# **Week 1 Documentation: IIT Dharwad Attendance System**

## **1. Overview**

The Student Attendance System is a web-based platform designed to streamline the process of tracking student attendance in educational institutions. Teachers can mark attendance, students can view their attendance history, and administrators can manage class data and generate attendance reports. The system is designed with role-based access control, ensuring that students, teachers, and administrators have access only to the features they need.

### **Key Features:**

1. *Teacher Dashboard*: Allows teachers to mark attendance with barcode scanning for their classes.
2. *Student Dashboard*: Enables students to view their attendance history.
3. *Admin Dashboard*: Provides administrators with tools to manage class data and generate attendance reports.
4. *Role-Based Login*: Differentiated access for students, teachers, and administrators.
5. *API-Driven Architecture*: Backend APIs will handle login, attendance records, and report generation.

## **2. Problem Statement**

*Problem*: Managing student attendance is often a tedious and manual process in schools, especially in large classes. Teachers need a quick and easy way to mark attendance, while students should have access to their attendance history in real-time. Administrators or Teachers require reports to track absenteeism trends and manage student data efficiently.

\*Solution: A \*Student Attendance System that allows teachers to mark attendance in real-time, while providing students and administrators access to attendance records and reports. The system also includes predictive analytics to help identify students at risk of absenteeism.

## **3. Dataset**

### **Custom Dataset Structure:**

To simulate real-world functionality, the following dataset will be created to use throughout the project. A sample dataset might also be derived from a Kaggle student attendance dataset, modified to fit this system's requirements or we can create our custom dataset.

1. *Students*: Information about students including name, ID, class, and other relevant details.
2. *Classes*: Information about each class, including class name, teacher, and enrolled students.
3. *Attendance Records*: Daily records that indicate whether each student attended or missed a class.
4. *Teachers*: Details about the teachers, including name, ID, and the classes they teach.

*Example*:

1. *Students Table*:  
   student\_id: Unique identifier for each student.  
   name: Full name of the student.

2. Enrolls (Student course enrollment table)   
 student \_id : Unique identifier for each student.  
 class\_id: Identifier for the class the student is enrolled in.

3. Course Table:  
 course\_code: Unique identifier for each class.  
 course\_name: Name of the class.  
 credits :Course credit.

4. Teachers table  
 teacher\_id: Identifier for the teacher in charge of the class.  
 teacher\_name

5. Attendance Table:  
 student\_id: Identifier for the student.  
 course\_code: Identifier for the class.  
 attendance\_date: Date of the attendance record.

## **4. Content Structure**

### **Pages and Navigation**

The Student Attendance System will consist of multiple pages, each with specific functionality based on the user role (Student, Teacher, Admin).

1. *Login Page*:  
   Role-based login for students, teachers, and admins.  
   Secure authentication to prevent unauthorized access.  
     
   
2. *Student Dashboard*:   
   - Enroll Courses  
   - Enrolled Courses  
    - Attendance History  
    - Filter Attendance Record by date or class  
   - View Weekly/Monthly Analysis Report and Statistics  
   - Student Profile  
   
3. *Teacher Dashboard*:  
   - View assigned courses  
    - mark attendance for students.  
   - View/update past attendance records  
   - View Report and Statistics  
   - Teacher Profile  
     
   
4. *Admin Dashboard*:  
   - Admin Profile  
   - Student  
    - Create/Update/Edit Student Data.  
   - Teachers   
    - Create/Update/Edit Teacher Data  
   - Courses  
    - Create/Update/Edit Course Data  
   - Classes  
    - Enroll Student to Course  
    - Assign Teacher to Course



## **5. Functionality**

### **Teacher Role:**

1. *Mark Attendance*: Teachers can select their class and mark students as present or absent. Attendance can be recorded efficiently using a barcode scanner for student ID cards. If there are any issues with the camera, teachers have the option to switch to manual attendance entry, ensuring flexibility in attendance management.
2. *View Attendance*: Teachers can view past attendance records, allowing them to correct any errors and ensure accurate reporting.
3. *View Report* : Teachers can access detailed Course Attendance Analysis, including statistics on attendance trends and patterns. This section will also provide machine learning (ML) suggestions for identifying students at risk of absenteeism, enabling proactive intervention.

