
**Software Requirements Specification
for
Anonymous Harassment Reporting
Platform**

Version 1.0

Prepared by Ashwatha S

04-10-2025

Table of Contents

| | |
|---|-----------|
| Table of Contents | 2 |
| Revision History | 2 |
| 1. Introduction..... | 3 |
| 1.1 Purpose..... | 3 |
| 1.2 Document Conventions..... | 3 |
| 1.3 Intended Audience and Reading Suggestions | 3 |
| 1.4 Project Scope | 4 |
| 1.5 References..... | 4 |
| 2. Overall Description..... | 5 |
| 2.1 Product Perspective..... | 5 |
| 2.2 Product Features | 9 |
| 2.3 User Classes and Characteristics | 10 |
| 2.4 Operating Environment..... | 11 |
| 2.5 Design and Implementation Constraints | 12 |
| 2.6 User Documentation | 14 |
| 2.7 Assumptions and Dependencies | 15 |
| 3. System Features | 16 |
| 3.1 Anonymous Report Submission | 16 |
| 3.2 Report Tracking | 16 |
| 3.3 Admin and Safety Officer review | 17 |
| 4. External Interface Requirements | 17 |
| 4.1 User Interfaces | 17 |
| 4.2 Hardware Interfaces | 18 |
| 4.3 Software Interfaces | 18 |
| 4.4 Communications Interfaces | 19 |
| 5. Other Nonfunctional Requirements..... | 20 |
| 5.1 Performance Requirements | 20 |
| 5.2 Safety Requirements | 20 |
| 5.3 Security Requirements | 20 |
| 5.4 Software Quality Attributes | 21 |
| 6. Other Requirements | 21 |
| Appendix B: Analysis Models | 22 |

Revision History

| Name | Date | Reason For Changes | Version |
|------|------|--------------------|---------|
| | | | |
| | | | |

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document specifies the requirements for the **Anonymous Harassment Report Platform**. The platform aims to provide a secure and confidential online system that allows individuals to report harassment incidents anonymously without requiring registration. This encourages victims and witnesses to come forward without fear of retaliation or exposure.

The SRS describes the core functionality, user interactions, and system constraints, focusing on the modules related to anonymous reporting, administration, and safety officer review and follow-up.

1.2 Document Conventions

- **Bold text** is used to emphasize key features or important terms.
- *Italic text* is used to highlight user roles or actions.
- Monospaced text is used to refer to system messages or specific code snippets where applicable.
- Requirements are categorized with identifiers like FR- (Functional Requirement) or NFR- (Non-Functional Requirement) when needed.
- Priority levels (High, Medium, Low) are included next to major requirements.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- Individuals or small groups deploying the system: To understand how to set up, configure, and maintain the anonymous harassment reporting system.
- Organizations or institutions (e.g., schools, colleges, communities): To evaluate the system for integration into their harassment reporting or feedback mechanisms.
- Moderators, safety coordinators, or designated reviewers: To understand how to receive, review, and respond to submitted reports.
- End users (victims, witnesses, or community members): To gain a general understanding of the system's purpose, data privacy, and how their reports will be handled anonymously.
- Admin: To manage the submitted reports, route them to appropriate safety officers, update their status throughout the process, and ensure reports are resolved and closed while maintaining oversight of the reporting workflow.

It is recommended that readers begin with the Introduction for a high-level understanding of the system's goals and context. Functional Requirements provide specific features and

behaviors of the system, followed by Non-Functional Requirements and Use Cases, which offer deeper insight into expected performance and usage scenarios.

1.4 Project Scope

The **Anonymous Harassment Report Platform** is a web-based system designed to enable individuals to anonymously submit harassment reports directly to authorized safety officers and administrators.

Key objectives and benefits:

- **Anonymity:** No user account or personal identification required for submitting reports.
- **Accessibility:** Available online 24/7, accessible via desktop and mobile devices.
- **Action-oriented:** Enables safety officers to follow up on reports internally without compromising reporter identity.
- **Transparency and trust:** Builds confidence in the reporting process and encourages more reports.
- **Admin and safety officer roles:** Manage, categorize, and track reports, generate analytics, and mark case status.

This software aligns with organizational strategies to create a safe and inclusive environment by empowering individuals to report misconduct without fear.

1.5 References

- IEEE 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
- OWASP Secure Coding Practices (<https://owasp.org/www-project-secure-coding-practices/>)
- Web Content Accessibility Guidelines (WCAG) 2.1 (<https://www.w3.org/TR/WCAG21/>)

Overall Description

2.1 Product Perspective

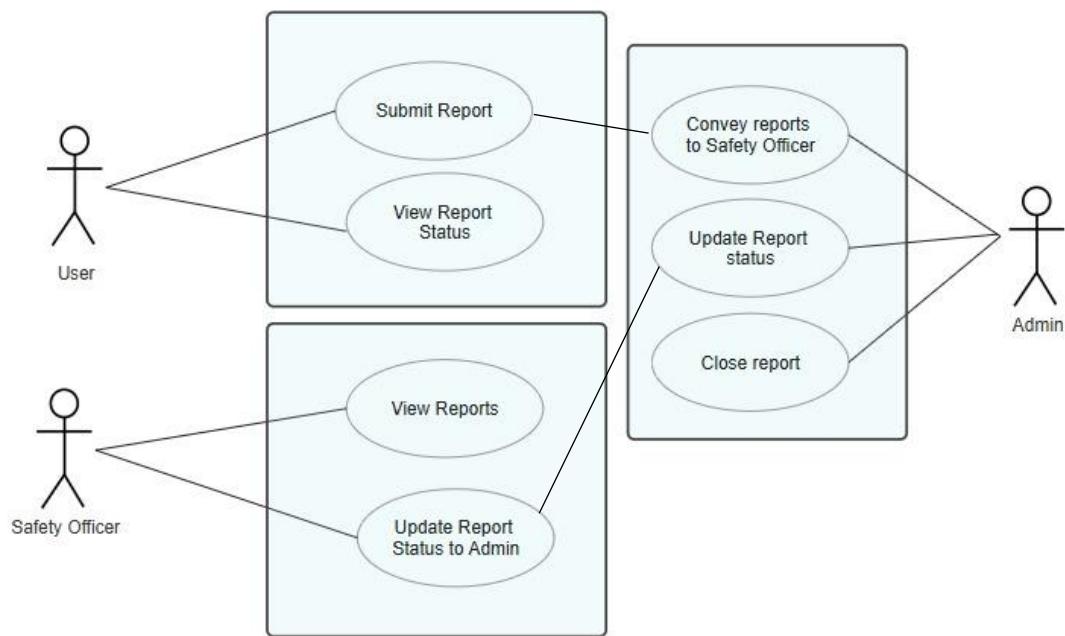
This product is a new, self-contained web-based system designed to allow users to anonymously report harassment incidents within an institution, such as a school, college, or organization. It is not a component of a larger product family but a standalone tool meant to facilitate safe, confidential communication between the affected users and designated reviewers (safety officers).

There are two primary actors and one secondary actor in the system:

- Anonymous Users (victims or witnesses),
- Administrators (system managers),
- Safety Officers (authorized reviewers)(secondary).

The diagram below outlines the Use Case Diagram, showing the main interactions between users and the system.

Figure 2.1 – Use Case Diagram of the Anonymous Harassment Reporting System



Use Case Relationships Explanation

- The "**Convey Reports to Safety Officer**" use case <<include>> the "**Update Report Status**" use case, as updating the status is a mandatory part of conveying a report.
- The "**Close Report**" use case also <<include>> the "**Update Report Status**" use case to ensure the report's final status is properly recorded.
- The "**Submit Report**" and "**View Report Status**" use cases are directly accessed by the user without any inclusion or extension relationships.

These relationships ensure clarity about the system's behavior and dependencies among functionalities.

Actors:

| Actor | Type | Description |
|----------------|-----------|--|
| User | Primary | Submits anonymous harassment reports and views report status. |
| Admin | Primary | Conveys reports to safety officers, updates report status, and closes reports. |
| Safety Officer | Secondary | Views reports and updates report status as instructed by admin. |

Use Case Descriptions:

UC1 – Submit Report (User)

Preconditions:

- User can access the platform (web or mobile).
- Reporting form is operational and available.
- No login or registration is required.

Trigger:

- User decides to report an incident and clicks the "Submit Report" button.

Main Success Scenario:

1. User opens the report submission form.
2. User fills in details (description, incident details, optional evidence).
3. User clicks "Submit".
4. System validates input.
5. System saves report and generates a unique reference ID.

6. System displays confirmation and reference ID to the user.

Exception Scenarios:

- Incomplete or invalid input → System displays validation errors.
- Server/system downtime → System shows an error message.

UC2 – View Report Status (User)

Preconditions:

- User has a valid report reference ID.
- The report exists in the system.
- The system is online and accessible.

Trigger:

- User enters a valid reference ID to check the report status.

Main Success Scenario:

1. User accesses "View Report Status" page.
2. Enters valid reference ID.
3. System retrieves the report.
4. System displays the current report status.

Exception Scenarios:

- Invalid or incorrect reference ID → "Invalid ID" message.
- Report not found or deleted → System informs the user.
- System error → System shows a message.

