## **🗓️ Phase 1: Core Functionality — Database, Reporting, and User Flow**

Objective: Build the foundation of the Malshinon system — including the database schema, user input flow, intel submission, and basic metrics tracking.

### **🗂️ Database Design**

#### **✅ 1. Create People Table (suggestion - build diff tables if you want!)**

This table stores **all individuals** (reporters, targets, or both).

**Required columns:**

* id (Primary Key, auto-increment)
* first\_name (VARCHAR)
* last\_name (VARCHAR)
* secret\_code (VARCHAR, UNIQUE)
* type (ENUM or VARCHAR) — values: reporter, target, both, potential\_agent
* num\_reports (INT, default 0) — used if person is a reporter
* num\_mentions (INT, default 0) — used if person is a target

#### **✅ 2. Create IntelReports Table**

This table stores **each piece of submitted intel**.

**Required columns:**

* id (Primary Key, auto-increment)
* reporter\_id (Foreign Key → People.id)
* target\_id (Foreign Key → People.id)
* text (TEXT)
* timestamp (DATETIME, default NOW())

#### **✅ 3. Export to .sql Script**

* Create a single .sql file that:  
  + Creates both tables
  + Sets appropriate constraints and defaults
  + Ensures all IDs are auto-incremented
  + Uses UTF-8 encoding

### **💻 C# Application Tasks**

#### **✅ 1. Git & GitHub Setup**

* Create a **new GitHub repository**
* Work in a **dedicated feature branch** (not main/master)
* Use frequent commits with meaningful messages
* After completing each feature, **open a pull request (PR)** for review and merging

#### **✅ 2. Person Identification Flow**

* Prompt the user to enter their full name
* Search for a matching person in the database
* If no match is found:  
  + Generate a **unique secret code**
  + Insert a new record into the People table
  + Set type to reporter (by default)

#### **✅ 3. Intel Submission Flow**

* Prompt user for a **free-text report**
* Scan the text for a **target name**:  
  + Assume names are always in **Capitalized First and Last Name** format
  + Extract the name and look it up in the People table
  + If not found → create a new person of type target
* Insert the intel into the IntelReports table:  
  + Link to reporter and target by ID
  + Store the full text
  + Use the current timestamp

#### **✅ 4. Metrics & Status Updates**

* After inserting the intel:  
  + Increment the reporter's num\_reports
  + Increment the target's num\_mentions
* Check thresholds:  
  + If **reporter has ≥10 reports** and **average text length ≥100 characters**, update their type to potential\_agent
  + If **target has ≥20 mentions**, log a **potential threat alert**

### **🧪 Additional Engineering Requirements**

#### **✅ Logging**

* Log major actions such as:  
  + New person creation
  + Report submission
  + Status changes (e.g., flagged as potential agent or threat)