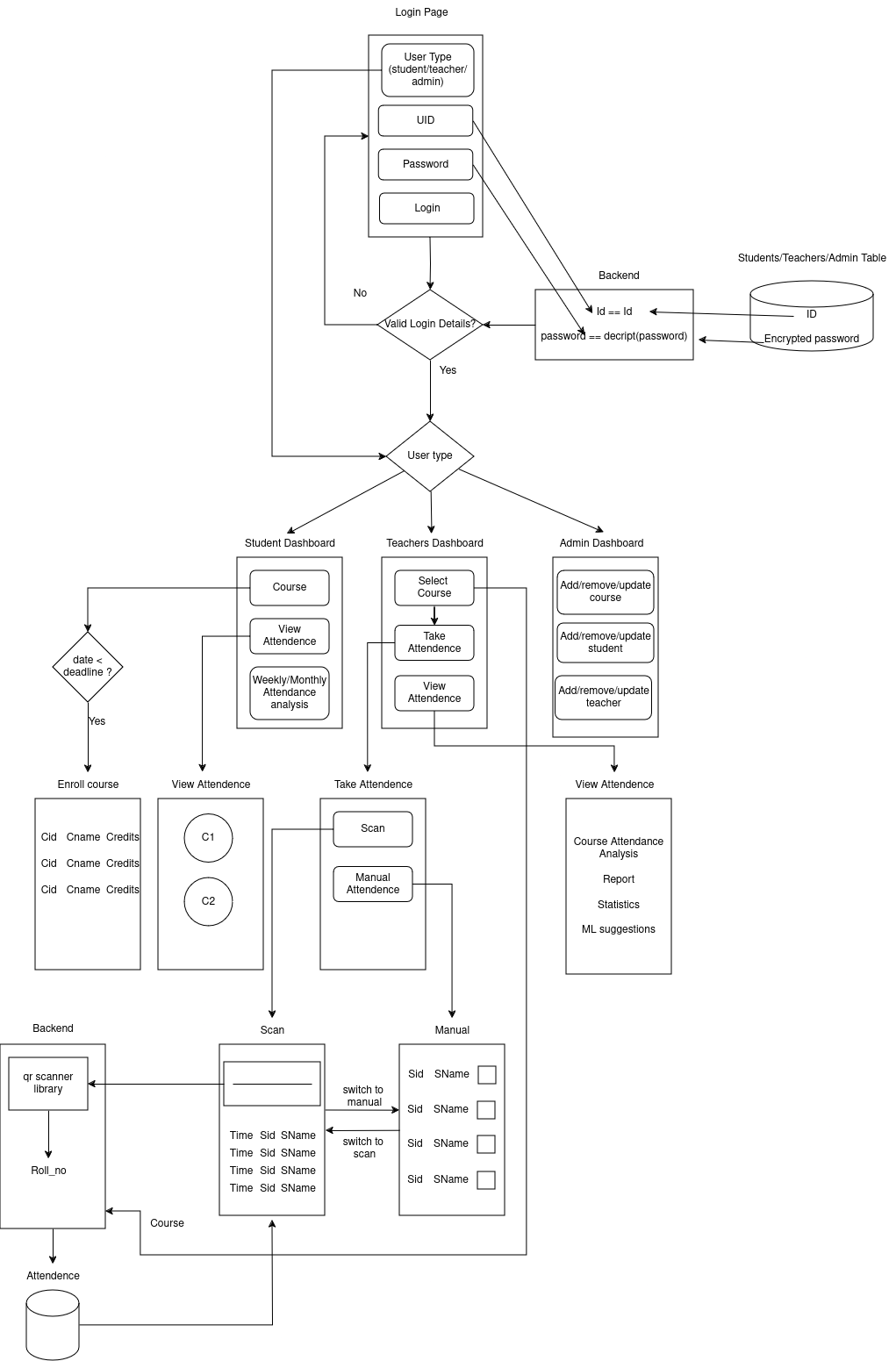
### **Student Role:**

1. *View Attendance History*: Students can view their attendance history for all classes, organized by date or subject, promoting transparency and accountability.
2. *View Report* : Students can access personalized attendance reports that summarize their attendance patterns over time, highlighting any concerns or trends. Reports may include statistics such as total days present, absent, and any notable trends in attendance, along with recommendations for improving attendance based on historical data.

### **Admin Role:**

1. *Generate Reports*: Admins can generate reports for absenteeism trends, including identifying students with consistent absences.
2. *Manage Users and Classes*: Admins can add, update, or delete students, teachers, and classes from the system.