UC3 – Convey Reports to Safety Officer (Admin)

Preconditions:

- There is at least one pending/new report.
- Admin is logged in and authorized.
- Safety officer details are available.

Trigger:

- Admin chooses to forward a report to the safety officer.

Main Success Scenario:

1. Admin logs in.

2. Reviews pending reports.
3. Selects a report to forward.
4. Assigns safety officer and clicks "Convey".
5. System updates status and notifies officer.

Exception Scenarios:

- No pending reports → System shows "No reports to convey".
- Safety officer not available → System displays error.
- Admin session expired → Prompts re-login.

UC4 – Update Report Status (Admin or Safety Officer)

Preconditions:

- The report exists and is not closed.
- Admin or safety officer is authorized.

Trigger:

- Admin or officer chooses to update the status after review/action.

Main Success Scenario:

1. Logs in and selects report.
2. Chooses a new status (e.g., "Under Review").
3. Clicks save.
4. System updates and confirms.

Exception Scenarios:

- Report already closed → Update blocked.
- Unauthorized access → Access denied.
- Invalid status → System prompts correction.

UC5 – Close Report (Admin)

Preconditions:

- Report is in a final state (e.g., "Investigated").
- Admin is authorized.
- All details are updated.

Trigger:

- Admin decides the report is fully resolved and closes it.

Main Success Scenario:

1. Admin logs in and reviews report.
2. Confirms completion.
3. Clicks "Close Report".
4. System updates status to "Closed".
5. System shows confirmation.

Exception Scenarios:

- Report not in final state → System blocks closure.
- Unauthorized admin → Access denied.
- System/database error → Report remains open.

2. Product Features

The Anonymous Harassment Reporting Platform provides a secure, user-friendly interface for individuals to submit harassment reports anonymously while enabling administrators to manage and track reported cases. At a high level, the major product features include:

1. Anonymous Reporting

- Users can submit harassment complaints without revealing their identity.
- Option to attach details such as date, place, and supporting proof (documents, images, audio, video).

2. Harassment Details Collection

- Structured input for personal details (optional: name, age, gender).
- Dedicated field for incident description/harassment details.
- Dropdowns and date pickers to standardize inputs (e.g., date and location).

3. Unique Reference Code Generation

- Each submitted report is assigned a unique reference code.
- Users can use this code to track their case status.

4. Case Management Dashboard (for Admin/Authorities)

- Secure login for authorized personnel.
- View, filter, and manage submitted reports.

- Update report status (e.g., Under Review, Action Taken, Closed).

5. Data Security and Privacy

- End-to-end encryption of submitted reports.
- Role-based access to prevent unauthorized viewing.
- Secure storage of attachments and user-provided evidence.

6. Multi-Platform Accessibility

- Responsive web interface accessible on desktop and mobile devices.
- Cross-browser compatibility for seamless user experience.

7. User Support Features

- Help/FAQ section explaining how to submit and track reports.
- Tutorials or on-screen guidance for first-time users.

2.3 User Classes and Characteristics

This system is designed to be used by three main user classes, each with distinct roles, access privileges, and technical expectations:

1. Anonymous Users (Victims or Witnesses)

Characteristics:

- Do not require login or user accounts.
- May be emotionally distressed or fearful of retaliation, requiring a simple, intuitive, and non-intrusive interface.
- May only use the system occasionally or once.
- Require assurance of anonymity and privacy.
- Typically have low to moderate technical expertise.
- May come from diverse backgrounds (students, employees, bystanders).

Key Needs:

- Ability to submit a report easily and anonymously.
- Option to attach supporting evidence (screenshots, messages).
- (Optional) Receive a tracking code to follow up on the report.
- Access to guidelines, privacy policy, and support resources.

2. Safety Officers (Moderators or Reviewers)

Characteristics:

- Designated by the institution to review and manage incoming reports.
- Require secure, role-based access to view and act on reports.
- May use the system frequently, depending on report volume.
- Must handle sensitive information ethically and confidentially.
- Have moderate technical proficiency.
- Need to comply with institutional or legal protocols.

Key Needs:

- Dashboard to view, categorize, and respond to reports.
- Ability to mark cases as resolved or escalate if needed.
- Access to report summaries and statistics.
- Training on privacy, data handling, and user protection policies.

3. Administrators

Characteristics:

- Responsible for initial setup, maintenance, and management of the platform.
- May be part of the IT team or designated by the institution.
- Have high-level access to all system settings.
- Use the system periodically, especially during setup or updates.
- Typically possess advanced technical knowledge.

Key Needs:

- Configure system categories, workflows, and permissions.
- Manage user roles (assign/revoke Safety Officer access).
- Maintain data integrity, backups, and security.
- Monitor usage and troubleshoot issues.

2.4 Operating Environment

The software will operate within a typical institutional web-application environment designed for secure, privacy-focused report management.

Client (User) Environment

- Supported Browsers: Google Chrome (latest two major versions), Mozilla Firefox (latest two), Microsoft Edge (Chromium, latest two), and Safari (latest two).
- Device Compatibility: Responsive design supporting desktops ($\geq 1024 \times 768$), tablets, and mobile devices ($\geq 320\text{px}$ width).
- Client Requirements: JavaScript and cookies must be enabled for proper functionality.
- Accessibility: Compliant with WCAG 2.1 AA standards for accessible interaction.

Server Environment

- **Web Server:** Apache Tomcat 10+ configured to serve Java Servlets and JSP pages.
- **Application Platform:** Java EE (Jakarta EE) stack using Servlets, JSP, and JDBC for database interaction.
- **Operating System:** Linux (Ubuntu 22.04 LTS / CentOS 8) or equivalent cloud-hosted virtual machine/container environment (e.g., Docker).
- **Database:** MySQL 8.0+ for data persistence of user credentials, anonymous reports, and administrative data.
- **Storage:** Local or cloud-based persistent storage for evidence attachments (secure folder or S3-compatible object storage such as AWS S3 / MinIO).
- **TLS/HTTPS:** TLS 1.2+ (preferably TLS 1.3) enabled for all web traffic. Certificates managed via institutional CA or Let's Encrypt.

Other Software Components

- **Reverse Proxy / Load Balancer (optional):** Nginx or Apache HTTPD for handling HTTPS termination, caching, and load balancing.
- **Email Delivery:** SMTP-based mail server for sending acknowledgment and administrative notifications (optional for anonymous submissions).
- **Authentication Provider:** Local authentication using MySQL for admin login, with optional integration for LDAP or SSO (OAuth2/SAML) if required by the institution.
- **Logging and Monitoring:** Application and access logs maintained on the server with optional integration to ELK stack (Elasticsearch, Logstash, Kibana) or cloud monitoring solutions.
- **Backup and Recovery:** Automated database backups and file storage snapshots to ensure data integrity and compliance with institutional data retention policies.

2.5 Design and Implementation Constraints

1. Privacy-First Requirement

- The platform is designed for anonymous harassment reporting.
- Reporters are not required to create accounts or share personal information.

- Only minimal, voluntarily provided details are stored for analytical or contextual purposes.
- All identifying information is optional and securely stored with consent.

2. Regulatory and Institutional Compliance

- The system adheres to institutional reporting and data privacy policies.
- Design principles align with GDPR-style privacy laws, ensuring no unauthorized storage or sharing of personal information.
- Periodic audits may be performed by institutional IT or legal departments.

3. Data Retention and Audit Policy

- Anonymous reports are retained for a predefined duration (e.g., 3 years) and automatically purged unless flagged for ongoing investigation.
- Administrative actions (e.g., report assignment or status changes) are logged for accountability without linking to reporter identity.

4. Hardware and Deployment Constraints

- The application must be deployable on standard institutional infrastructure (2–4 vCPUs, 4–8 GB RAM).
- For scalability, the system can be containerized using Docker and deployed on Kubernetes with persistent storage for the database and uploaded files.

5. Interoperability

- The system exposes REST/JSON APIs for integration with campus case-management or reporting systems.
- APIs follow REST conventions and use HTTPS for secure communication.
- Example endpoints include /api/reports, /api/reports/{referenceCode}, and /api/admin/reports/{referenceCode}/assign.

6. Third-Party Services

- External services (e.g., SMTP or cloud storage) may be used if they comply with institutional privacy requirements.
- Personally Identifiable Information (PII) must not be transmitted to third-party providers.

7. Security Constraints

- All communications must occur over HTTPS.
- Sensitive data is encrypted both in transit (via TLS) and at rest (via database or object storage encryption).
- Administrative logins must use secure credentials (or institutional SSO/MFA in production).

- Database access is restricted to least-privilege accounts.
- File uploads are sanitized and stored using randomized identifiers to prevent malicious file execution.

8. Technology and Coding Standards

- Java Servlets and JSP follow the Google Java Style Guide.
- Frontend code adheres to HTML5/CSS3 standards and cross-browser compatibility.
- Unit and integration testing are required before deployment.
- Continuous integration pipelines (e.g., GitHub Actions) are recommended for automated builds and deployment.

9. Browser Compatibility

- Compatible with modern browsers (Chrome, Firefox, Edge, Safari).
- Avoids experimental browser APIs to ensure consistent accessibility and security.

