# PROJECT ON

**Institute Management System**

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

****

#### SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

Hinjewadi

**SUBMITTED BY:**

DHRUV GARG

**UNDER THE GUIDENCE OF:**

Snehal Jadav

Faculty Member

Sunbeam Institute of Information Technology, Pune

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right person at the right time. We avail ourselves of this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT, Pune).

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune, for their support.

DHRUV GARG

0225 PG-DAC

SIIT Pune

**CERTIFICATE**

This is to certify that the project work under the title ‘Institute Management System’ is done by **DHRUV GARG** in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

**Mr. Yogesh Kolhe**

**Project Guide Course Co-Coordinator**

Date: 11th August 2025

# INTRODUCTION TO PROJECT

The web-based "Institute Management System" project is an attempt to automate and streamline the core operations of an educational institute. The system enables different stakeholders — Admin, Teacher, and Student — to manage academic and administrative activities efficiently through a unified digital platform.

The system provides role-based access:

Admin can manage students, teachers, courses, attendance, marks, fees, notices, subjects, and complaints.

Teacher can view student details, attendance, marks, courses, and institute notices.

Students can access their profile, view attendance and fees, give complaints, and check notices.

The system simplifies tasks such as adding new students or teachers, recording attendance, managing marks, and publishing notices. It ensures real-time data availability so that all stakeholders can make informed decisions.

The ‘Admin’ dashboard allows quick access to all institute data and operations without the need for manual record-keeping. Teachers can directly update or check academic records, while students have a self-service portal to track their academic progress and institute updates.

For example, the Student Module allows a student to log in and view personal details, attendance records, fee status, marks, and institute notices. Similarly, the Admin Module enables quick student enrollment, teacher assignment, subject allocation, and complaint resolution.

Overall, the system reduces paperwork, improves efficiency, and keeps centralized, secure records for smooth institute management operations.

**2.REQUIREMENTS**

**2.1 FUNCTIONAL REQUIREMENTS**

A diagram of a person's diagram

AI-generated content may be incorrect.

**2.1 Admin Functionality**

1. Student Management

One of the core responsibilities of the Admin in the Institute Management System is student management, which includes adding, displaying, checking attendance, managing marks, and updating fee status. The Add Student functionality allows the admin to enroll a new student by entering complete details such as name, email, date of birth, address, course, and image. This ensures that each student has a unique, verified profile in the system. Once registered, the admin can use the Display All Students feature to view a complete list of all students currently enrolled in the institute. This list acts as a centralized directory, making it easier to search for specific students or verify their details.

The See All Student Attendance feature enables the admin to access the complete attendance records of all students in the system. This is crucial for checking participation and finding cases of irregular attendance. Additionally, the Student Marks functionality allows the admin to view academic performance for each student across subjects and courses. These centralized marks help in academic evaluation and report preparation. Lastly, the admin can Change the Status of Student’s Fees to reflect whether fees are paid, pending, or overdue. By updating fee statuses in real-time, the institute can keep correct financial records and take prompt action about defaulters.

2. Teacher Management

The admin is also responsible for teacher-related data management, ensuring that faculty records are correct and up to date. Using the Add Teacher functionality, the admin can register a new teacher by entering relevant details, creating a digital profile for them in the system. Once registered, the Display All Teachers feature enables the admin to view a complete list of faculty members in the institute. This list is essential for administrative purposes such as distribution workload, departmental assignments, and communication.

The Check Teacher’s Attendance functionality gives the admin a clear view of the attendance records of all faculty members. Monitoring teacher attendance is essential to ensure accountability, keep academic schedules, and find any gaps in the delivery of classes. With these features, the admin ensures that faculty data is always well-maintained and accessible.

3. Course and Subject Management

The admin also plays a vital role in shaping the academic offerings of the institute through the Course and Subject Management functionalities. The Add Courses feature allows the admin to introduce new academic courses into the system. Each course can be defined and registered to ensure it appears in the institute’s official offerings. Once courses are created, the Display All Courses functionality provides a combined list of all active courses, enabling easy reference for both internal and external use.

