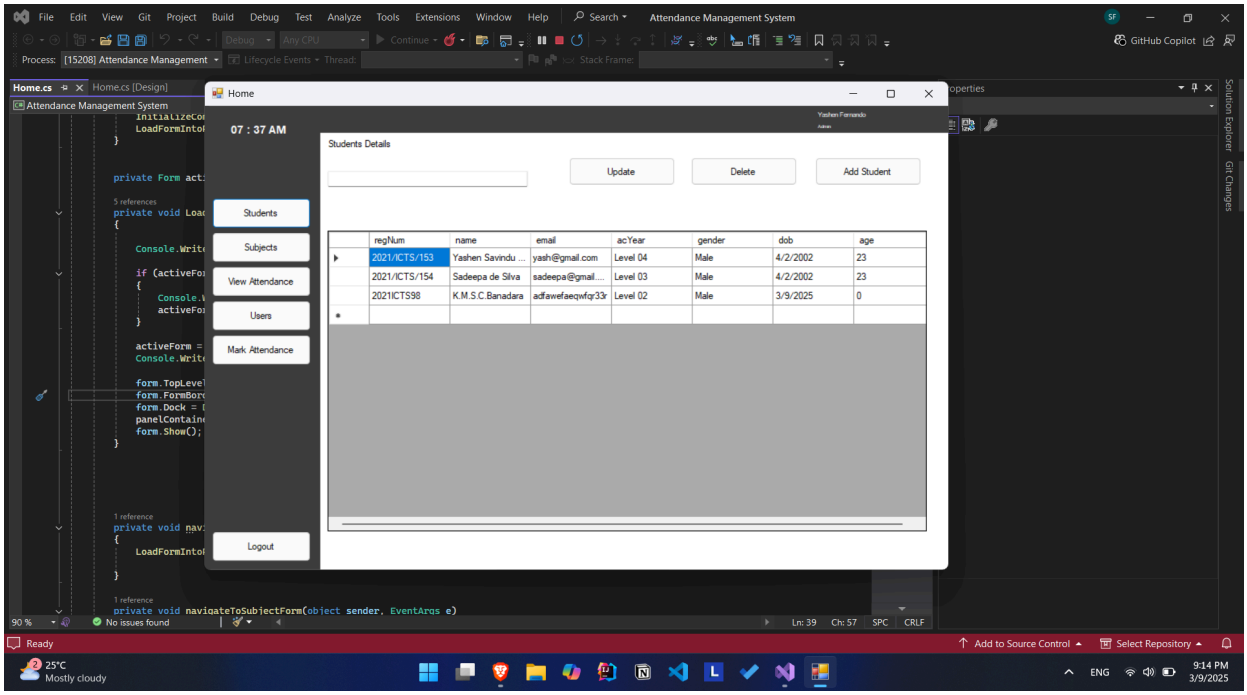
   - Examples of buttons:
     - "Student Management"
     - "Subject Management"

        - "Attendance Marking"
        - "View Attendance"
        - "User Management"
        - "Logout"
    - Each button is styled with a modern design (e.g., hover effects, icons) for an enhanced user experience.
    - Clicking a button triggers the loading of the corresponding form.

2. Container Panel:
    - A dedicated panel occupying the main area of the Home Screen.
    - Acts as a dynamic container to display forms based on the selected menu button.
    - When a menu button is clicked, the associated form (e.g., Student Management Form, Attendance Marking Form) is loaded into this panel, replacing any previous content.
    - Ensures a seamless, single-window experience without opening multiple windows.

**Functionality**

- Upon clicking a menu button, the system instantiates the relevant form and embeds it into the container panel using .NET Framework's Windows Forms capabilities.
- The panel adjusts dynamically to fit the loaded form's size and layout, maintaining a clean and organized interface.




The Subject Form is a key component of the Smart Attendance System, designed to manage subject-related information efficiently. It allows users to insert, update, delete, and view subject details within a modern and user-friendly interface. The form is loaded into the container panel on the Home Screen when the "Subject Management" menu button is clicked.