10. Accessibility Compliance

- The interface follows WCAG 2.1 AA guidelines.
- Supports keyboard navigation, screen readers, and clear form labels for all inputs.

11. Localization and Extensibility

- The current version supports English.
- The design allows future internationalization through resource bundles and message files.

2.6 User Documentation

Deliverables and formats:

- **User manual (reporters)**
 - PDF and online HTML: “How to submit an anonymous report”, explanation of anonymity guarantees, what happens after submission, FAQs.
- **Administrator manual**
 - PDF and HTML: managing incoming reports, triage workflow, evidence handling, user/role management, audit logs.
- **On-line contextual help**
 - In-app tooltips and a Help page describing key features and privacy guarantees.
- **Quick start guides**

- One-page printable cheat-sheets for campus safety officers and system admins.
- **Tutorial / walk-through**
 - Interactive web walkthrough (first-time admin login) and recorded demo video (MP4).
- **Developer documentation**
 - API reference (OpenAPI/Swagger JSON), data model diagrams, deployment guide (Docker-compose and Kubernetes manifest examples).
- **Standards**
 - Documentation will use accessible HTML (semantic tags) and PDFs generated from the canonical markdown source.

2.7 Assumptions and Dependencies

Important assumptions and external dependencies that affect the project:

- **Assumptions**
 - The institution will provide hosting (VMs, storage, and domain) or approve cloud hosting.
 - Campus legal/privacy team will provide retention and disclosure policies early in the design phase.
 - Staff who handle reports will have or receive training in handling confidential/anonymous reports.
 - Users have access to the internet and modern browsers.
- **Dependencies**
 - External SMTP / SMS provider for notifications (optional for anonymous flow).
 - Authentication/SSO provider for admin users (LDAP/SAML/OAuth2) if required by institution.
 - Object storage (S3) for attachments and evidence; otherwise filesystem with proper backups.
 - Database system (PostgreSQL/MySQL) and regular backups.
 - Optional integration endpoints for campus case-management / incident-tracking systems (APIs).
- **Risk if assumptions fail**
 - If hosting or legal policies are not provided, deployment and retention plans will be delayed; if SSO is unavailable, an alternative secure admin account mechanism must be implemented.

3. System Features

Below are grouped functional areas with sample features, stimulus/response sequences, and detailed functional requirements.

3.1 Feature 1: Anonymous Report Submission

3.1.1 Description and Priority

- Allows users to file harassment reports anonymously with optional identity details.
- Priority: High (core functionality).

3.1.2 Stimulus/Response Sequence

- User clicks Add Report.
- System displays a secure form (name, age, gender, harassment details, date, place, optional proof).
- User submits → system generates unique reference code.
- System confirms submission and stores encrypted report.

3.1.3 Functional Requirements

- REQ-1: The system shall allow users to submit reports anonymously without mandatory personal details.
- REQ-2: The system shall generate a unique reference code for each report.
- REQ-3: The system shall encrypt and securely store report details and uploaded files.

3.2 Feature 2: Report Tracking

3.2.1 Description and Priority

- Allows users to track the status of their report using the reference code.
- Priority: High.

3.2.2 Stimulus/Response Sequence

- User clicks Track Report Status.
- System asks for reference code.
- System retrieves and displays the report's review status.

3.2.3 Functional Requirements

- REQ-4: The system shall validate reference codes before displaying status.
- REQ-5: The system shall ensure that no personally identifiable data is leaked during status display.

3.3 Feature 3: Admin and Safety Officer Review

3.3.1 Description and Priority

- Admins and safety officers can review, categorize, and resolve reports.
- Priority: High.

3.3.2 Stimulus/Response Sequence

- Admin/Safety Officer logs in.
- System displays assigned reports.
- They review, update status, and take necessary actions.

3.3.3 Functional Requirements

- REQ-6: The system shall require secure login for admin and safety officers.
- REQ-7: The system shall allow authorized personnel to update report status.
- REQ-8: The system shall log all actions for audit purposes.

4. External Interface Requirements

4.1 User Interfaces

- **Web-based Interface:**
The platform provides a responsive, browser-based interface built using HTML5, CSS3, and JavaScript, integrated with JSP and Java Servlets on the backend. The UI adopts a modern cybersecurity-themed design consistent with institutional branding.
- **Key Features and Buttons:**
Common user actions include:
 - Submit Report – Opens the anonymous report form modal.
 - Attach Proof – Allows optional file upload (images, PDFs, documents).
 - Track Status – Opens the tracking form for entering a reference code.
 - Admin / Officer Login – Restricted access for reviewing and managing submitted reports.
 - Close – Closes modal dialogs.
- **Accessibility:**
The UI conforms to WCAG 2.1 AA accessibility standards, ensuring compatibility with screen readers, keyboard navigation, and high-contrast viewing.
- **Anonymity Consideration:**
Reporters do not need to register or log in. Personal details are optional and collected only if voluntarily shared.

4.2 Hardware Interfaces

- **Client Devices:**
The system is accessible through any modern device (desktop, laptop, tablet, or smartphone) with an up-to-date web browser supporting JavaScript and file uploads.
- **Peripheral Support:**
No dedicated hardware peripherals are required. The platform supports scanning or uploading attachments (evidence, documents, or screenshots) directly from client devices via browser file input controls.
- **Server-Side Requirements:**
If deployed on-premises, the application may require integration with existing institutional network storage (e.g., SMB/NFS) for secure file retention. These interactions follow standard authenticated network mount protocols.

4.3 Software Interfaces

- **Database:**
The system uses MySQL 8+ as the primary database, connected via JDBC. Schema includes tables for:
 - user_reports – stores anonymous report details and reference codes.
 - Optional future tables for officers, admins, and audit_logs.
Database connections occur over secure TLS channels when deployed in production.
- **Object Storage:**
File attachments (proof documents) are securely stored in the application server directory or optionally in an S3-compatible object storage such as MinIO or AWS S3. Bucket or directory permissions ensure that uploaded files are not publicly accessible and can only be retrieved by authorized admins.
- **Authentication:**
The platform currently uses local authentication for admin and officer logins (admin / admin123). Future versions may integrate LDAP or SSO (SAML/OAuth2) for institutional authentication.
Reporters remain anonymous and are not authenticated.
- **Email/SMS (Optional):**
Integration with SMTP-based mail servers (e.g., SendGrid, Gmail SMTP) may be used for notifications to administrators or safety officers.
Notifications do not include reporter identity to preserve privacy.
- **APIs:**
The system may provide RESTful API endpoints for external integration, following REST/JSON conventions:
 - POST /api/reports – Submit a new report (with attachment metadata).
 - GET /api/reports/{referenceCode} – Retrieve report status for tracking.
 - GET /api/admin/reports – View submitted reports (admin only).

- PUT /api/admin/reports/{referenceCode}/assign – Assign officer to report.
All API endpoints require HTTPS and proper authorization (for admin routes).

- **Data Flows:**

- Incoming:
 - Report submission form data (age, gender, description, location, incident date).
 - Optional uploaded files (proof).
- Outgoing:
 - Reference code display for the reporter (via success.jsp).
 - Optional admin notifications.
 - Report data retrieval for tracking and administrative review.

4.4 Communications Interfaces

- **Protocols**
 - HTTPS (TLS 1.2/1.3) for all web/API traffic.
 - SMTP for email (STARTTLS).
 - S3 over HTTPS for object storage.
- **Message formats**
 - JSON for REST APIs; multipart/form-data for file uploads.
- **Security**
 - OAuth2 / SAML for admin authentication where used; JWT tokens for internal services with short TTLs.
 - Webhooks should be HMAC-signed and validated on receipt.
- **Performance**
 - File uploads should support resumable uploads (optional for large files) and provide progress reporting.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system shall handle at least 100 concurrent anonymous submissions per minute with median response time < 2 seconds for form submission (excluding large file uploads).
- Admin dashboard list queries for the last 30 days shall return within 3 seconds for up to 10k report records (with pagination).
- File download latency for attachments shall be under 5 seconds for files \leq 10 MB when served from object store via CDN.

5.2 Safety Requirements

- The product must include guidance and disclaimers: it is not a replacement for emergency services. Prominent emergency contact information and “If you are in immediate danger, call emergency services” must be present on the submission page.
- File scanning for malware: attachments must be scanned (virus/malware) on upload and quarantined if suspicious.
- Admin actions that could disclose reporter identity must be explicitly blocked or require elevated privileges and recorded.

5.3 Security Requirements

- **Authentication & Authorization**
 - Admin access must require institutional SSO with MFA; local admin accounts allowed only for emergency with strict policies.
 - RBAC with least privilege principle.
- **Data protection**
 - Sensitive fields and attachments must be encrypted at rest (AES-256 or equivalent).
 - All network communication must use HTTPS; database connections use TLS.
- **Anonymity guarantees**
 - Default behavior: do not store or associate PII with anonymous reports.
 - If reporter provides contact info, store it only with explicit consent and mark record as contactable; staff must have an explicit action to reveal contact info.
- **Audit & logging**
 - Immutable audit logs of admin activity (who did what, when) retained per policy.