Similarly, the admin has control over the subject database. Using the Add New Subjects feature, the admin can register new subjects under specific courses. This ensures that academic curricula stay updated and relevant. The Display All Subjects feature presents a complete list of subjects in the system, allowing for quick access and verification. Together, these course and subject management functionalities ensure that the institute’s academic structure stays organized, well-documented, and adaptable to changing educational needs.

4. Notice and Communication Management

Communication within the institute is managed effectively through the admin’s ability to manage notices. Using the Add New Notices functionality, the admin can publish important announcements or updates to both students and teachers. These notices can cover a wide range of topics, such as academic schedules, exam dates, event notifications, fee reminders, or policy changes. The Display All Notices functionality allows the admin to review all active notices in the system, ensuring that information stays relevant and outdated notices can be found for removal or updating.

By controlling the notices, the admin ensures a centralized communication channel where all stakeholders can access authentic, institute-approved information. This reduces the risk of miscommunication and ensures that both staff and students stay informed about important developments in real time.

5. Complaint Management and Overall Administrative Control

Finally, the admin has responsibility for managing complaints given by students. The Display All Complaints functionality gives the admin access to the complete list of complaints, enabling them to review and prioritize cases that require attention. Each complaint can then be marked as either Active or Resolved, offering a clear tracking mechanism for grievance redressal. This structured approach to complaint managing not only increases transparency but also improves trust between students and the administration.

Collectively, all these functionalities empower the admin to have complete oversight and control over the institute’s operations. By integrating student, teacher, course, subject, notice, and complaint management into one centralized platform, the Institute Management System cuts the need for manual records and reduces administrative workload. The admin, through these tools, ensures operational efficiency, prompt communication, correct record-keeping, and a smooth academic experience for all stakeholders. This comprehensive control makes the admin role the central pillar of the system’s effectiveness and reliability.

**2.2 Student Functionalities**

A diagram of a student function

AI-generated content may be incorrect.

1. Profile Management

One of the primary functionalities available to the Student in the Institute Management System is the ability to view their own profile. Upon logging into the system, the student can access a dedicated profile section that has all personal and academic details stored in the database. This may include their full name, date of birth, contact information, email address, course enrolled, gender, and admission date. Having direct access to their profile allows students to verify that their information is correct and up to date. Since the profile acts as the foundation for other modules such as attendance, fees, and performance, it ensures that all records are linked to the correct student. The availability of the profile module reduces the need for students to contact the administration for basic details, as they can check them anytime through the system. This promotes transparency and provides students with a sense of ownership over their academic records.

2. Complaint Management

The student module also provides a channel for grievance redressal through the Add Complaint and Display Complaints functionalities, combined into a single workflow. If a student faces any academic, administrative, or facility-related issue, they can make a complaint through the system. The complaint form allows them to specify the issue, provide relevant details, and give it directly to the administrative team. Once gave, the complaint becomes part of the central complaints database, accessible to the admin for review. Through the Display Complaints option, the student can check the status of their submitted grievances — whether they are Active (pending resolution) or resolved. This transparent tracking system builds trust and ensures that students are aware of how their issues are being managed. The ability to both gives and track complaints from a single platform cuts the need for physical follow-ups and streamlines communication between students and the administration.

3. Attendance Viewing

Another key feature of the student module is the View Attendance functionality. This allows students to check their attendance records for all subjects or courses they are enrolled in. The attendance records display data such as total classes conducted, number of classes attended, and overall attendance percentage. By having real-time access to this information, students can find attendance shortfalls and take corrective action before it affects their eligibility for exams or internal assessments. This feature also promotes accountability, as students can compare their attendance with academic requirements set by the institute. Since the data is kept and updated by the faculty or admin, it ensures accuracy and cuts disputes over attendance records. The integration of attendance viewing into the student’s dashboard makes it a convenient, self-service tool for academic tracking.

4. Fee Status Viewing

The View Fees functionality allows students to check the status of their tuition or other institute-related payments. This section displays whether the fees have been paid, are pending, or overdue. Along with the status, students may also see details such as the total amount due, payment dates, and transaction history. Having this information readily available ensures that students stay informed about their financial obligations to the institute. It also reduces confusion about fee deadlines and payment confirmations, as the system serves as an authoritative source of truth. By being aware of their fee status, students can plan payments, accordingly, avoid late fees, and ensure uninterrupted access to academic services. This digital access to financial records is far more efficient than relying on manual queries to the administration.

