**Updated Project Plan (SQLite + Streamlit)**

**1. User Authentication**

* Local authentication using **SQLite**
* Table: users(email, password, role)
* Roles: Admin, Teacher, Student

**2. Local Database (SQLite)**

* users → stores email, password (hashed), and role
* marks → stores student marks
  + Fields: id, student\_email, subject, marks, submitted\_by
* notices → stores notices for students/teachers
  + Fields: id, title, message, created\_by, timestamp

**3. Dashboards**

**Admin Dashboard**

* View all users and marks
* Add / Remove users (Admin can create Teacher/Student accounts)
* Post notices

**Teacher Dashboard**

* Submit marks for students
* View only their submitted marks
* Add notices for students

**Student Dashboard**

* View **real-time marks** (polled every 10s)
* Receive notices

**4. WhatsApp Notifications (Optional for Local)**

* Using **Twilio WhatsApp API**
* Triggered when:
  1. Teacher submits marks → Notifies the student/parent
  2. Admin/Teacher posts a notice → Notifies relevant users

*(If offline use only, this step can be skipped or replaced with on-screen notifications.)*

**5. Streamlit UI**

* **Login Page** → checks SQLite DB for credentials
* **Role-based Dashboard**:
  + Admin → Full control
  + Teacher → Submit marks + notices
  + Student → View marks + notices

**6. Project Folder Structure**

bash

CopyEdit

school\_management/

│

├─ app.py # Main Streamlit entry point

├─ db.py # SQLite DB functions

├─ auth.py # Authentication and session management

├─ utils.py # Helper functions (WhatsApp, notifications)

├─ requirements.txt # Dependencies

│

├─ data/

│ └─ school.db # SQLite database file

│

├─ pages/

│ ├─ Admin\_Dashboard.py

│ ├─ Teacher\_Dashboard.py

│ └─ Student\_Dashboard.py

│

└─ assets/

└─ logo.png # Optional branding

**Next Improvements**

**1. Admin Dashboard (Enhance UI)**

* Display **Users, Marks, Notices** in **tables** using st.dataframe
* Add **buttons** to:
  + **Add new user** (Teacher/Student)
  + **Post a new notice**
  + **Delete a user** (optional)

**2. Teacher Dashboard**

* Show **only teacher’s submitted marks**
* Display marks in a **nice table**
* Trigger **WhatsApp notification** when marks are submitted

**3. Student Dashboard**

* Display **marks in a formatted table**
* Display **notices in a card/list view**
* Optional: **Auto-refresh every 10 seconds** for real-time effect

**4. WhatsApp Notifications**

* Use your existing gupshup\_sender.py to:
  + Notify student/parent when **marks are submitted**
  + Notify all students when **a notice is posted**

**5. Optional Enhancements**

* **Admin Panel:** Reset student password or delete marks
* **Export Marks:** Download as CSV
* **Streamlit UI polish:** Add sidebar navigation and logos

 **Auto-refresh for Student Dashboard**

python

CopyEdit

st.experimental\_set\_query\_params(refresh="true")

st\_autorefresh(interval=10000) # every 10 sec

 **Export Marks as CSV / Excel** for Admin

 **Notice Expiry Dates** or categories (like Holiday, Exam, General)

 **Parent Numbers** in WhatsApp map for parallel notifications

 **Deploy on Streamlit Cloud** or a local server for your school

---==========================================================================

**Implications for Dashboards**

1. **Admin Dashboard**
   * Adds students using **Student ID** (mandatory)
   * Email optional for students
   * CSV/Excel export includes Student ID
2. **Teacher Dashboard**
   * Submits marks using **Student ID**
   * Auto-clear form remains
   * WhatsApp notifications will map via student\_id → phone
3. **Student Dashboard**
   * Students can log in via **Student ID** (or email if available)
   * Shows marks & notices

**What you get with this patch**

1. **Login flow**
   * Admin/Teacher → Email
   * Student → Student ID
2. **Teacher Dashboard**
   * Dropdown: Class → Section → Student → Subject
   * Submits marks using **student\_id**
   * Auto-clear after submit
3. **Student Dashboard**
   * Shows marks for student\_id
4. **Admin Dashboard**
   * Can remain same, but when adding a student:
     + student\_id required
     + student\_name, class, section required
     + email optional

**Features in this patch:**

1. **Student creation**
   * student\_id, student\_name, class, section required
   * email optional
   * password hashed
2. **Data export**
   * Users → CSV/Excel
   * Marks → CSV/Excel
3. **Notices**
   * Expiry date support
   * Active & expired notices view

**Suggested Real‑World Enhancements**

To make this system more production‑ready and marketable:

**a. Attendance Management**

* Teacher marks daily attendance (Present/Absent/Late).
* Generate monthly attendance reports per student.
* Parent auto‑notifications for absences.

**b. Timetable & Schedules**

* Class-wise timetable upload (CSV/Excel).
* Teacher can view assigned classes and subjects.
* Students can see timetable in their dashboard.