- Logs shall not contain reporter IP or user-agent unless explicit consent is logged.

- **Vulnerability management**

- Periodic dependency scans and security testing (SAST/DAST) and patching policy.

5.4 Software Quality Attributes

- **Availability:** Target 99.5% uptime (excluding planned maintenance) for the application.
- **Reliability:** System shall survive transient failures (retries, graceful degradation). No single point of failure in production (use redundant components if hosted).
- **Maintainability:** Modular codebase with documented APIs and tests; CI/CD pipelines for automated builds and tests.
- **Portability:** Deployable on common cloud providers and on-premises Linux environments; use Docker containers where possible.
- **Usability:** Minimal form complexity for reporters; accessible and mobile-friendly.
- **Testability:** Unit tests, integration tests, and test data generation provided; acceptance test cases map to REQ-IDs.

6. Other Requirements

- **Database requirements**

- Reports and audit logs partitioned by date for efficient queries and retention management.

- **Internationalization**

- UI text externalized into resource bundles for future localization.

- **Legal**

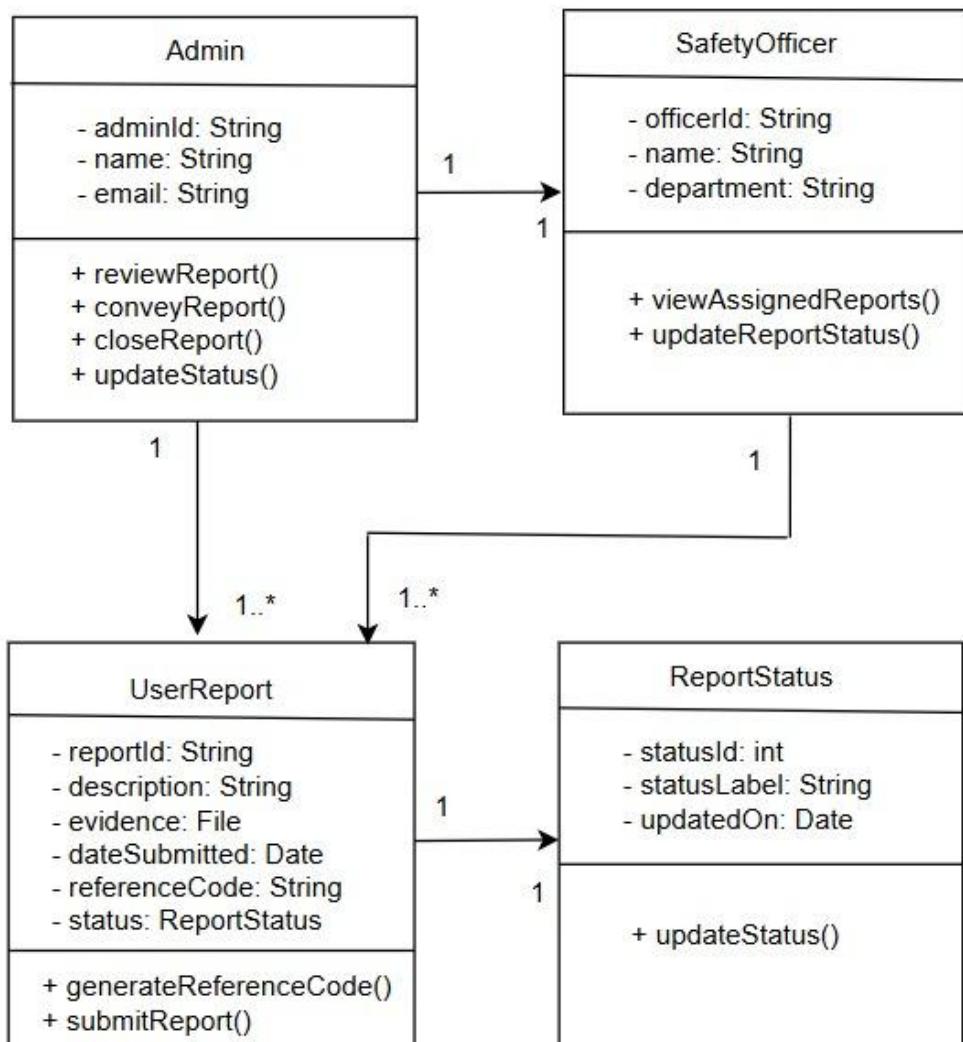
- Mechanism to support legal holds on specific reports (prevent automated deletion) when required by investigations.

- **Reuse objectives**

- Exportable anonymized datasets to support campus analytics while preserving

APPENDIX : ANALYSIS MODELS

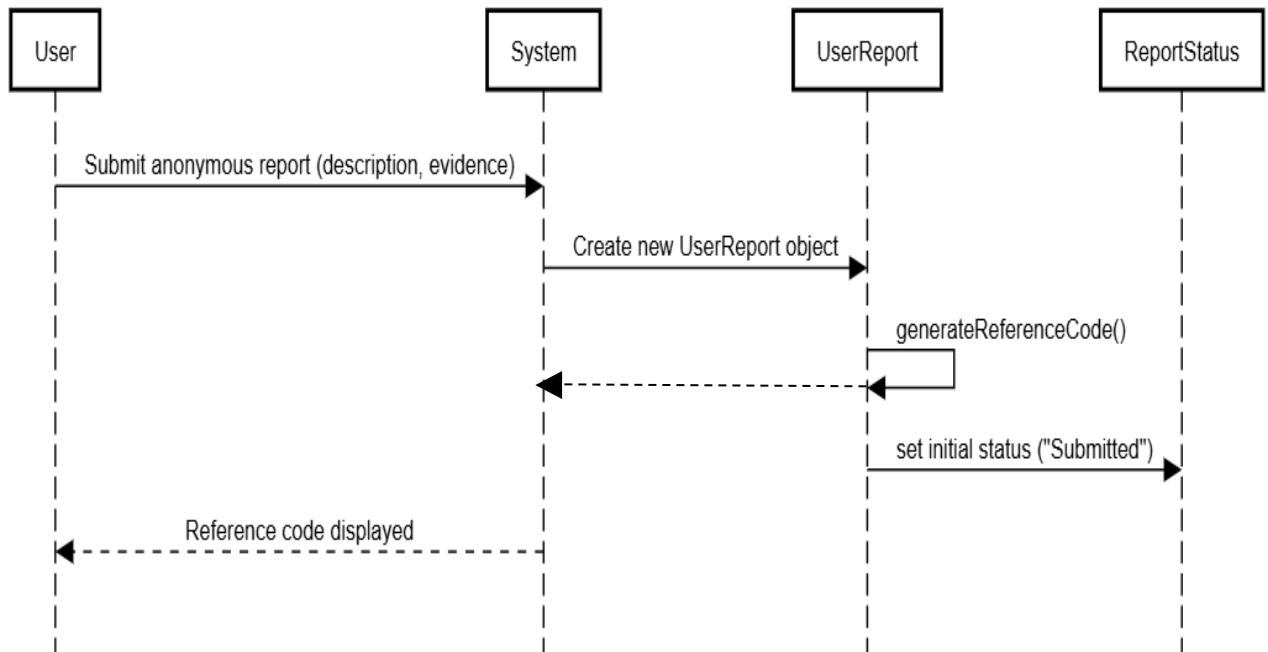
Class Diagram:



Sequence Diagram:

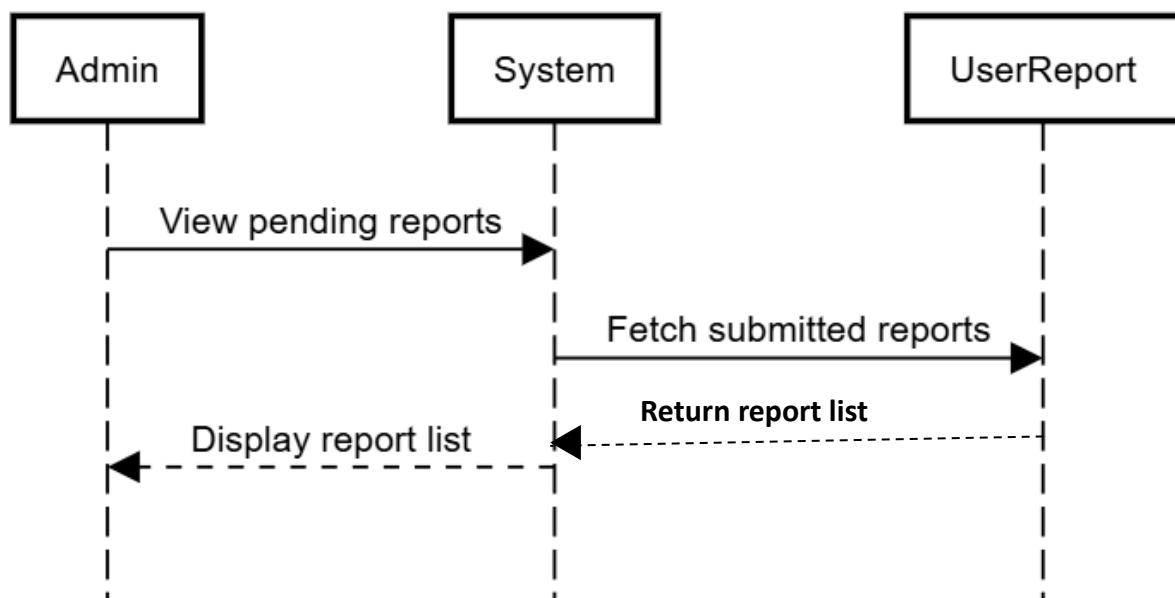
1. Sequence Diagram for Report Submission and Reference Code Generation