5. Performance Statement Viewing

One of the most academically expressive features for students is the View Performance Statement functionality. This section allows students to see their marks and overall performance in various subjects or courses. The performance statement can display both individual subject scores and aggregate results, offering a comprehensive view of academic progress. In addition to raw scores, the performance section may also present calculated totals, percentages, and grades, making it easier for students to interpret their results. Having direct access to this data helps students find strengths and weaknesses, enabling them to take corrective measures in future assessments. This feature not only supports self-assessment but also aids in preparing for examinations, applying for scholarships, or taking part in academic competitions. Since the performance data is updated by the faculty or admin, it ensures reliability and authenticity.

**2.3 Teacher Functionality**

A diagram of a teacher

AI-generated content may be incorrect.

1. Viewing All Students

One of the core functionalities available to the Teacher in the Institute Management System is the ability to display all students. Through this feature, a teacher can view a comprehensive list of students enrolled in the institute, irrespective of the course or department. The list provides essential details such as student ID, name, course name, email, date of birth, and gender. This centralized view helps teachers quickly find students for academic, administrative, or counseling purposes. In cases where a teacher is managing multiple classes or subjects, having access to a unified student directory significantly reduces the time spent searching for individual records. It also ensures that teachers can familiarize themselves with their students beyond just the ones they directly teach. This database-style listing is particularly useful for collaborative activities, inter-class competitions, or when substituting for another teacher. By combining all student information into one easily accessible module, the system improves efficiency and cuts the need for manual roll calls or paper-based student lists.

2. Accessing Student Attendance Records

The See All Student Attendance functionality allows teachers to check and analyze attendance patterns for students across the institute. This feature presents detailed attendance records, including the total number of classes conducted, classes attended by each student, and the calculated attendance percentage. Teachers can filter and search for attendance data to focus on specific courses, batches, or individual students. By having access to attendance data, teachers can find students who are irregular, intervene early, and offer necessary guidance to help improve their participation. This module is especially helpful for academic evaluation, as attendance is often a factor in internal assessments and exam eligibility. Furthermore, teachers can use this data to provide correct feedback during parent-teacher meetings or counseling sessions. The availability of attendance information in a structured digital format ensures transparency, reduces disputes, and enables data-driven decision-making for improving student engagement.

3. Viewing Student Marks

Another important academic function available to teachers is the Student Marks module. This allows teachers to access the marks obtained by students in various subjects and assessments. The system presents detailed records that may include marks per subject, total marks, obtained marks, percentages, and grades. Having this data centralized allows teachers to evaluate overall academic performance and find high performance as well as struggling students. By analyzing this information, teachers can adjust their teaching strategies, offer remedial classes, or provide extra challenges for advanced learners. This feature also helps in preparing performance reports, communicating academic results to students, and making recommendations for awards or scholarships. Since the marks are stored digitally, the data stays correct, easily retrievable, and accessible for long-term tracking of academic progress. This module not only saves time compared to manual grade book maintenance but also ensures secure and consistent record-keeping.

4. Viewing Other Teachers and Courses

The Teacher role in the system also includes access to Display All Teachers and Display All Courses functionalities. The Display All Teachers module provides a complete directory of faculty members within the institute, along with details such as name, department, contact information, and subjects taught. This functionality helps in fostering collaboration between faculty members, organizing departmental activities, or coordinating substitute classes when needed. It also serves as a quick reference for new teachers to familiarize themselves with colleagues across different departments.

The Display All Courses feature enables teachers to view all the courses currently offered by the institute. This list typically includes course names, duration, syllabus outline, and related subjects. Access to this information allows teachers to understand the broader academic framework of the institute and ensures that their subject delivery aligns with the overall curriculum. It also helps in finding opportunities for cross-departmental collaboration, interdisciplinary teaching, and curriculum improvements.

