# Section 5. Terms of Reference

## 1. Objective and Scope of Work

The consulting services aim to design, develop, test, and deploy a robust web-based **Online Complaint Registration and Management System** for **\_\_\_\_\_\_\_\_ Municipality**. This system will allow citizens to register complaints online while enabling the administration to manage the entire lifecycle of each complaint—from submission and pending edits to review, case conversion, hearing scheduling, and resolution.

**The consulting firm shall perform the following tasks:**

* **Process Review:**  
  Evaluate any existing complaint registration process and identify areas for improvement.
* **Feasibility Analysis:**  
  Recommend an economically viable solution for system development.
* **Database Design:**  
  Architect and design a scalable, secure database that supports complaint tracking, case management, and file storage.
* **System Development:**  
  Design and develop the web-based application using a modular MVC approach with distinct modules for Users, Admins, and Members.
* **Testing and Quality Assurance:**  
  Conduct comprehensive testing—including functional, security, and performance tests—to ensure the system meets quality standards.
* **Hosting Services:**  
  Provide and manage the hosting environment for the deployed application.

## 2. Domain Analysis

In collaboration with **\_\_\_\_\_\_\_\_ Municipality**, the consulting firm shall perform a detailed domain analysis to ensure that the system aligns with current government guidelines and user needs.

**This analysis will include:**

### 2.1 Compliance Review

* Evaluate and ensure adherence to the Government Website Development and Management Guidelines and the Right to Information Act.

### 2.2 Platform and Hosting Evaluation

* Assess and recommend the optimal development platform and hosting environment to ensure reliability and security.

### 2.3 Requirements Gathering

* Organize meetings with municipal officials to collect comprehensive system requirements, including:
  + Complaint workflow
  + Role-based access (User, Admin, and Member modules)
  + File management
  + Notification mechanisms (SMS/Email)
  + Reporting capabilities

## 3. System Development

Following domain analysis and requirements gathering, the consulting firm will mobilize a dedicated development team to build the web-based application. The system must be:

* **Scalable**
* **Secure**
* **User-friendly**

**Core features include:**

* **Complaint Module:**  
  Allows users to register, track, and edit complaints (status remains “pending” until approval).
* **Approval Workflow:**  
  Enables Admins and Members to review complaints, add files/comments, approve or reject applications, and convert approved complaints into cases.
* **Case Management:**  
  Facilitates scheduling of hearings (first, second, third dates, etc.), assignment of case handlers, and management of final resolutions with case result uploads.
* **Notification System:**  
  Integrates SMS and email notifications to keep users updated throughout the process.
* **Reporting and Dashboard:**  
  Provides visual dashboards and reporting tools (charts: bar, line, pie) for tracking complaint statuses, timelines, and ward-specific data.

## 4. Documentation

Throughout the project and upon system completion, the consulting firm shall deliver the following documentation in accordance with mandatory GEA2.0 artifacts or equivalent standards:

* Project Charter
* Project Management Plan
* User Requirements Specification (SRS/FRS)
* Software Architecture Design Document
* Software Quality Assurance Plan
* Test Plans/Test Cases
* Development Plan
* Code Review Checklist
* Test Report
* User Manual
* Administrative Plan
* Performance Test Summary Report
* Support and Maintenance Strategy
* Work Completion Report

## 5. System Orientation and Handover

Upon completion, the consulting firm shall perform the following:

### 5.1 Orientation Sessions

* Conduct comprehensive orientation sessions for officials of **\_\_\_\_\_\_\_\_ Municipality** to demonstrate system functionalities.

### 5.2 Formal Handover

* Execute a formal handover that includes:
  + Delivery of updated, fully functional source code
  + Complete documentation
  + A signed acceptance document

### 5.3 Technical Training

* Provide detailed technical training for the municipality’s staff on system administration, user management, and best practices.

### 5.4 Post-Deployment Support

* Implement bug fixes and security patches in accordance with the latest governmental IT system standardization guidelines (e.g., recommendations from the Nepal Government, Department of Information Technology).

