Project Name: Smart Campus Issue & Resource Management System

1. Overall Approach

To address the challenges of decentralized reporting, a **Centralized Web-Based Architecture** has been adopted. This solution replaces informal communication with a structured, digital workflow.

The core strategy is **Input Standardization**. By requiring structured data entry (Category, Location, Priority) at the source, the system removes ambiguity. The application features distinct interfaces:

- **Student Portal:** Focuses on quick reporting and history tracking.
- **Admin Dashboard:** Focuses on high-level analytics and resolution management.

2. System Architecture Overview

The system follows a modular **3-Tier Architecture**:

- **Frontend (Presentation Layer):**
  - **Student View:** Provides a responsive form to capture Location, Category, Priority, and Description. It includes a "My Reported Issues" panel for status tracking.
  - **Admin View:** Features a statistical overview (Total Issues, Resolved Counts) and a master table for managing ticket status.
- **Backend Logic (Business Layer):**
  - Acts as the central controller to validate incoming data.
  - Enforces the "Smart Duplicate Detection" logic to prevent redundant entries.
  - Manages the status workflow: Reported $\rightarrow$ In Progress $\rightarrow$ Resolved.
- **Data Persistence Layer:**
  - Stores structured records of users, issues, and locations to ensure data integrity and retrieval for reporting.

3. Key Components

The system is composed of three primary logical engines:

- **Issue Management Engine:** governs the lifecycle of a complaint. It assigns a unique ID to every submission and enforces status transitions.
- **Smart Duplicate Detection System:**
  - **Logic:** Before saving a new issue, the system checks for *active* issues matching the same **Location** and **Category**.

    - **Action:** If a match is found, the submission is blocked, and the user is alerted to track the existing ticket. This prevents database clutter.
- **Reporting Service:** Aggregates data to display key metrics (e.g., "High Priority Issues") on the Admin Dashboard via visual cards and color-coded tables.

4. Data Flow Explanation

1. **Submission:** A user logs in, selects a Location/Category, sets Priority, and submits.
2. **Validation:** The system runs the duplicate check. If unique, the data is saved with a timestamp.
3. **Visualization:** The Admin Dashboard updates immediately, highlighting "High Priority" items in red for urgent attention.
4. **Resolution:** An administrator updates the status to "Resolved," which reflects on the Student's history view.

5. Design Decisions

- **Web Application:** Selected to ensure accessibility across laptops and mobile devices without installation.
- **Priority Flags:** Mandatory "High/Medium/Low" fields were implemented to ensuring safety hazards are not ignored.
- **Color-Coded UX:** The Admin interface uses visual cues (Red/Green badges) to allow rapid scanning of critical issues.