5. Viewing Notices and Staying Updated

The Display All Notices functionality ensures that teachers stay informed about important announcements, events, and administrative updates. This centralized noticeboard includes information such as exam schedules, staff meetings, training sessions, academic policy updates, and institute events. By accessing this section, teachers can stay aligned with institutional requirements and ensure they communicate relevant information to students in their classes. The availability of notices in a digital format ensures that information is given quickly and uniformly to all teachers, reducing the risk of miscommunication. It also creates a permanent record of announcements, which can be referred to later if needed. This feature supports smooth communication between the administration and teaching staff, enabling better planning and coordination of academic and extracurricular activities.

* 1. **NON-FUNCTIONAL REQUIREMENTS**

**2.2.1 Interface**

Go to Appendix B for user interfaces.

**2.2.2 Performance**

* **Number of Concurrent Users:**

IMS shall be able to manage at least 1000 transactions/inquiries per

second

**2.2.3 Constraint**

IMS shall be able to manage at least 1000 transactions/inquiries per second.

**2.2.4 Other Requirements:**

* **Hardware Interfaces**

The SPMS is expected to function on Intel PIII 900 MHz Processor equivalent or above, 128 MB RAM, 20 GB HDD.

* **Software Interfaces**

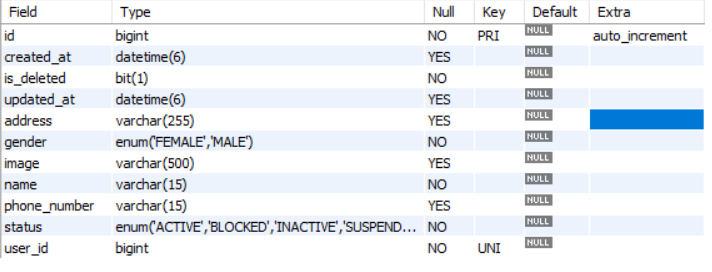
The SPMS shall work on MS Windows operating systems family (MS Windows 98, MS Windows NT Workstation, MS Windows 2000, MS Windows XP). It configures to work with Oracle database. This System works on Apache Tomcat server. It uses browser IE 5.0 & above. It uses IIS 5.0 server.

**3. DESIGN**

**3.1 Database Design**

The following table structures depict the database design.

**Table 1: admin**



## **Table2: Attendance**

A screenshot of a computer

AI-generated content may be incorrect.

# Table 3: Course

A screenshot of a computer

AI-generated content may be incorrect.

### Table 4: course\_subject\_teacher

A screenshot of a computer

AI-generated content may be incorrect.

**Table 5: Fee**

### A screenshot of a computer AI-generated content may be incorrect.

### Table 6: login

A screenshot of a computer

AI-generated content may be incorrect.

**Table 7: marks**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Table 8: notice**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Table 9: student\_complaints**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Table 10: students**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Table 11: subjects**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Table 12: teachers**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**E-R Diagram, Dataflow diagram and Class Diagram:**

Go to Appendix A

**4. CODING STANDARDS IMPLEMENTED**

### Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e., capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **Case** | **Examples** | **Additional Notes** |
| Class | Pascal | Admin, Students, Teachers | Class names should be based on "objects" or "real things" and should generally be **nouns**. No \_’ signs allowed. Do not use type prefixes like ‘C’ for class. |
| Method | Camel | getStudents, getName | Methods should use **verbs** or verb phrases. |
| Parameter | Camel | address, phoneNumber | Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to decide its meaning in most scenarios. |
| Interface | Pascal with "I" prefix | Disposable | Do not use the \_’ sign |
| Property | Pascal | ForeColor, BackColor | Use a noun or noun phrase to name properties. |
| Associated private member variable. | \_camelCase | \_foreColor, \_backColor | Use underscore camel casing for the private member variables |
| Exception Class | Pascal with "Exception" suffix | WebException, |  |

### Comments

* Comment on each type, each non-public type of member, and each region’s declaration.
* Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.
* Separate comments from comment delimiters (apostrophe) or // with one space.
* Begin the comment text with an uppercase letter.
* End the comment with a period.
* Explain the code; do not repeat it.