## 6. Scope of Work

The developed system shall include the following modules and components:

### 6.1 Complaint Registration and Management Module

* **Form/Data Entry Section:**
  + User-friendly registration form to capture user details, complaint description, and supporting documents.
  + Automated generation of a unique reference number for each complaint.
  + Ability for users to edit submissions while the complaint status remains “pending.”
* **File Upload Section:**
  + Enables users to upload and manage supporting files (images, PDFs, etc.).

### 6.2 Complaint Review and Approval Module

* **Admin/Member Interface:**
  + Comprehensive view of complete complaint details, including text, files, and user comments.
  + Functions to edit complaint data, add comments, attach additional files, and update status.
  + Options to approve or reject complaints, with an integrated notification system (via SMS/Email).

### 6.3 Case Management Module

* **Conversion and Case Details:**
  + Convert approved complaints into formal cases.
  + Capture detailed case information, including:
    - Scheduling hearing dates (first, second, third, etc.)
    - Time, location, and case handler assignments.
  + Support ongoing case updates via comments, file uploads (e.g., case results, hearing documents), and status adjustments.

### 6.4 Dashboard and Reporting Module

* **Visual Dashboard:**
  + Display key performance indicators (KPIs) and real-time statistics using dynamic charts (bar, line, pie) that reflect complaint statuses, trends, and ward-specific data.
  + Provide filtering options for date ranges, complaint types, and geographical areas.
* **Reporting Features:**
  + Generate detailed and summary reports exportable in various formats (e.g., Excel, PDF) for analysis and review.

### 6.5 System Administration Module

* **User Management:**
  + Enable creation, modification, and deletion of user accounts for all system roles (Public Users, Admins, Members).
  + Implement robust role-based access control.
* **Configuration and Security:**
  + Offer settings to manage notifications, data backup, and security protocols.
  + Ensure secure handling of sensitive data with encryption and regular validation.

## 7. Requirements

### 7.1 Proposed Technology

The system shall be developed using the following technology stack:

* **Presentation / Front-end Tier:**
  + HTML, CSS, JavaScript, and Bootstrap for a responsive, user-friendly interface.
  + Compatibility with all major browsers.
* **Application Tier:**
  + PHP implementing an MVC architecture for separation of concerns and maintainability.
  + Supports login/logout functionality for Admin and Member users.
* **Persistence / Database Tier:**
  + MySQL for reliable, scalable data storage and management.
* **Deployment Environment:**
  + Deployed on either Windows IIS or Linux, with hosting provided by the Municipality.
* **Security and Session Management:**
  + Data encryption using AES256CBC to protect sensitive information at rest and in transit.
  + User passwords hashed using SHA256 with unique salts.
  + Secure management of cookies, sessions, and HTTP headers, including encryption, appropriate expiry settings, and HTTPOnly flags.

### 7.2 Core Modules

* **User Module (Public):**
  + Users can view the homepage, register complaints, track status, edit pending complaints, view uploaded files, and upload additional files.
* **Admin Module:**
  + Full system control including viewing, editing, commenting, sending notifications (SMS/Email), file management, and complaint data handling.
  + Ability to approve, reject, or assign hearing dates.
  + Administrative capabilities to add, delete, or update Admins and Members.
* **Member Module:**
  + Similar to the Admin module but with restricted control.
  + Members (Ward Members, Case Handlers, or both) can view, edit, comment, send notifications, and manage files related to cases.
  + Access restricted to complaints from their designated ward.
* **Dashboard Module:**
  + Visual summaries with absolute numbers and 2D charts (standard and bifurcated by status) such as Count vs. (Type, Time, Ward).
  + Filtering options for pending cases by date range.
  + Admins view all cases; Members view only cases relevant to their ward.

### 7.3 Security and Session Management

* **Authentication:**
  + Robust login/logout for Admin and Member users to ensure only authorized access to complaint data.
* **Encryption:**
  + Sensitive data is encrypted using AES256CBC during transmission and storage.
  + User credentials are hashed using SHA256 with unique salts.
* **Cookie/Session and Header Security:**
  + Sessions and cookies are encrypted and configured with HTTPOnly and Secure flags.
  + Strict expiry policies minimize session hijacking risks.
  + Security-related HTTP headers are implemented to safeguard against common vulnerabilities.

