

1. Project Title

SMART ATTENDANCE PORTAL

2. Duration

Estimated Time: 2 hrs 40 min

(Depending on familiarity with Streamlit, Python, and databases)

3. Introduction

This project is a digital attendance management system designed to replace manual roll calls. Using Streamlit for the UI and Supabase as the backend, the system allows students to mark attendance securely while enabling administrators to manage classes, attendance codes, and analytics.

It ensures transparency, reduces errors, and prevents attendance fraud using roll-number locking and daily limits.

4. Aim

To develop a simple, secure, and efficient web-based attendance system that allows real-time tracking and prevents duplicate or fraudulent attendance submissions.

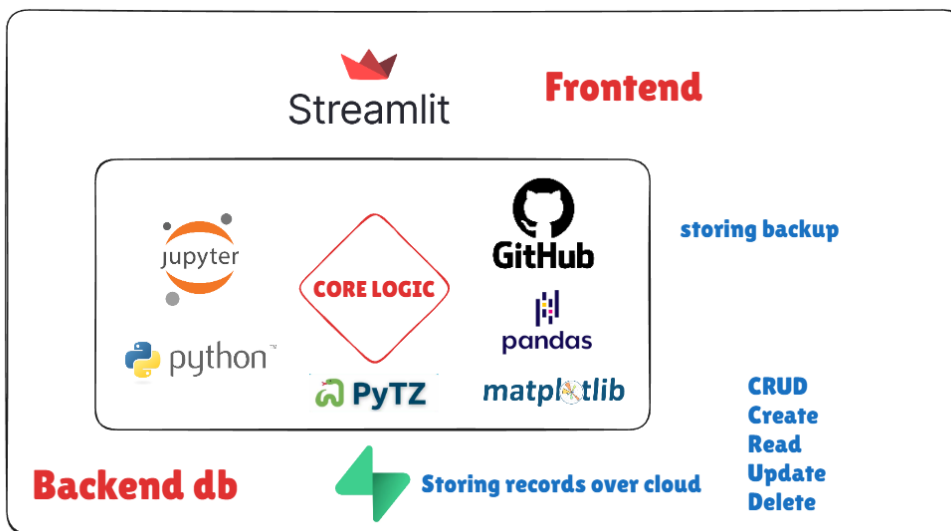
5. Dataset Involved

Dataset Name	Format	Description
attendance	Table (Supabase/Postgres)	Stores daily attendance of students with class, roll number, name, and date.
roll_map	Table	Maps roll numbers to names to prevent impersonation.
classroom_settings	Table	Stores class names, daily attendance codes, and attendance limits.
attendance matrices	CSV (auto-generated)	Exported daily/weekly attendance summaries.

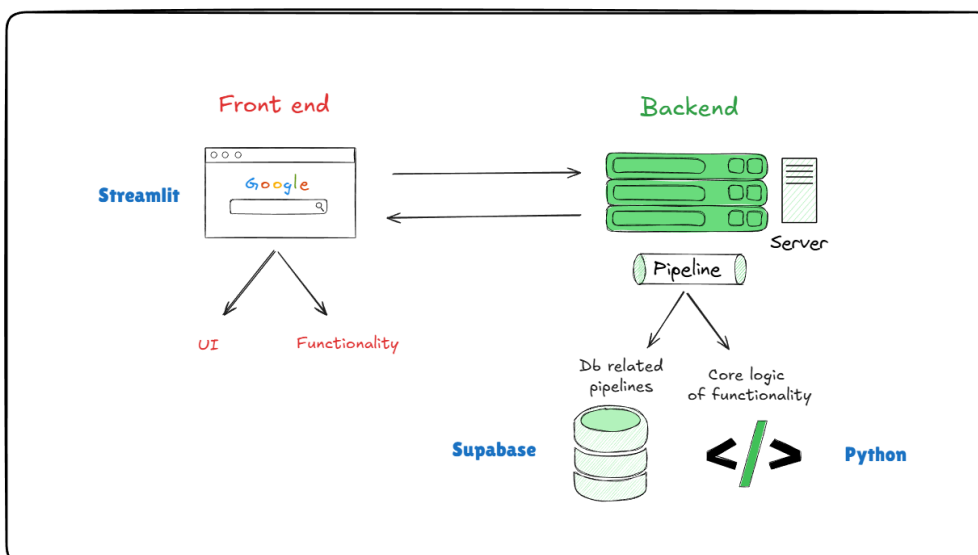
(These datasets are stored directly in Supabase tables, not external CSVs.)

6. Tools & Technologies Used

Category	Tools
Programming Language	Python
Frontend / UI	Streamlit
Backend / Database	Supabase (PostgreSQL)
Utilities	Logging, Custom Modules, Timezone Handling
Data Export	CSV files
Deployment	Streamlit Cloud



7. Architecture Diagram



8. Key Steps / Modules

1. Supabase Client Initialization

Connects Python application to the backend database using API keys.

2. Admin Module

Allows the admin to:

- Open/close classroom attendance
- Set attendance codes and limits
- View analytics
- Download attendance matrices

3. Student Module

Allows students to:

- Select class
- Enter roll number and name
- Enter attendance code
- Submit attendance if validation succeeds

4. Roll Number Locking System

Once a roll number is used, the name is stored and cannot be changed.

5. Attendance Duplication Check

Prevents multiple entries for the same student on the same day.

6. Attendance Limit Feature

Blocks submissions once the maximum daily limit for a class is reached.

7. Analytics & CSV Export

Generates attendance matrices for a given range of dates.

8. Centralized Logging System

Every action (errors, warnings, submissions) is logged to file and console with filename & line numbers.

9. Learning Outcomes

By completing this project, students will learn:

- How to build a modular Python application with multiple files.
- How to use **Streamlit** to create interactive dashboards.
- How to connect Python to a **Supabase database**.
- CRUD operations: Insert, Read, Filter, Count.
- How to implement real-world logic like:
 - Input validation
 - Duplicate checks
 - Secure code verification
 - Roll-number/name locking
- How to design a clean folder structure for long-term maintenance.
- How to generate CSV analytics from database records.
- How to build and use a **custom logging system** for debugging.
- Understanding of client-server interaction in web apps.

10. Optional Add-ons (Future Improvements)

- OTP-based student verification
- Auto-generation of daily attendance codes
- Teacher login system
- Detailed attendance analytics graphs
- Face recognition integration
- Mobile-friendly UI
- Schedule-based auto open/close classroom attendance
- Email alerts for low attendance

12. Resource Links (Optional)

Topic	Link
Streamlit Documentation	https://docs.streamlit.io
Supabase Documentation	https://supabase.com/docs
Python Logging Basics	https://docs.python.org/3/library/logging.html
Pandas CSV Manipulation	https://pandas.pydata.org/docs/