Components:

1. Subject Details Input Fields:
    - Subject Name Text Box: A field to enter or edit the name of the subject (e.g., "Mathematics").
    - Subject Code Text Box: A field to input or modify a unique subject code (e.g., "MATH101").
    - (Optional fields like "Assigned Teacher" can be added if needed.)
2. Action Buttons:
    - Insert Button:
        - Labeled "Add Subject".
        - Adds a new subject to the SQL Server database when clicked, after validating that fields are not empty.
    - Update Button:
        - Labeled "Update".
        - Saves changes to an existing subject’s details in the database.

    - Delete Button:
        - Labeled "Delete".
        - Removes the selected subject from the database after user confirmation (e.g., "Are you sure?").
    - Clear Button (Optional): Resets input fields for a new entry.
3. Subject List View:
    - A grid or table (e.g., DataGridView) displaying all existing subjects.
    - Columns include: Subject Name, Subject Code, and (optionally) other details.
    - Users can click a row to populate the input fields for updating or deleting a subject.

Functionality:

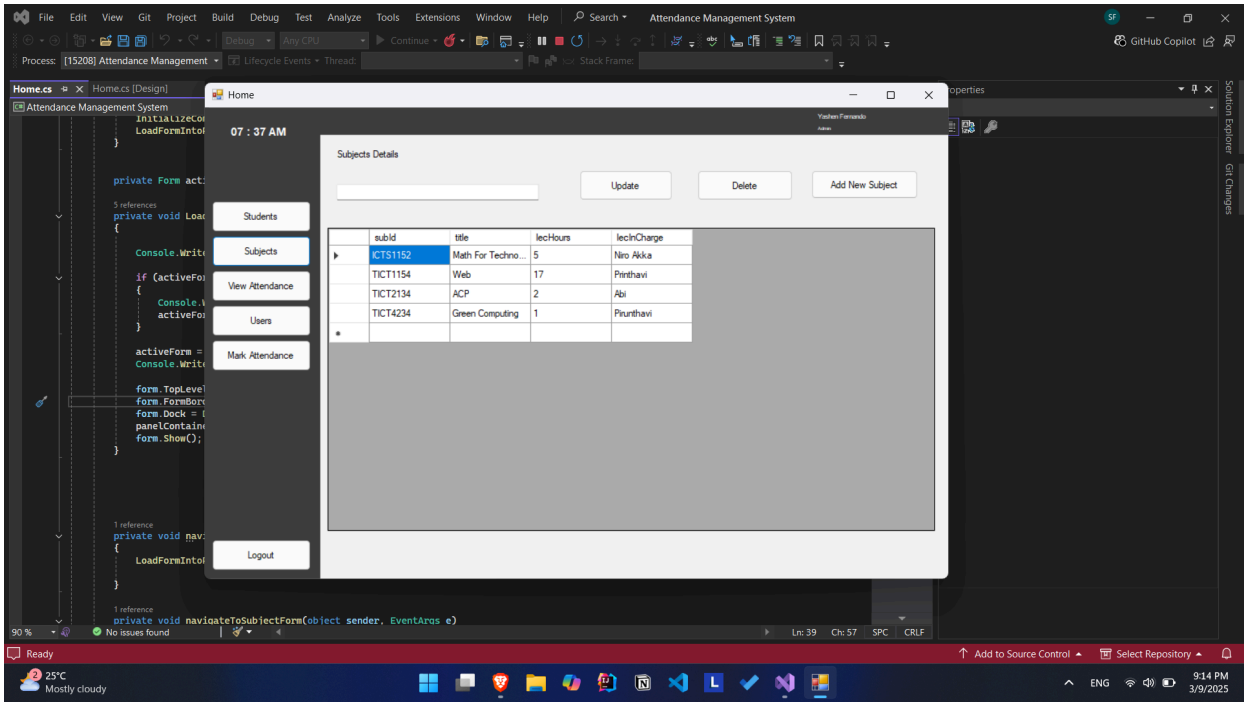
- Insert: Users enter subject details in the text boxes and click "Add Subject" to save the new subject to the database.
- Update: Users select a subject from the list, modify its details in the text boxes, and click "Update" to save changes.
- Delete: Users select a subject from the list and click "Delete" to remove it, with a confirmation prompt for safety.
- View: The list view automatically refreshes to display all subjects after each action, pulling data from the SQL Server database.