**6. Flow Diagram :**



## **7. Technical Considerations**

### **Backend Architecture:**

1. ***Database***: A relational database (e.g., MySQL) will store student, class, teacher, and attendance information. The database will be structured to efficiently manage relationships between these entities, ensuring data integrity and performance.
2. ***APIs***: The system will use RESTful APIs to enable seamless communication between the front-end and back-end. The APIs will handle core functionalities such as login authentication, attendance management, and report generation, ensuring secure and efficient data transfer.
3. **Attendance Recording**: The system will scan the barcode from the student ID card to record attendance. The scanned data will then be matched with the appropriate student and logged into the designated course table within the database.

### **Frontend Design:**

1. The front-end will be designed in HTML, CSS, and JavaScript with Django as the web framework to build dynamic and interactive user interfaces.
2. Figma will be used to prototype the UI/UX for the system.

## **8. Future Considerations:**

### **ML Integration:**

In future weeks, we plan to integrate ML APIs for predictive analytics. This will enhance the system by allowing it to do much more than just track attendance. For example, the ML models can analyze historical attendance data to predict absenteeism trends, enabling early interventions for students at risk of chronic absenteeism. Additionally, ML can help identify patterns of class participation, such as students who frequently arrive late or leave early, allowing for targeted support where needed.

Moreover, the system can detect anomalies like frequent absences during specific periods (e.g., before exams), which could indicate academic stress or other underlying issues. The models can also assist in resource planning by forecasting attendance trends across classes or semesters, helping institutions optimize classroom utilization and staff scheduling. By continuously learning from attendance data, the system becomes more effective at improving overall student engagement and operational efficiency.

### **Security:**

Data security and privacy are essential, especially for a system that handles student information. The system will ensure:Enroll Courses

1. Secure password encryption and storage.
2. The system will employ role-based access control (RBAC) to restrict data access based on user roles (e.g., students, teachers, administrators). This ensures that users can only access the information necessary for their role, thereby preventing unauthorized access to sensitive data.

**9. Conclusion**

The groundwork laid in Week 1 focuses on defining the scope of the project, identifying the key features, and outlining the content structure. Moving forward, we will begin prototyping the front end in Figma, drafting the ER diagram, and further refining the dataset for implementation in the coming weeks.

# **Week 2 Documentation:**

# **1. Tables description**

### 

#### **1. Student**

* **Description**: Stores information about students, including their roll numbers, names, and email addresses.
* **Attributes**:
  + roll (Primary Key): Unique identifier for each student.
  + name: Name of the student.
  + email: Email address of the student.

#### **2. Teacher**

* **Description**: Stores information about teachers, including their unique IDs, names, and email addresses.
* **Attributes**:
  + id (Primary Key): Unique identifier for each teacher.
  + name: Name of the teacher.
  + email: Email address of the teacher.

#### **3. Admin**

* **Description**: Stores information about administrators, including their unique IDs, names, and email addresses.
* **Attributes**:
  + id (Primary Key): Unique identifier for each admin.
  + name: Name of the admin.
  + email: Email address of the admin.

#### **4. Course**

* **Description**: Stores details about the courses offered.
* **Attributes**:
  + code (Primary Key): Unique course code.
  + name: Name or title of the course.
  + credits: Number of credits for the course.
  + type: Type of course (e.g., lecture, lab, seminar).
  + venue: The venue where the course is conducted.

#### **5. Slot**

* **Description**: Stores information about the time slots available for courses.
* **Attributes**:
  + id (Primary Key): Unique identifier for each slot.
  + day: Day of the week (e.g., Monday, Tuesday).
  + start\_time: The starting time of the slot.
  + duration: Duration of the slot (in minutes).

#### **6. Course-Slot**

* **Description**: Represents the association between courses and the slots they are scheduled in.
* **Attributes**:
  + course\_code (Foreign Key to Course.code): The course assigned to this slot.
  + slot\_id (Foreign Key to Slot.id): The time slot assigned to the course.

#### **7. Course-Teacher**

* **Description**: Represents the association between courses and the teachers assigned to them.
* **Attributes**:
  + course\_code (Foreign Key to Course.code): The course being taught.
  + teacher\_id (Foreign Key to Teacher.id): The teacher assigned to the course.