Report Submission and Reference Code Generation



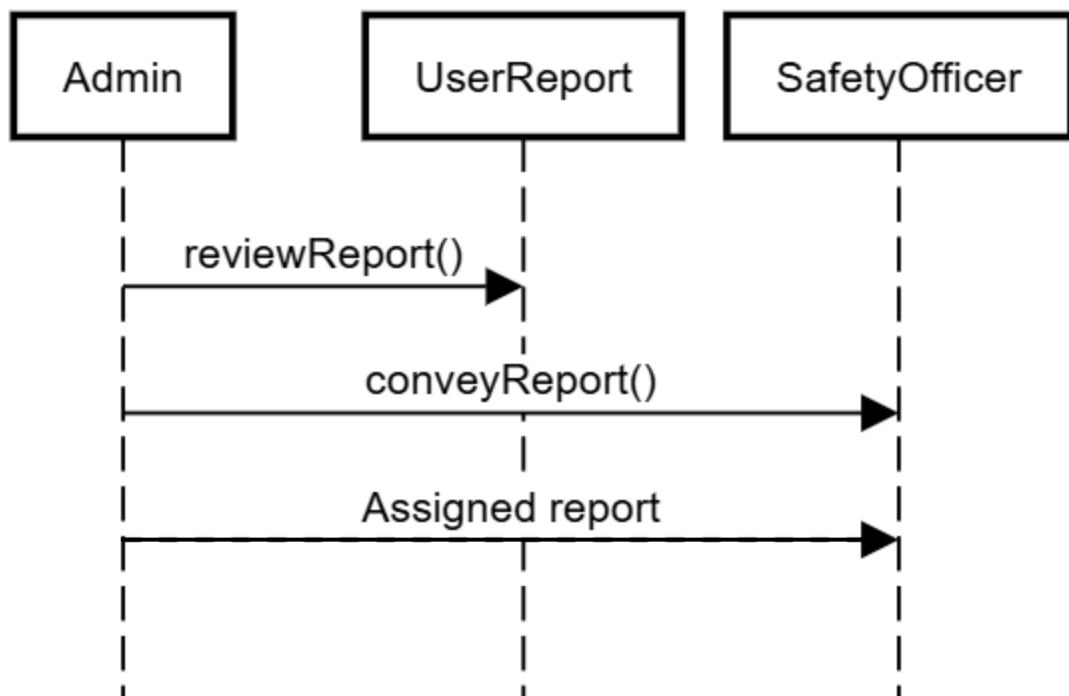
2. Sequence diagram for Admin Views Pending Reports

Admin Views Pending Reports



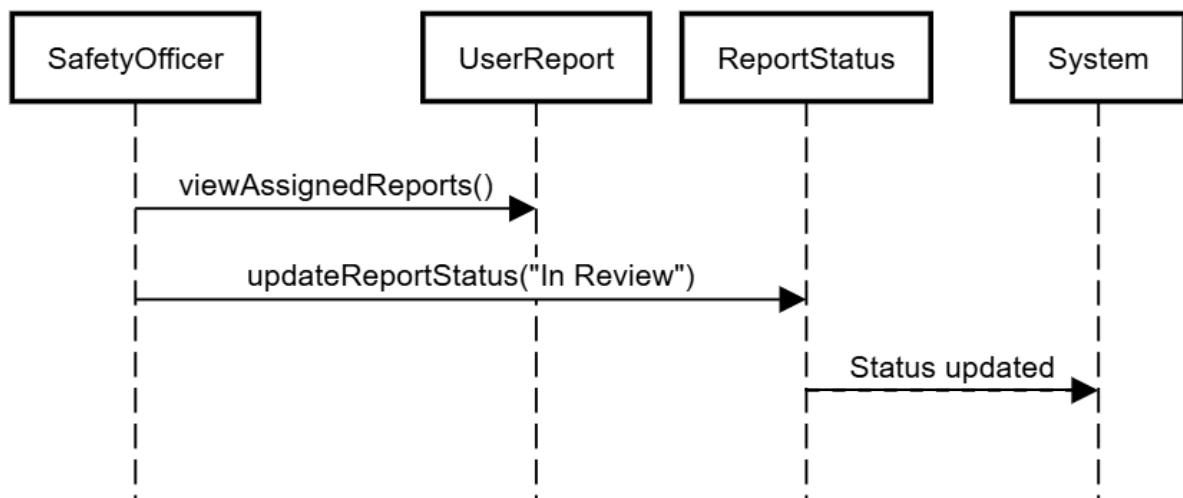
3. Sequence Diagram for Report Assignment by Admin