Integration:

- The form is embedded into the Home Screen’s container panel when the "Subject Management" menu button is clicked, ensuring a seamless navigation experience.
- Data is stored and retrieved using the service-based SQL Server database via .NET Framework.

The Subject Form combines simplicity and functionality, enabling users to manage subjects efficiently within a modern UI design.

**04. Subject Menu**




The Subject Screen is a dedicated interface within the Smart Attendance System for managing subject information. It allows users to insert, update, delete, and view subject details efficiently. This screen is loaded into the Home Screen's container panel when the "Subject Management" menu button is clicked, featuring a modern UI design consistent with the application.

Components:

1. Subject Input Fields:
    - Subject Name Text Box: A field to input or edit the subject's name (e.g., "Physics").
    - Subject Code Text Box: A field to enter or modify a unique subject identifier (e.g., "PHY201").
    - (Optional: Additional fields like "Teacher Name" or "Class" can be included.)
2. Action Buttons:
    - Insert Button:
        - Labeled "Add Subject".
        - Saves a new subject to the SQL Server database after ensuring all required fields are filled.

    - Update Button:
        - Labeled "Update".
        - Modifies the details of an existing subject in the database based on user edits.
    - Delete Button:
        - Labeled "Delete".
        - Removes the selected subject from the database, with a confirmation prompt (e.g., "Confirm deletion?").
    - Clear Button (Optional): Resets the input fields for a fresh entry.
3. Subject Display Area:
    - A table or grid (e.g., DataGridView) showing all subjects stored in the system.
    - Columns: Subject Name, Subject Code, and (optionally) additional details.
    - Users can click a row to load its details into the input fields for editing or deletion.

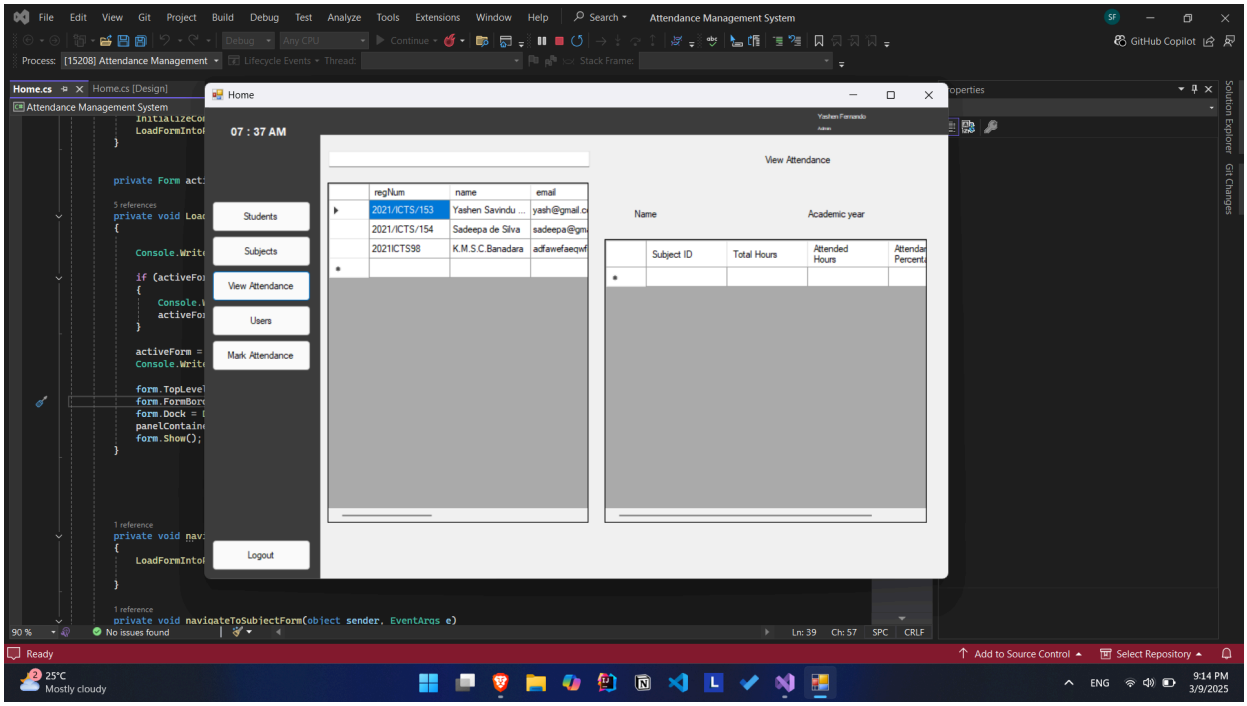
Functionality:

- Insert: Users fill in the subject name and code, then click "Add Subject" to store the new subject in the database.
- Update: Users select a subject from the display area, edit its details in the text boxes, and click "Update" to save changes.
- Delete: Users select a subject and click "Delete" to remove it, with the system confirming the action to prevent accidental deletion.
- View: The display area automatically updates after each action, fetching and showing the latest subject list from the SQL Server database.

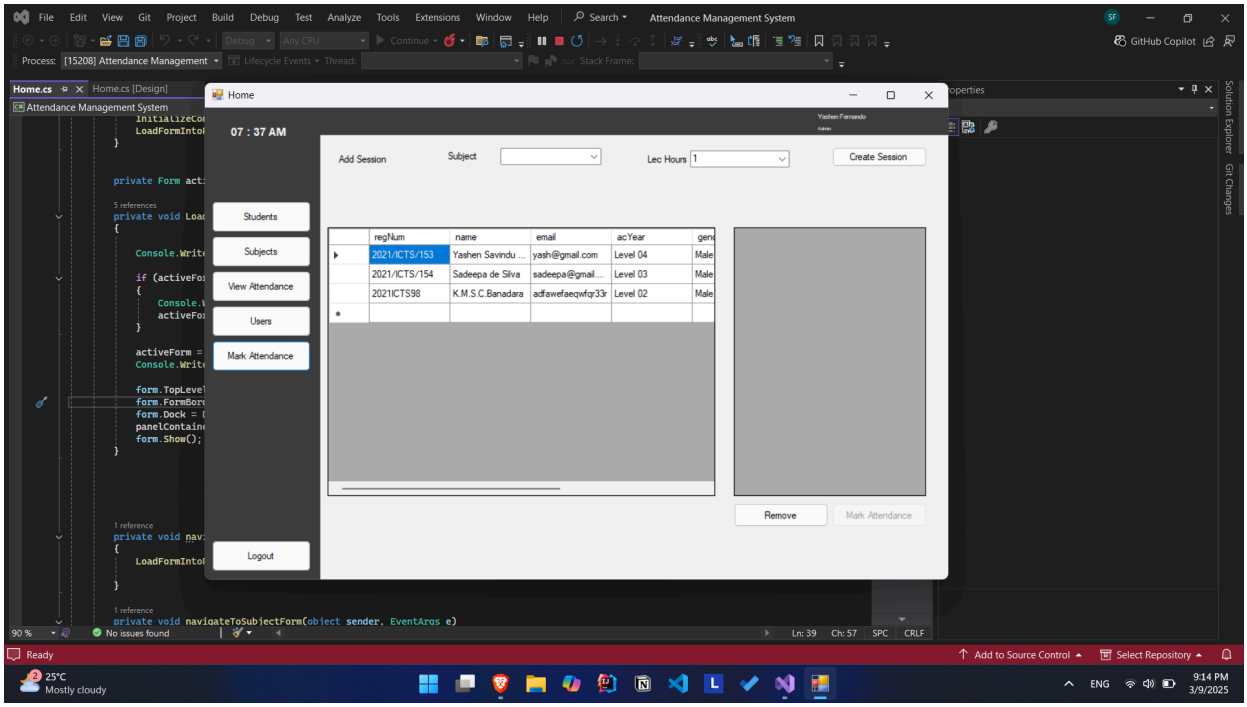
Integration:

- The Subject Screen is dynamically loaded into the Home Screen's container panel when the "Subject Management" menu button is clicked.
- It interacts with the service-based SQL Server database via the .NET Framework for data operations.