### 7.4 Security

* **7.4.1 User Verification and Authorization:**  
  Implement a robust user verification process along with role-based authorization to control access to system functionalities and sensitive data.
* **7.4.2 Data Validation:**  
  Enforce comprehensive data validation to maintain integrity and prevent injection of malicious code or unauthorized data.
* **7.4.3 User Password Encryption:**  
  Encrypt user passwords using secure hashing algorithms (SHA256 with unique salts) to prevent unauthorized access, ensuring credentials are not stored or transmitted in plain text.

### 7.5 Project Management

* **7.5.1 Agile Methodology:**  
  Adopt an Agile project management approach to facilitate iterative development, frequent collaboration, and adaptability to evolving requirements, thereby ensuring timely delivery and enhanced project efficiency.

### 7.6 Testing

* **7.6.1 Vulnerability Assessment and Penetration Testing (VAPT):**  
  Conduct thorough vulnerability assessments and penetration testing to identify, evaluate, and remediate potential security risks, ensuring the system meets industry-standard security benchmarks.

## 8. Database Schema and Tables

The system’s data layer is built on MySQL and follows a normalized structure to efficiently manage the complaint lifecycle. Key tables include:

### 8.1 Application Table

Captures all details related to user complaints, with key fields such as:

* **reference\_id:** A unique identifier (generated via a stored procedure) for each complaint.
* **title, subject, type:** Categorization details (e.g., Property Dispute, Land Acquisition).
* **description:** Detailed explanation of the complaint.
* **plantiff\_name, plantiff\_address, plantiff\_ward\_number, plantiff\_mobile, plantiff\_email, plantiff\_adhaar:** Complainant information.
* **plantiff\_father\_name, plantiff\_grandfather\_name:** Additional identifiers.
* **defendant\_name, defendant\_address, defendant\_ward\_number, defendant\_mobile, defendant\_email, defendant\_adhaar, defendant\_father\_name, defendant\_grandfather\_name:** Details of the opposing party.
* **file\_upload:** References to uploaded files (images, PDFs, etc.).
* **created\_at, updated\_at:** Timestamps for record creation and modification.

### 8.2 Application\_Status Table

Logs status updates and review activities for each complaint:

* **reference\_id:** Links to the Application table.
* **status:** Current status (e.g., Pending, Approved, Rejected).
* **comment:** Reviewer/editor comments.
* **file\_upload:** Additional files uploaded during review.
* **editor\_name, editor\_email, editor\_mobile:** Details of the Admin or Member who performed the update.
* **created\_at, updated\_at:** Timestamps for status entries.

### 8.3 Case\_Status and Resolved\_Status Tables

Capture evolving case details once a complaint is approved:

* **Case\_Status:** Stores scheduled hearing dates (first, second, third, etc.), case handler assignments, and interim updates.
* **Resolved\_Status:** Records final resolution details, including comments, resolution documents, and concluding file uploads.

### 8.4 Admin and Member Tables

Manage user authentication and roles:

* **Admin Table:**
  + Stores system administrator details (username, hashed password using SHA256 with unique salts, email, contact details).
  + Includes role and permission fields for full system control.
* **Member Table:**
  + Contains information on members (e.g., Ward Members, Case Handlers) with similar fields to the Admin table.
  + Enforces restrictions so members access only complaints relevant to their ward.

**Key Points Regarding Field Definitions:**

* The **Application** table is designed with dynamic fields to capture comprehensive complaint details, ensuring each complaint is uniquely identified.
* The **Application\_Status** table maintains a history of updates, supporting detailed audit trails.
* Additional tables (e.g., **Case\_Status**, **Resolved\_Status**) will similarly store hearing schedules, case updates, and final resolutions.
* The **Admin** and **Member** tables support secure login/logout and session management, in line with the security measures detailed in Section 7.3.