Report Assignment by Admin

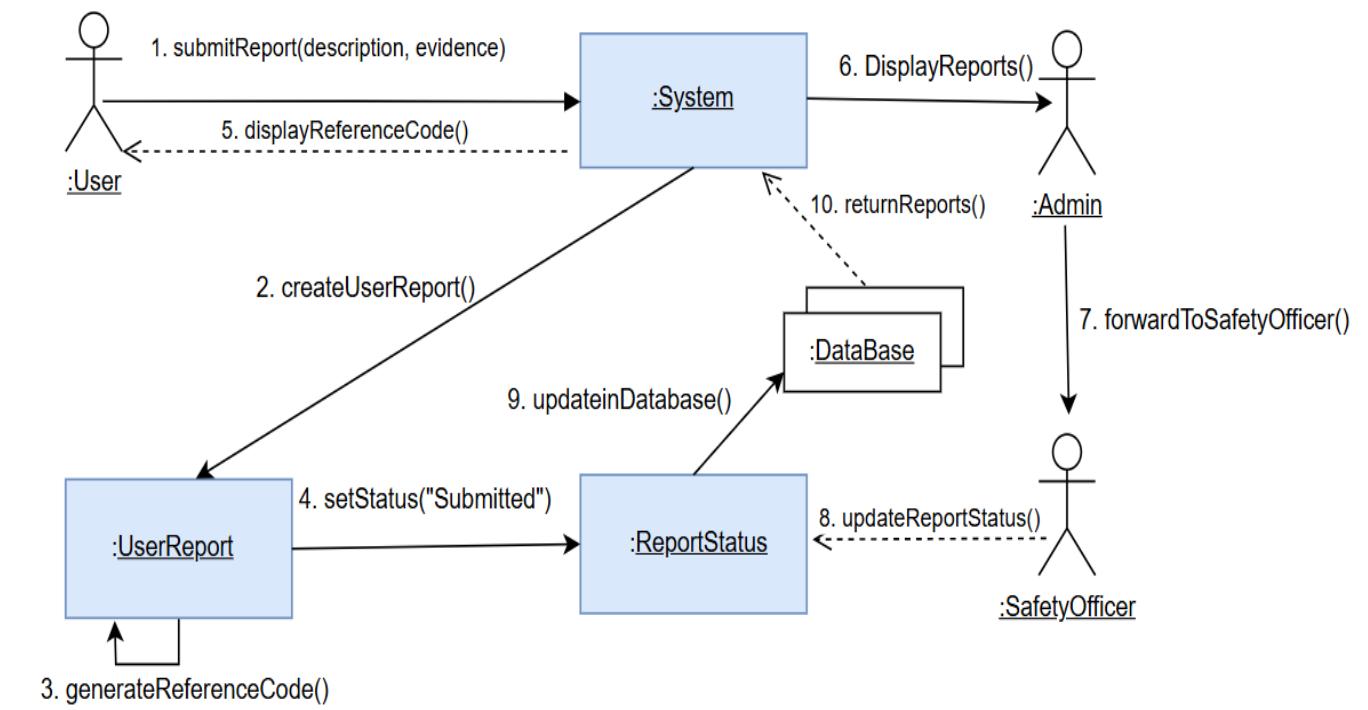


4. Sequence Diagram for Safety Officer Reviews and Updates Report

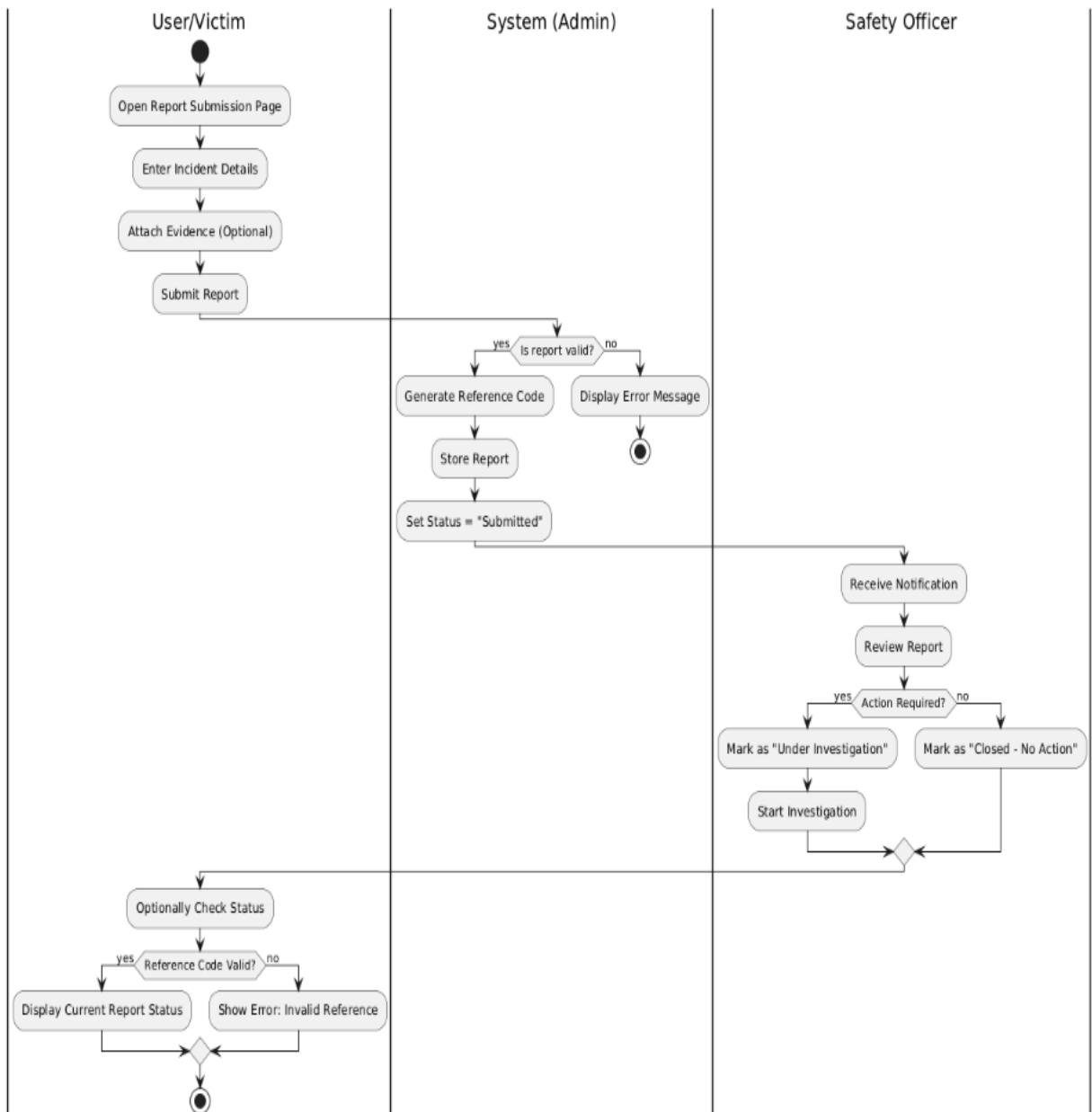
Safety Officer Reviews and Updates Report



Collaboration Diagram:

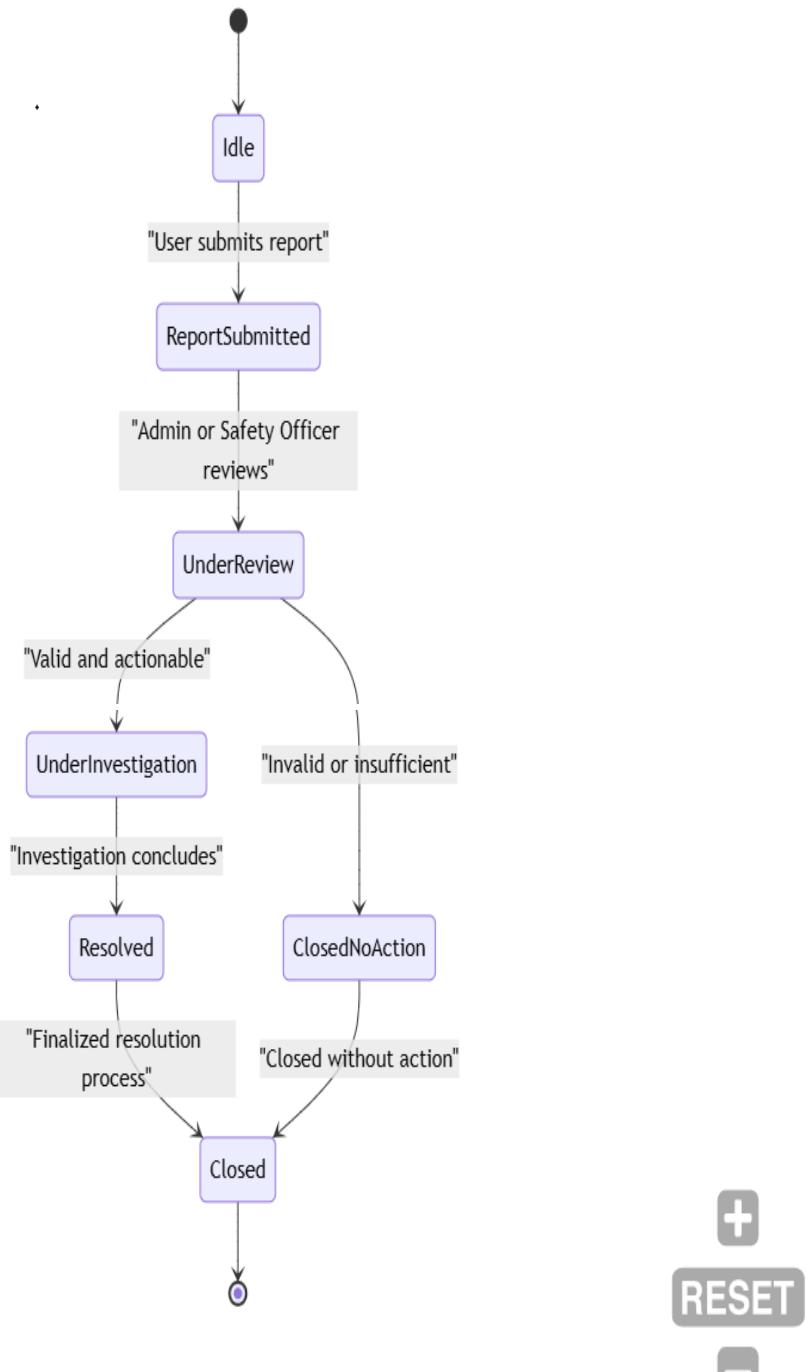


Activity diagram:



State Chart diagram:

1. State Chart diagram for Anonymous Harassment Reporting Platform

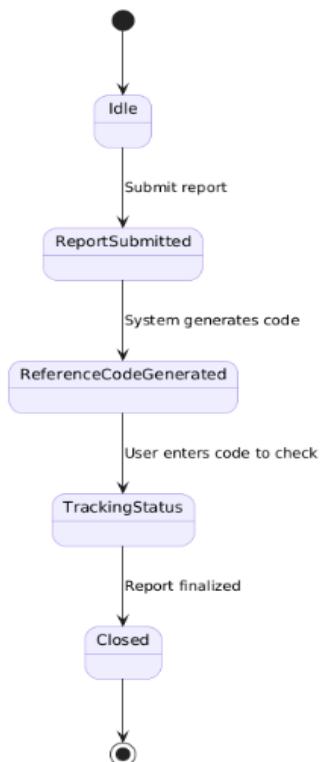


RESET

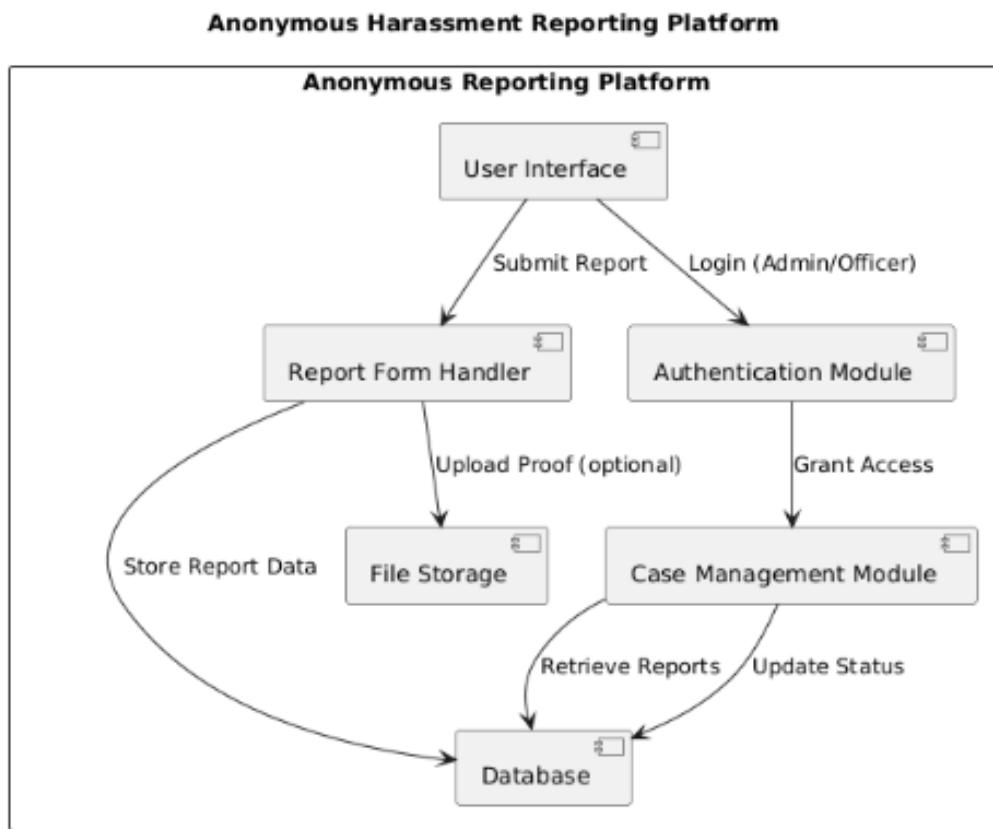
2. State chart diagram for Admin/safety officer view