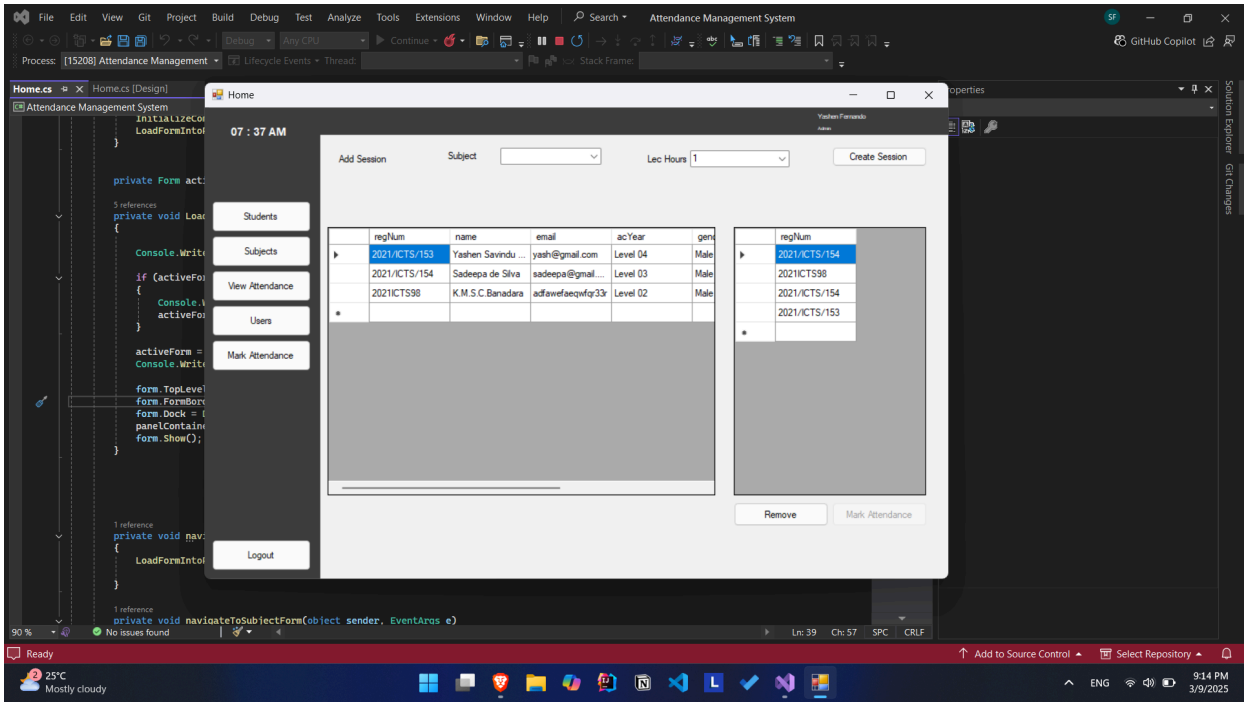
**05. View Attendance**



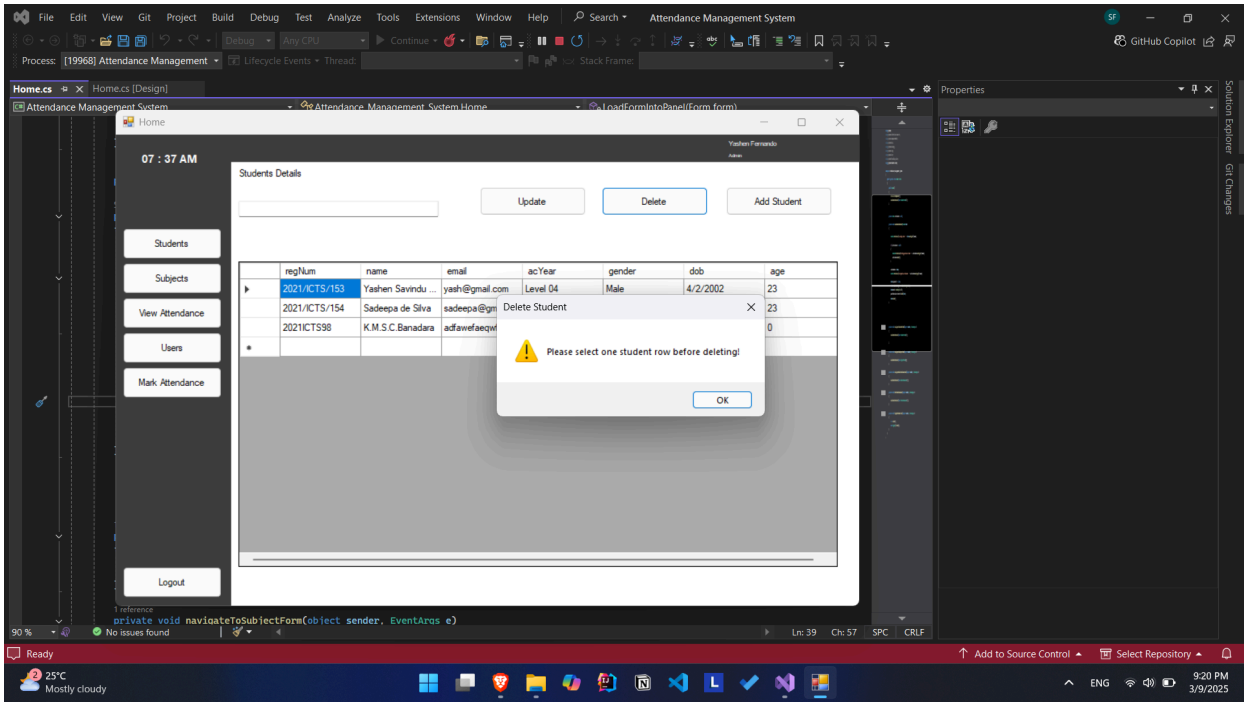

## 06. Create Session



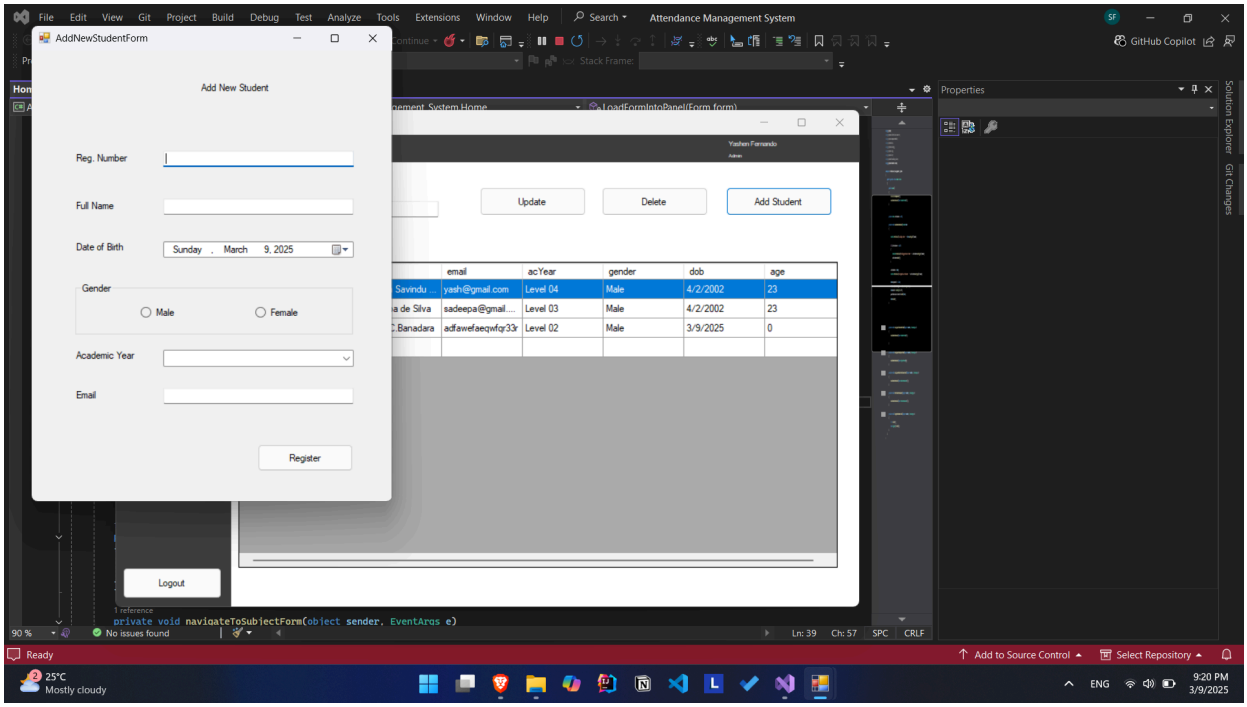
## 07. Mark Attendance



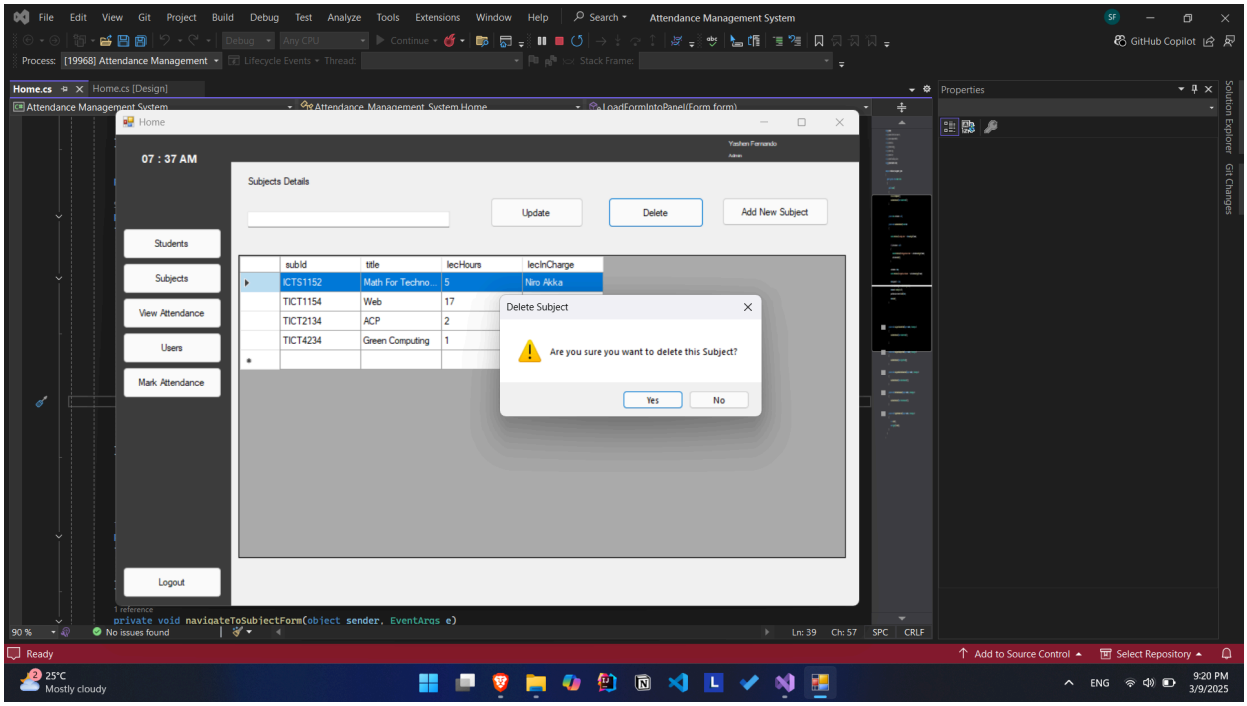

## 08. Delete Student



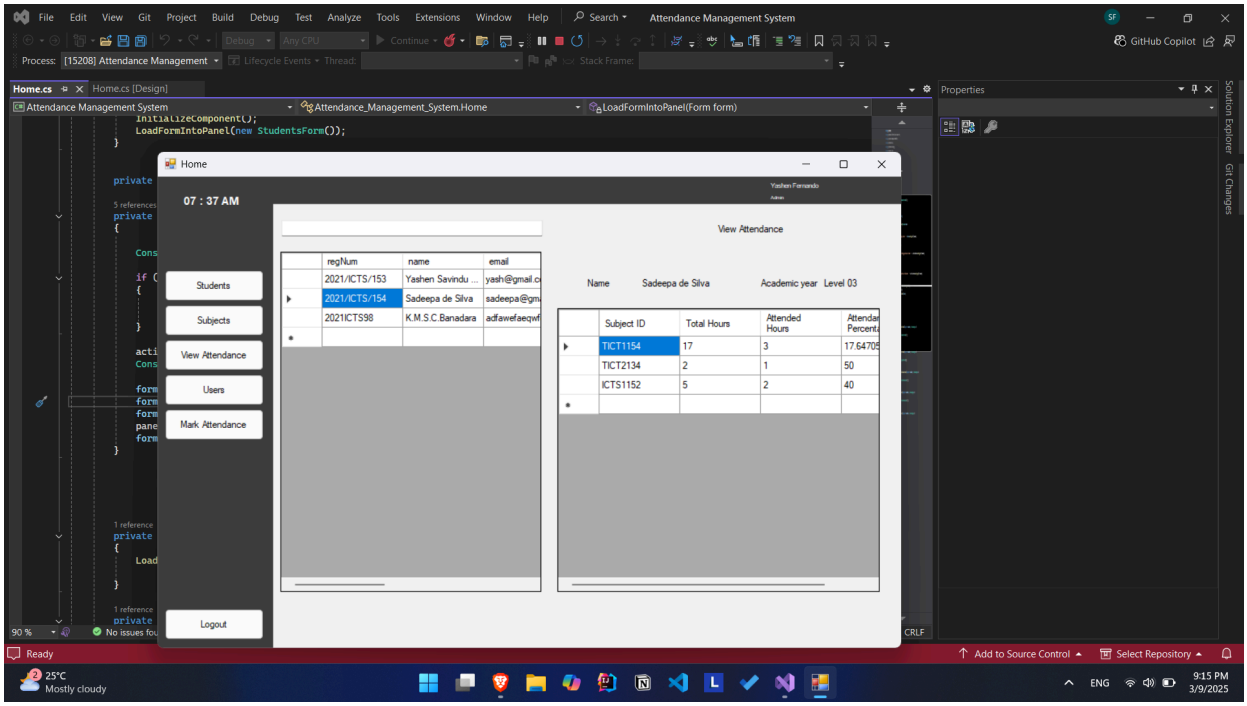

## 09. Add Student




## 10. Delete Subject




## 11. Show Attendance Percentage



## Challenges and Solutions: Smart Attendance System Project

### 1. Challenge: Database Not Working Properly

- Issue: The service-based SQL Server database encountered connectivity or functionality issues, preventing proper data storage and retrieval.
- Impact: Hindered the ability to manage students, subjects, and attendance records effectively.
