

# Project Report: Online Attendance Management System

Name: Rahul Prasad Vagu

Internship ID: TS-RISE-DBSM-2579

Date: 15-06-2025

## 1. Project Overview

This project is a web-based Attendance Management System built using Flask and MySQL. It allows marking, viewing, and summarizing student attendance across multiple subjects and dates. It features a clean UI, summary statistics, and scalable database design.

## 2. Technologies Used

- Frontend: HTML, CSS, Bootstrap
- Backend: Python (Flask)
- Database: MySQL
- Tools: VS Code, MySQL Workbench

## 3. Features Implemented

- Mark attendance for all students by subject and date
- View attendance records in tabular format
- Generate summary statistics (Present/Absent/Total)
- Stylish homepage with navigation cards
- Back buttons and smooth navigation

## 4. Application Screenshots

## Homepage

The screenshot shows a web browser window titled "Attendance Management System" with the URL "127.0.0.1:5000". The page has a light blue background. At the top, there is a blue header bar with the title "Attendance Management System" and a subtitle "Manage and track student attendance easily". Below the header are three main buttons in white boxes:

- Mark Attendance**: Shows a plus icon and the text "Mark daily attendance for all students by subject and date." A green button labeled "Mark Attendance" is at the bottom.
- View Attendance**: Shows a document icon and the text "View attendance records for a given date and subject." A blue button labeled "View Attendance" is at the bottom.
- Attendance Summary**: Shows a bar chart icon and the text "See summary of total, present, and absent students." A yellow button labeled "View Summary" is at the bottom.

At the bottom of the page, there is a dark blue footer bar with the copyright text "© 2025 Rahul Prasad Vagu | Internship ID: TS-RISE-DBSM-2579" and a small address "127.0.0.1:5000/view-attendance".

## Mark Attendance Form

Mark Attendance

Date: 14-06-2025

Subject: English

Students:

<input checked="" type="checkbox"/> 1 - Student 1	<input checked="" type="checkbox"/> 2 - Student 2	<input checked="" type="checkbox"/> 3 - Student 3
<input type="checkbox"/> 4 - Student 4	<input checked="" type="checkbox"/> 5 - Student 5	<input checked="" type="checkbox"/> 6 - Student 6
<input checked="" type="checkbox"/> 7 - Student 7	<input type="checkbox"/> 8 - Student 8	<input type="checkbox"/> 9 - Student 9
<input checked="" type="checkbox"/> 10 - Student 10	<input checked="" type="checkbox"/> 11 - Student 11	<input checked="" type="checkbox"/> 12 - Student 12
<input type="checkbox"/> 13 - Student 13	<input checked="" type="checkbox"/> 14 - Student 14	<input type="checkbox"/> 15 - Student 15
<input checked="" type="checkbox"/> 16 - Student 16	<input type="checkbox"/> 17 - Student 17	<input checked="" type="checkbox"/> 18 - Student 18
<input type="checkbox"/> 19 - Student 19	<input type="checkbox"/> 20 - Student 20	<input checked="" type="checkbox"/> 21 - Student 21
<input checked="" type="checkbox"/> 22 - Student 22	<input checked="" type="checkbox"/> 23 - Student 23	<input type="checkbox"/> 24 - Student 24
<input type="checkbox"/> 25 - Student 25	<input type="checkbox"/> 26 - Student 26	<input checked="" type="checkbox"/> 27 - Student 27
<input checked="" type="checkbox"/> 28 - Student 28	<input type="checkbox"/> 29 - Student 29	<input type="checkbox"/> 30 - Student 30
<input checked="" type="checkbox"/> 31 - Student 31	<input checked="" type="checkbox"/> 32 - Student 32	<input checked="" type="checkbox"/> 33 - Student 33
<input type="checkbox"/> 34 - Student 34	<input checked="" type="checkbox"/> 35 - Student 35	<input type="checkbox"/> 36 - Student 36
<input type="checkbox"/> 37 - Student 37	<input type="checkbox"/> 38 - Student 38	<input type="checkbox"/> 39 - Student 39
<input checked="" type="checkbox"/> 40 - Student 40	<input checked="" type="checkbox"/> 41 - Student 41	<input checked="" type="checkbox"/> 42 - Student 42
<input checked="" type="checkbox"/> 43 - Student 43	<input type="checkbox"/> 44 - Student 44	<input type="checkbox"/> 45 - Student 45
<input checked="" type="checkbox"/> 46 - Student 46	<input type="checkbox"/> 47 - Student 47	<input checked="" type="checkbox"/> 48 - Student 48
<input type="checkbox"/> 49 - Student 49	<input type="checkbox"/> 50 - Student 50	