**c. Fees Management**

* Add fee structure per class.
* Record payments and generate receipts.
* Optional UPI/QR-based fee payment integration.

**d. Report Cards & Analytics**

* Auto‑generate report cards (PDF) with grades & remarks.
* Overall performance chart for each student.
* Comparison with class average.

**e. Parent Communication**

* In-app messaging between teachers and parents.
* WhatsApp/email broadcast for events, holidays, reminders.

**f. Multi‑School / Branch Support**

* Add school\_id column to support multiple branches.
* Super‑Admin can monitor all schools in one dashboard.

**g. Cloud & Backup**

* Auto DB backup to Google Drive/OneDrive.
* Optional hosting on AWS/Heroku for remote access.

**1. Transport Management**

**DB Changes**

* Create table transport\_routes  
  (id, route\_name, driver\_name, driver\_contact, vehicle\_number, stops JSON, timing JSON)
* Create table student\_transport  
  (student\_id, route\_id, pickup\_point, drop\_point)

**UI Changes**

* New menu in admin\_dashboard.py: **🚍 Manage Transport**
* CRUD operations for routes
* Assign students to routes
* View list by class/section

**WhatsApp Automation**

* Auto-send route change updates to parents
* Daily pickup/delay alerts

**2. User Management**

**DB Changes**

* Expand users table with  
  status (active/inactive), role enhancements (Librarian, Nurse, Driver)
* Audit log table: user\_audit\_logs (id, user\_id, action, timestamp)

**UI Changes**

* New menu: **👥 Manage Users**
* Add/Edit/Delete user records
* Filter by role, class, status

**WhatsApp Automation**

* New user creation → send welcome message
* Password reset link via WhatsApp

**3. Communication**

**DB Changes**

* Table messages  
  (id, sender\_id, recipient\_group, message, attachment\_path, timestamp)

**UI Changes**

* Menu: **💬 Communication Center**
* Broadcast messages to class, section, or role groups
* File/image attachments

**WhatsApp Automation**

* All communication automatically pushed to WhatsApp

**4. Library Management**

**DB Changes**

* library\_books (id, title, author, isbn, status, location)
* library\_issues (id, book\_id, student\_id, issue\_date, due\_date, return\_date)

**UI Changes**

* Menu: **📚 Library Management**
* Add books
* Issue/Return tracking

**WhatsApp Automation**

* Send due date reminders to students
* Overdue notice alerts

**5. School Configuration**

**DB Changes**

* school\_settings table for academic year, grading schema, holidays

**UI Changes**

* Menu: **⚙️ School Settings**
* Edit academic session dates
* Upload school logo, name

**WhatsApp Automation**

* Holiday alerts broadcast

**6. Front Office**

**DB Changes**

* visitor\_log (id, name, purpose, in\_time, out\_time)
* enquiries (id, name, contact, query, follow\_up\_date)

**UI Changes**

* Menu: **🏢 Front Office**
* Track visitors
* Record admissions enquiries

**WhatsApp Automation**

* Follow-up reminders to staff

**7. Website Management**

**DB Changes**

* website\_pages (id, page\_name, content\_html, last\_updated)

**UI Changes**

* Menu: **🌐 Website CMS**
* Edit About Us, News, Events pages

**WhatsApp Automation**

* Share event/news links to parents

**8. Homework/Assignment**

**DB Changes**

* assignments (id, title, subject, class, section, file\_path, due\_date)

**UI Changes**

* Menu: **📘 Homework Management**
* Upload assignments with due date
* Students can view/download

**WhatsApp Automation**

* Assignment notification with download link

**9. Student Health Management**

**DB Changes**

* student\_health (id, student\_id, checkup\_date, height, weight, notes, doctor\_name)

**UI Changes**

* Menu: **🏥 Health Records**
* Add/Update medical records

**WhatsApp Automation**

* Alert parents about upcoming health checkups

**10. Admission Management**

**DB Changes**

* admissions (id, name, dob, contact, applied\_class, status, documents\_path)

**UI Changes**

* Menu: **📝 Admissions**
* Track applicants
* Approve/reject

**WhatsApp Automation**

* Application status updates

**11. Examination Management**

**DB Changes**

* exam\_schedule (id, class, section, subject, date, time, max\_marks)
* exam\_results (id, student\_id, subject, marks, grade)

**UI Changes**

* Menu: **🧾 Exams**
* Set exam timetable
* Enter marks

**WhatsApp Automation**

* Send timetable & result summaries

**12. Principal Dashboard**

**DB Changes**

* No new DB — reuse all tables

**UI Changes**

* Aggregate view of attendance, fee collection, exam results

**WhatsApp Automation**

* Alerts for low attendance/high dues

**13. Staff Management**

**DB Changes**

* staff (id, name, role, contact, department)
* staff\_attendance (id, staff\_id, date, status)

**UI Changes**

* Menu: **👨‍🏫 Staff Management**
* Track attendance & info

**WhatsApp Automation**

* Daily attendance summaries to principal