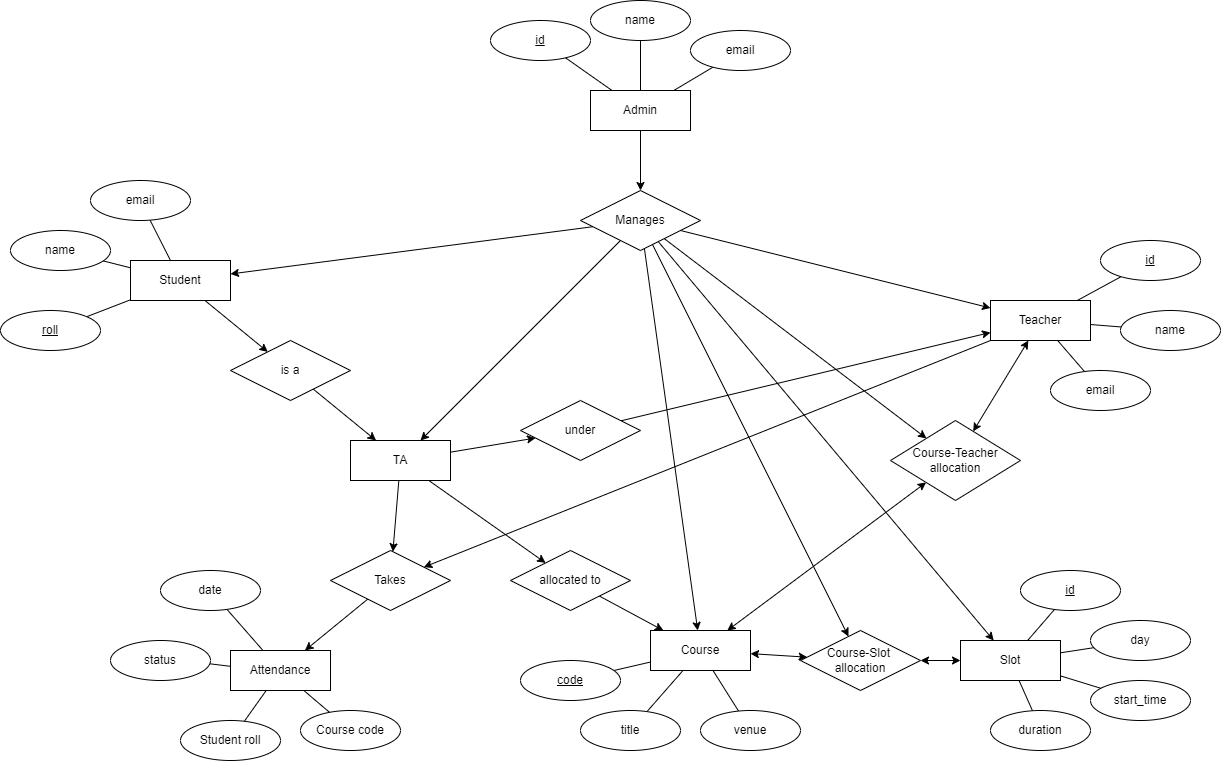
#### **8. TA (Teaching Assistant)**

* **Description**: Represents students who serve as teaching assistants (TAs) for courses. Includes the specific teacher under which the student serves as a TA.
* **Attributes**:
  + student\_roll (Foreign Key to Student.roll): The student who is the TA.
  + course\_code (Foreign Key to Course.code): The course the student is TA for.
  + teacher\_id (Foreign Key to Teacher.id): The teacher who supervises the TA.

#### **9. Attendance**

* **Description**: Stores attendance records for students in courses.
* **Attributes**:
  + student\_roll (Foreign Key to Student.roll): The student whose attendance is being recorded.
  + course\_code (Foreign Key to Course.code): The course for which the attendance is being recorded.
  + date: The date of the attendance.
  + status: Attendance status (present: 1, absent: 0).

# **2. ER Diagram**



# **3. Figma Link**

#### **1. Student**

<https://www.figma.com/proto/OOImvNIyHxVUsvRjBkCbU8/Student-Attendance-management-system?node-id=1050-1285&node-type=canvas&t=D2OkBShakkTicVnM-1&scaling=min-zoom&content-scaling=fixed&page-id=11%3A25>

#### **2. Teacher**

<https://www.figma.com/proto/OOImvNIyHxVUsvRjBkCbU8/Student-Attendance-management-system?node-id=407-4079&node-type=canvas&t=Wn7nVmR8l4114K58-1&scaling=min-zoom&content-scaling=fixed&page-id=314%3A1339&starting-point-node-id=349%3A546>

#### **3. Admin**

<https://www.figma.com/proto/OOImvNIyHxVUsvRjBkCbU8/Student-Attendance-management-system?node-id=360-3668&node-type=canvas&t=icr3ESS1fM5SyhGj-1&scaling=min-zoom&content-scaling=fixed&page-id=360%3A3667&starting-point-node-id=360%3A3914>