Submit Attendance

## View Attendance Table

Attendance on 2025-06-14 - English

Student ID	Name	Status
1	Student 1	Absent
1	Student 1	Present
2	Student 2	Absent
2	Student 2	Present
3	Student 3	Absent
3	Student 3	Present
4	Student 4	Present
4	Student 4	Absent
5	Student 5	Absent
5	Student 5	Present
6	Student 6	Present
6	Student 6	Absent
7	Student 7	Present
7	Student 7	Absent
8	Student 8	Absent
8	Student 8	Absent
9	Student 9	Absent
9	Student 9	Absent
10	Student 10	Absent
10	Student 10	Present
11	Student 11	Absent
11	Student 11	Present
12	Student 12	Absent
12	Student 12	Present
13	Student 13	Absent
13	Student 13	Absent

## Summary Result

The screenshot shows a web browser window with the title "Summary Result". The URL in the address bar is "127.0.0.1:5000/attendance-summary". The main content area displays a summary for Mathematics on 2025-06-14. It includes the following data:

Total Students	Present	Absent
50	11	39

Below the table is a button labeled "Back to Summary Form".

**Summary for Mathematics on 2025-06-14**

Total Students: <b>50</b>
Present: <b>11</b>
Absent: <b>39</b>

[Back to Summary Form](#)

## Code and Flask Logs in VS Code

The screenshot shows a code editor interface with a sidebar containing icons for file operations, search, and navigation. The main area displays a Python script named `app.py` which sets up a Flask application to handle attendance-related routes. The script includes functions for displaying the home page, marking attendance, and viewing attendance summaries. Below the code editor is a terminal window showing a log of requests from June 14, 2025, at 14:24:30 to 14:29:14, indicating interactions with the application's endpoints.

```
from flask import Flask, render_template, request
from db_config import get_connection

app = Flask(__name__)

# Home route
@app.route('/')
def index():
    return render_template('index.html')

# Mark Attendance route
@app.route('/mark-attendance', methods=['GET', 'POST'])
def mark_attendance():
    conn = get_connection()
    cursor = conn.cursor()

    if request.method == 'POST':
        subject_id = request.form['subject']
        attendance_date = request.form['date']
        present_ids = request.form.getlist('present')

        cursor.execute("SELECT student_id FROM Students")
        student_ids = cursor.fetchall()

        for (sid,) in student_ids:
            POST /mark-attendance HTTP/1.1 200 -
    127.0.0.1 - [14/Jun/2025 14:24:32] "GET /mark-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:24:40] "GET /view-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:24:53] "POST /view-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:24:56] "GET /view-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:25:01] "GET /mark-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:26:41] "GET /mark-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:27:37] "POST /mark-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:27:39] "GET /mark-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:27:42] "GET /view-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:27:52] "POST /view-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:28:28] "GET /view-attendance HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:28:23] "GET /attendance-summary HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:28:31] "POST /attendance-summary HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:28:51] "GET /attendance-summary HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:29:02] "POST /attendance-summary HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:29:06] "GET /attendance-summary HTTP/1.1" 200 -
    127.0.0.1 - [14/Jun/2025 14:29:14] "POST /attendance-summary HTTP/1.1" 200 -
```

## 5. Developer Remarks

This project demonstrates a complete CRUD-based web app with clean design and MySQL integration. The use of Bootstrap ensures a responsive UI, while Flask keeps the backend logic simple and effective.