# 

**INTRODUCTION**

1. **TEAM DETAILS:**

The Online Attendance System has been developed by the following team members of Techno India College of Technology as MINOR PROJECT:

1. SOURAV DAS - 31401215050
2. EJAZ AHMED -

**1.Requirements**

1. **SOFTWARE REQUIREMENT SPECIFICATIONS:**

i. CLIENT REQUIREMENTS:

* The system in use should automatically update each data pertaining to a students attendance or a faculty member.
* It should require minimum technical attention to operate properly as attendance would be taken on a daily basis.
* Calculations can be done automatically.
* It should be able to hold data for a minimum of 10000 students and faculty member at a time.
* The database should be made such that minimum space is required without the need of keeping any back -up.

ii. PROJECT REQUIREMENTS:

* The project to be developed requires continuous internet connectivity with a minimum speed of \_\_\_\_\_\_.
* A device with minimum of 512MB RAM.
* Database containing information about all classes in all streams and every student.

**Iii.definitions:**

|  |  |
| --- | --- |
| WORDS | MEANINGS |
| Project | An Online attendance system. |
| Client | College faculty |
| User/actions | Faculty members and students. |
| Application |  |
| Developer | SOURAV DAS and EJAZ AHMED |
| Stakeholder |  |

**b. GOALS AND OVERVIEW:**

* **PURPOSE**: The ONLINE ATTENDANCE SYSTEM is developed to reap the benefits of internet along-with a regular task such as taking attendance as well as calculation of various statistic. It is designed for the benefit of both students and faculty in a campus.
* We offer an Online Attendance System to automate the entire attendance-related work. This system is aimed at helping both the teachers and the students. The faculty members can make a personal account on this website after which he/she can customize their classes and take attendance as well each day. The students , on the other hand , can check their status directly from the website along-with a graphical display automatically generated.
* We believe that this system would bring uniformity and reduce a lot of manual effort. There would be no need of maintaining heaps of physical data. Faculty would find it comforting to access their daily chores on-the-go as well as view the entire session at the end each of term. Students would be more aware of their attendance status in each class and therefore would be more cautious in maintaining the requirements.

**C.SCOPE:**

* This system has potential of automating office work and reducing work load thereof.
* It will take up minimum space for maintaining database .
* It will also provide a database back up of 12 years.
* It provides absolute security with the use of an log-in id and password for faculty.
* It will help students to view their attendance status and maintain transparency.

**D.CONSTRAINTS:**

**i.USER CONSTRAINTS:**

* The user should have either a smartphone or a desktop with continuous internet connectivity.
* They should also have a minimum knowledge of handling internet related activity.

**ii.SYSTEM CONSTRAINTS:**

* A server that can store 1gb of data.

**iii. TECHNICAL CONSTRAINTS:**

* A web browser that can support CSS and JAVA Script.

**E.NONFUNCTIONAL REQUIREMENT**

**i. OPERATIONAL REQUIREMENT:**

* The system can operated using the internet on any device- smartphone screen or desktop screen.
* Internet used should be be of 2G 3G 4G speed for the system to operate functionally.

**Ii.PERFORMANCE REQUIREMENTS:**

* The **response time** and **through-put** periods shall take a maximum of 30 seconds with any number of users simultaneously.

**iii. SECURITY REQUIREMENTS:**

* The system will provide each faculty members with a ID and a password to log in .
* The data of one member will not be displayed with the data of another member.
* It shall restrict access to functions to authorized group and class of users.
* It shall provide a mechanism to distinguish between the inquiry and update capabilities of each of the user class:

|  |  |  |
| --- | --- | --- |
| USER CLASS | INQUIRY ACCESS | UPDATE ACCESS TO DATA OWNED BY USER. |
| SUPER ADMIN | ALL DATA | All data except which is private to a faculty. |
| Faculty MEMBER | All private data of the account holder. | All account related data. |
| Student | Data pertaining to the average attendance. | NONE |

**iv. USER TRAINING REQUIREMENTS:**

* Basic knowledge of working with internet on various devices.

**V. THIRD PARTY SOFTWARE REQUIREMENTS:**

**F.APPLICATION INTERFACE:**

**i. user interface:** Graphical user interface.

It has following features:

**ii.Application Programming Interface:**

**T**he O-A-S API uses css and java script.

**G.Functional Requirements**

* + 1. Required Features:
* Automatic upload of data using internet.
* Member log-in facility.
* Maximum security to prevent discrepancy in data.
* All data should be stored online without a need for back up.
* Automatic calculations of statistics.
* Auto-refresh facility after completion of each semester.
  + 1. Optional Features:
* Student access to view individual data.
* Graphical display of student data.

**3. Software Project Management Plan**

A.Deliveries:

Schedule Summary

C.Start Up Plan

* + 1. Team Organization
    2. Project Communications
    3. Environment Finalization
    4. SDLC Model Finalization
  1. Work Plan
     1. Plan by Feature
     2. Flow Chart
     3. Iteration Plan
     4. Control Plan
     5. Risk Management Plan
     6. Testing Plan
  2. Design
     1. Database Design
        1. Dates
        2. Assumed Constrains
        3. Tools Required and Environment
        4. **E**ntity **R**elationship **D**iagram
        5. **D**ata **F**low **D**iagram
        6. Table Design
        7. View Design
        8. Database Constraints (primary keys, foreign keys)
        9. Cardinality of referential constraints
        10. Cascading Policy for referential constraints
     2. Interface Design
        1. User Interface Design
           1. Features Design
           2. Functional Design
           3. Page Design
     3. Back End / Server Design
        1. **O**bject **R**elational **M**apper
        2. UML Design
        3. Class Design
        4. Data Flow Diagram
        5. UI Database Mapping
  3. Technical
     1. Database Implementation
        1. Tools Used
        2. SQL’s
        3. Database Overview
     2. User Interface Implementation
        1. Tools Used
        2. Code
        3. Third Party Dependencies
     3. Server Implementation
        1. Tools Used
        2. Code
        3. Third Party Dependencies
  4. Testing
     1. Ad-Hoc Testing
        1. Test Cases and Result
     2. White Box Testing
        1. Test Cases and Result
     3. Black Box Testing
        1. Test Cases and Result

1. End User