**5. TEST REPORT**

**Another group called Linux did the testing and the report of the testing is given hereunder.**

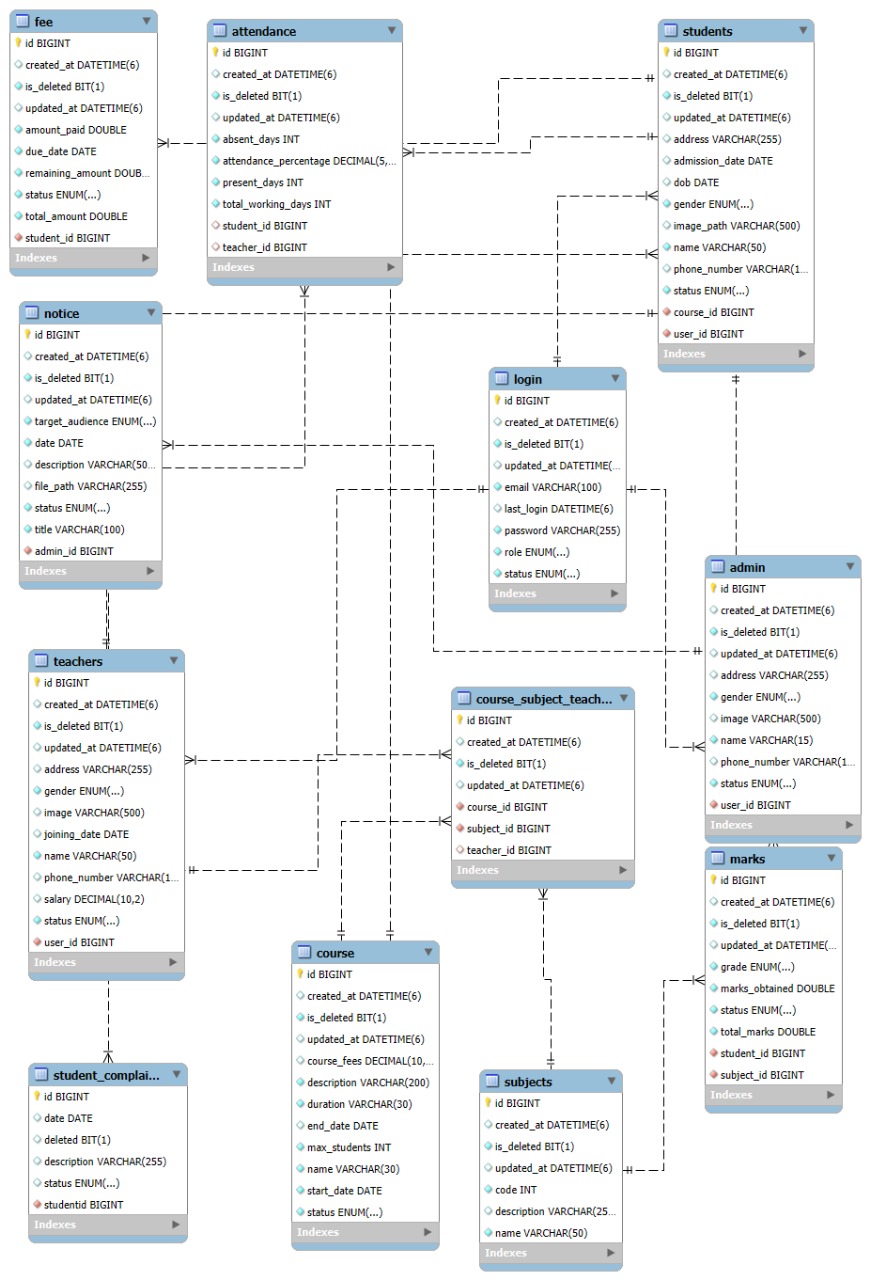
**GENERAL TESTING:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR-NO** | **TEST CASE** | **EXPECTED RESULT** | **ACTUAL RESULT** | **ERROR MESSAGE** |
| 1 | Register Page | Redirected to Next page | OK | Nothing |
| 2 | Login Page | Pop-up will come | Ok | Please enter username and password again. |
| 3 | Dashboard | Count of courses, students, teachers, latest notices, details of topper | Ok | Nothing |
| 4 | Student functionalities | Redirect to next: Add, display, attendance, marks, fees of students | Ok | Nothing |
| 5 | Teacher Functionalities | Redirect to next: Add, display, attendance of teachers | Ok | Nothing |
| 6 | Course | Redirect to next: Add, display all courses | Ok | Nothing |
| 7 | Notices | Redirect to next: Add, display all notices | Ok | Nothing |
| 8 | Subjects | Redirect to next: Add, display all subjects | Ok | Nothing |
| 9 | Complaints | Redirect to next: Display and update all complaints | Ok | Nothing |
| 10 | Profile | View, update profile | Ok | Nothing |
| 11 | Marks | Redirect to next: Add, update student marks | Ok | Nothing |
| 12 | Logout | It will log out from user profile. | Ok | Nothing |
|  | **STATIC TESTING** |  |  |  |
| **SR-NO** | **Deviation** | **Program** |  |  |
| 1 | Commenting not followed | All Web Application |  |  |