- Solution: Updated Visual Studio and the .NET Framework to their latest versions. This resolved compatibility issues between the development environment and the database, ensuring smooth interaction with the SQL Server. Post-update, the database operations (insert, update, delete) worked seamlessly.

### 2. Challenge: Menu Screen Form Loading Issue

- Issue: Loading forms into the container panel on the Home Screen failed multiple times, resulting in either blank panels or runtime errors.
- Impact: Disrupted navigation and the single-window experience intended for the application.
- Solution: Researched solutions on StackOverflow and other .NET-related websites. Discovered a working approach involving proper instantiation and docking of forms into the panel using .NET Framework methods (e.g., setting TopLevel = false and Dock = DockStyle.Fill). After iterative testing, the forms loaded correctly into the container panel.

### 3. Challenge: Attendance Percentage Update with SMS Integration

- Issue: Planned to update students' attendance percentages and notify via SMS, but encountered problems because SMS providers required paid packages.
- Impact: Limited real-time notifications, affecting the system's ability to communicate attendance updates instantly.
- Solution: Due to budget constraints, deferred SMS integration and focused on in-app percentage calculation instead. Developed a robust algorithm within the .NET application to compute and display attendance percentages accurately using database records. For future enhancements, identified free-tier SMS APIs or email notifications as cost-effective alternatives.

---

## Summary

The Smart Attendance System project faced technical hurdles related to database functionality, form loading, and SMS integration. By leveraging updates to development tools, community resources like StackOverflow, and prioritizing core functionality over paid services, the team successfully overcome these challenges. These solutions ensured the application remained functional, user-friendly, and aligned with the project's goals.

**Reference** - https://github.com/Yashenf/Attendance-management-system