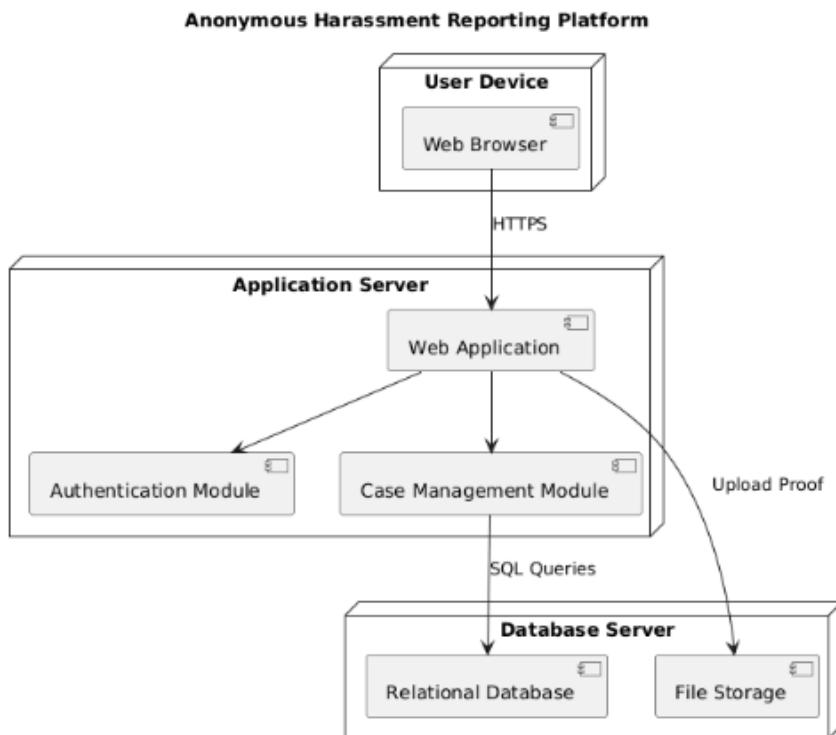
3. State chart diagram for user/victim view



Component Diagram:



Deployment Diagram:



Source Code:

Index.html:

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Anonymous Harassment Reporting Platform</title>
<style>
header { position: fixed; top:0; left:0; width:100%; padding:10px; text-align:center; background: rgba(0,15,40,0.85); border-bottom:2px solid #00c6ff; z-index:1000; }
body { margin:0; font-family:'Segoe UI',sans-serif; background:url("https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg") no-repeat center center fixed; background-size:cover; color:#e0f7ff; padding-top:70px; }
header h1 { margin:0; font-size:28px; color:#00c6ff; text-shadow:0 0 10px #00c6ff,0 0 20px #0093ff; }
.top-right { position:absolute; top:20px; right:20px; }
.top-right button { background: rgba(0,20,50,0.85); color:#00c6ff; border:1px solid #00c6ff; padding:8px 14px; margin-left:10px; border-radius:6px; cursor:pointer; font-weight:bold; transition:0.3s; }
.top-right button:hover { background:#00c6ff; color:#000; box-shadow:0 0 12px #00c6ff; }
.right-center { position:absolute; top:55%; right:6%; transform:translateY(-50%); width:500px; text-align:center; opacity:0; animation:fadeIn 2s ease forwards; }
.right-center .assurance-message { max-width:100%; margin:0 auto 20px auto; padding:28px; font-size:1.3rem; line-height:1.7; background: rgba(0,20,40,0.65); border:1px solid #00bcd4; border-radius:14px; box-shadow:0 0 15px #00e5ff,0 0 30px #00bcd4; backdrop-filter:blur(6px); }
.right-center button { background: rgba(0,30,70,0.85); color:#fff; border:2px solid #00c6ff; padding:15px 25px; border-radius:10px; font-size:20px; cursor:pointer; text-shadow:0 0 12px #00c6ff; transition:0.3s; margin:10px; }
.right-center button:hover { background:#00c6ff; color:#000; box-shadow:0 0 20px #00c6ff; }
@keyframes fadeIn { from { opacity:0; transform:translateY(-60%); } to { opacity:1; transform:translateY(-50%); } }
.overlay { position:fixed; top:0; left:0; width:100%; height:100%; background:rgba(0,0,0,0.7); display:none; justify-content:center; align-items:center; z-index:1000; overflow-y:auto; padding:20px; }
.container { width:500px; background:rgba(0,0,20,0.95); padding:20px; border-radius:10px; border:1px solid #00c6ff; box-shadow:0 0 20px rgba(0,198,255,0.4); animation:fadeInModal 0.4s ease; margin:auto; max-height:90vh; overflow-y:auto; }
.container h2 { font-size:20px; color:#00c6ff; margin-bottom:12px; text-shadow:0 0 8px #00c6ff; text-align:center; }
form-group { margin:6px 0; }
label { display:block; margin-bottom:3px; font-size:13px; font-weight:500; }
input, select, textarea { width:100%; padding:6px; font-size:13px; border:1px solid #00c6ff; border-radius:5px; outline:none; background:#001020; color:#e0f7ff; box-sizing:border-box; }
textarea { resize:vertical; min-height:50px; }
input[type="file"] { padding:8px; cursor:pointer; }
input[type="file"]::file-selector-button { background:#00c6ff; border:none; padding:6px 12px; border-radius:4px; color:#000; font-weight:bold; cursor:pointer; margin-right:10px; transition:0.3s; }
input[type="file"]::file-selector-button:hover { background:#00e5ff; box-shadow:0 0 8px #00c6ff; }
file-note { font-size:11px; color:#a0d7ff; margin-top:3px; font-style:italic; }
button.action { background:#00c6ff; border:none; padding:10px 16px; border-radius:6px; margin-top:10px; cursor:pointer; font-weight:bold; color:#000; transition:0.3s; }
button.action:hover { background:#fff; color:#000; box-shadow:0 0 15px #00c6ff; }
@keyframes fadeInModal { from { opacity:0; transform:translateY(-20px); } to { opacity:1; transform:translateY(0); } }
</style>
</head>
<body>
<header>
<h1>Anonymous Harassment Reporting Platform</h1>
<div class="top-right">
<button onclick="openModal('adminLogin')">Admin</button>
<button onclick="openModal('officerLogin')">Safety Officer</button>
</div>
</header>
<div class="right-center">
<div class="assurance-message">
<p>
    You are not alone, and your voice deserves to be heard.
    This platform is designed to provide you with a safe, anonymous, and secure space
    to share your experience without fear of judgment or retaliation.

```

Every detail you submit is kept confidential and taken seriously,
 ensuring that your concerns are addressed with dignity, empathy, and respect.
 By speaking up, you take the first step towards justice

