**Safety Reporting System**

1. User Registration and Login (accounts) - Residents can create an account, log in, and securely submit safety reports.
2. Admin Panel for Authorities (admin\_panel) - Allows local authorities to view and manage submitted reports, categorize incidents, and update report statuses.
3. Incident Reporting (incidents) - Allows residents to report various safety incidents such as crimes, traffic accidents, or unsafe conditions.
4. Geolocation of Incidents (incidents) - Provides geolocation features to pinpoint the exact location of reported incidents using a map interface.
5. Status Tracking (incidents) - Residents can track the status of their submitted reports, including updates on actions taken by authorities.
6. Safety Dashboard and Analytics (admin\_panel) - Visualizes safety data and trends, helping authorities and residents understand the common concerns in the area.
7. Notifications and Alerts (notifications)
8. **Audit trail – Logs**

next lets implement Audit trail.. but we need to use mysqls workbench builtin functions instead of relying on djangos builtin function.. i have the app named audit\_trail for this..

1. **User's feedback – To improve the system or fix bugs**
2. **System Help - User’s manual**
3. **About -System info**
4. **Control Hub**:
   * **Access to Audit Logs**: Can view and manage all audit logs.
   * **Manage All Accounts**: Has the authority to create, update, and delete both admin and user accounts.
   * **Highest-Level Permissions**: Can override admin decisions and has comprehensive access to the system.
5. **Admin Panel**:
   * **Manage Basic Users**: Responsible for creating and managing user profiles.
   * **Limited Access**: Does not have access to audit logs or the ability to manage other admin accounts.
6. **Basic User**:
   * **Standard Access**: Can use the system for its intended purpose, such as reporting and viewing their own reports.
   * **No Administrative Privileges**: Cannot access admin or control hub functionalities.