**6. PROJECT MANAGEMENT RELATED STATISTICS**

|  |  |  |  |
| --- | --- | --- | --- |
| **DATE** | **WORK PERFORMED** | ****SLC Phase**** | **Additional Notes** |
| JULY 21,2025 | Project Allotment and User Requirements Gathering | Feasibility Study | Our team met the client Mr. Nitin kudale (CEO, SIIT Pune) to know his requirements. |
| JULY 22,2025 | Initial SRS Document Validation  And Team Structure Decided | Requirement Analysis  (Elicitation) | The first SRS was presented to the client to understand his requirements better |
| JULY 23,2025 | Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces | Requirement Analysis &  Design Phase | Database Design completed. |
| JULY 24,2025 | Business Logic Component Design Started | Design Phase | ---------------------- |
| JULY 25-26,2025 | Coding Phase Started | Coding Phase | 70% of Class Library implemented. |
| JULY 27-28,2025 | Implementation of Web Application and Window Application Started | Coding Phase | Class Library Development going on. |
| JULY 29,2025 | Off | Off | Off |
| JULY 30-31,2025 | Implementation of Web Application and Window Application Continued | Coding Phase and Unit Testing | Class Library Modified as per the need. |
| AUG 1-3,2025 | Implementation of Web Application and Window Application Continued | Coding Phase and Unit Testing | -- |
| AUG 4-5,2025 | After Ensuring Proper Functioning the Required Validations were Implemented | Coding Phase and Unit Testing | Module Integration was done by the Project Manager |
| AUG 6 ,2025 | The Project was Tested by the respective Team Leaders and the Project Manager | Testing Phase (Module Testing) | -- |
| AUG 7,2025 | The Project was Submitted to Other Project Leader of Other Project Group for Testing | Testing Phase (Acceptance Testing) | The Project of Other Team was Taken up by the Team for Testing |
| AUG 8-10,2025 | The Errors Found were Removed. | Debugging | The Project was complete for submission |
| AUG 11,2025 | Final Submission of Project |  |  |

Appendix A

Entity Relationship Diagram*`*

**Data Flow Diagram:**

A diagram of a person

AI-generated content may be incorrect.

Appendix B

Login Page:

A screenshot of a computer screen

AI-generated content may be incorrect.

**Admin Dashboard**

A screenshot of a computer

AI-generated content may be incorrect.

**Add Student**

A screenshot of a computer

AI-generated content may be incorrect.

**Student List**

A screenshot of a computer

AI-generated content may be incorrect.

**Student Attendance**

A screenshot of a computer

AI-generated content may be incorrect.

**Add teacher.**

A screenshot of a computer

AI-generated content may be incorrect.

**Teacher List**

A screenshot of a computer

AI-generated content may be incorrect.

**Add Courses**

A screen shot of a computer

AI-generated content may be incorrect.

Add Notices

A screenshot of a computer

AI-generated content may be incorrect.

Add Subjects

A screenshot of a computer

AI-generated content may be incorrect.

Display all subjects.

A screenshot of a computer

AI-generated content may be incorrect.

Logout

A screenshot of a computer

AI-generated content may be incorrect.

**7.REFERENCES:**

1. <http://www.google.com>
2. http://www.webdevelopersjournal.com/
3. http://www.w3.org
4. http://www.wikipedia.org
5. <https://www.syncfusion.com>
6. <https://telusko.com>