```

</p>
</div>
<button onclick="openModal('reportForm')">  Add Report</button>
<button onclick="openModal('trackReport')">  Track Report</button>
</div>
<div class="overlay" id="overlay">
<!-- Submit Report Form -->
<div class="container" id="reportForm">
<h2>Submit a Report</h2>
<form action="SubmitReportServlet" method="post" enctype="multipart/form-data">
<div class="form-group"><label>Age</label><input type="number" name="age" required></div>
<div class="form-group"><label>Gender</label>
<select name="gender" required>
<option value="">Select</option>
<option>Male</option>
<option>Female</option>
<option>Other</option>
</select>
</div>
<div class="form-group"><label>Harassment Information / Details</label><textarea name="details" required></textarea></div>
<div class="form-group"><label>Date of Incident</label><input type="date" name="incident_date" required></div>
<div class="form-group"><label>Place of Incident</label><input type="text" name="location" required></div>
<div class="form-group">
<label>Attach Proof (Optional)</label>
<input type="file" name="proofFile" accept="image/*,.pdf,.doc,.docx"/>
<div class="file-note">Supported: Images, PDF, Word documents (Max 10MB)</div>
</div>
<button type="submit" class="action">Submit Report</button>
<button type="button" class="action" onclick="closeModal()">Close</button>
</form>
</div>
<!-- Track Report Form -->
<div class="container" id="trackReport" style="display:none;">
<h2>Track Your Report</h2>
<form action="TrackReportServlet" method="post">
<div class="form-group"><label>Reference Code</label><input type="text" name="referenceCode" required></div>
<button type="submit" class="action">Check Status</button>
<button type="button" class="action" onclick="closeModal()">Close</button>
</form>
</div>
<!-- Admin Login -->
<div class="container" id="adminLogin" style="display:none;">
<h2>Admin Login</h2>
<form action="AdminLoginServlet" method="post">
<div class="form-group"><label>Username</label><input type="text" name="username" required></div>
<div class="form-group"><label>Password</label><input type="password" name="password" required></div>
<button type="submit" class="action">Login</button>
<button type="button" class="action" onclick="closeModal()">Close</button>
</form>
</div>
<!-- Safety Officer Login -->
<div class="container" id="officerLogin" style="display:none;">
<h2>Safety Officer Login</h2>
<form action="OfficerLoginServlet" method="post">
<div class="form-group"><label>Username</label><input type="text" name="username" required></div>
<div class="form-group"><label>Password</label><input type="password" name="password" required></div>
<button type="submit" class="action">Login</button>
<button type="button" class="action" onclick="closeModal()">Close</button>
</form>
</div>
</div>
```

```

<script>
  function openModal(id){
    document.getElementById("overlay").style.display="flex";
    document.querySelectorAll(".container").forEach(c=>c.style.display="none");
    document.getElementById(id).style.display="block";
  }
  function closeModal() { document.getElementById("overlay").style.display="none"; }
</script>
</body>
</html>
adminDashboard.jsp:
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Admin Dashboard</title>
<style>
  body {
    margin: 0;
    font-family: 'Segoe UI', sans-serif;
    background: url("https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg") no-repeat center center fixed;
    background-size: cover;
    color: #e0f7ff;
    padding: 20px;
    min-height: 100vh;
  }
  .header-section {
    text-align: center;
    margin-bottom: 30px;
    animation: fadeIn 1s ease;
  }
  h2 {
    font-size: 32px;
    color: #00c6ff;
    text-shadow: 0 0 10px #00c6ff, 0 0 20px #0093ff;
    margin: 0;
  }
  @keyframes fadeIn {
    from { opacity: 0; transform: translateY(-20px); }
    to { opacity: 1; transform: translateY(0); }
  }
  .success-message {
    max-width: 600px;
    margin: 0 auto 30px auto;
    background: rgba(0, 198, 255, 0.15);
    border: 2px solid #00c6ff;
    border-radius: 12px;
    padding: 20px;
    display: flex;
    align-items: center;
    gap: 15px;
    animation: slideDown 0.5s ease;
    box-shadow: 0 0 20px rgba(0, 198, 255, 0.4);
  }
  @keyframes slideDown {
    from {
      opacity: 0;
      transform: translateY(-30px);
    }
    to {
      opacity: 1;
      transform: translateY(0);
    }
  }
</style>

```

```

        }
    }
.success-icon {
    font-size: 40px;
    color: #00c6ff;
    animation: successPulse 1.5s ease-in-out infinite;
    line-height: 1;
}
@keyframes successPulse {
    0%, 100% {
        transform: scale(1);
    }
    50% {
        transform: scale(1.1);
    }
}
.success-text {
    flex: 1;
    text-align: left;
}
.success-text h3 {
    margin: 0 0 5px 0;
    color: #00c6ff;
    font-size: 18px;
    text-shadow: 0 0 8px rgba(0, 198, 255, 0.6);
}
.success-text p {
    margin: 0;
    color: #e0f7ff;
    font-size: 14px;
}
.close-success {
    background: rgba(0, 198, 255, 0.2);
    border: 1px solid #00c6ff;
    color: #00c6ff;
    padding: 8px 16px;
    border-radius: 6px;
    cursor: pointer;
    font-weight: bold;
    transition: all 0.3s ease;
}
.close-success:hover {
    background: rgba(0, 198, 255, 0.4);
    transform: translateY(-2px);
    box-shadow: 0 0 15px rgba(0, 198, 255, 0.4);
}
form {
    max-width: 1400px;
    margin: 0 auto;
    background: rgba(0, 20, 40, 0.85);
    padding: 30px;
    border-radius: 12px;
    border: 2px solid #00c6ff;
    box-shadow: 0 0 25px rgba(0, 198, 255, 0.4);
    backdrop-filter: blur(8px);
}
table {
    border-collapse: collapse;
    width: 100%;
    margin-top: 20px;
    background: rgba(0, 15, 30, 0.7);
    border-radius: 8px;
    overflow: hidden;
    box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);
}
th, td {
    border: 1px solid #00c6ff;
}

```

```

padding: 12px 10px;
color: #e0f7ff;
text-align: left;
}
  |
```

```

        transform: translateY(-3px);
        box-shadow: 0 0 30px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.6);
    }
.button-container {
    display: flex;
    gap: 15px;
    justify-content: center;
    margin-top: 20px;
}
.back-button {
    background: linear-gradient(135deg, rgba(0, 198, 255, 0.2) 0%, rgba(0, 147, 255, 0.2) 100%);
    color: #00c6ff;
    border: 2px solid #00c6ff;
    padding: 12px 25px;
    font-size: 16px;
    text-decoration: none;
    display: inline-block;
    border-radius: 8px;
    font-weight: bold;
    transition: all 0.3s ease;
    box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);
}
.back-button:hover {
    background: linear-gradient(135deg, rgba(0, 198, 255, 0.4) 0%, rgba(0, 147, 255, 0.4) 100%);
    color: #fff;
    transform: translateY(-3px);
    box-shadow: 0 0 25px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.5);
}
::-webkit-scrollbar {
    width: 10px;
    height: 10px;
}
::-webkit-scrollbar-track {
    background: rgba(0, 15, 40, 0.5);
}
::-webkit-scrollbar-thumb {
    background: #00c6ff;
    border-radius: 5px;
}
::-webkit-scrollbar-thumb:hover {
    background: #00e5ff;
}

```

</style>

</head>

<body>

<div class="header-section">

<h2>Admin Dashboard</h2>

</div>

<%>

String confirmation = (String) request.getAttribute("confirmation");

if (confirmation != null) {

%>

<div class="success-message" id="successMessage">

<div class="success-icon">/</div>

<div class="success-text">

<h3>Success!</h3>

<p><%= confirmation %></p>

</div>

<button class="close-success"

onclick="document.getElementById('successMessage').style.display='none'">x</button>

</div>

<%>

<%>

<form method="post" action="AssignOfficerServlet">

```

<table>
<tr>
<th>Ref Code</th>
<th>Age</th>
<th>Gender</th>
<th>Description</th>
<th>Date</th>
<th>Place</th>
<th>Status</th>
<th>Assign Officer</th>
</tr>
<%
Class.forName("com.mysql.cj.jdbc.Driver");
Connection conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/harassment_reporting", "root", "root");
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM user_reports ORDER BY created_at DESC");
while(rs.next()) {
String ref = rs.getString("referenceCode");
int age = rs.getInt("age");
String gender = rs.getString("gender");
String desc = rs.getString("description");
java.sql.Date date = rs.getDate("date_of_incident");
String place = rs.getString("place");
String status = rs.getString("status");
String officer = rs.getString("officer");
%>
<tr>
<td><%= ref %></td>
<td><%= age %></td>
<td><%= gender %></td>
<td><%= desc %></td>
<td><%= date %></td>
<td><%= place %></td>
<td><%= status %></td>
<td>
<select name="officer_<%= ref %>">
<option value="">Select Officer</option>
<option value="officer1" <%= "officer1".equals(officer) ? "selected" : "" %>>Officer1</option>
<option value="officer2" <%= "officer2".equals(officer) ? "selected" : "" %>>Officer2</option>
<option value="officer3" <%= "officer3".equals(officer) ? "selected" : "" %>>Officer3</option>
</select>
</td>
</tr>
<%
}
rs.close();
stmt.close();
conn.close();
%>
</table>
<div class="button-container">
<a href="index.html" class="back-button">← Go Back Home</a>
<button type="submit" class="assign-button">Assign Officers</button>
</div>
</form>
</body>
</html>
officerDashboard.jsp:
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.*" %>
<!DOCTYPE html>
<html>
<head>
<title>Officer Dashboard</title>
<style>
```

```

body {
    margin: 0;
    font-family: 'Segoe UI', sans-serif;
    background: url("https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg") no-repeat center center fixed;
    background-size: cover;
    color: #e0f7ff;
    padding: 20px;
    min-height: 100vh;
}
h2, h3 {
    text-align: center;
    color: #00c6ff;
    text-shadow: 0 0 10px #00c6ff, 0 0 20px #0093ff;
    animation: fadeIn 1s ease;
}
h2 {
    font-size: 32px;
    margin-bottom: 10px;
}
h3 {
    font-size: 24px;
    margin-top: 5px;
    margin-bottom: 30px;
    color: #00e5ff;
}
@keyframes fadeIn {
    from { opacity: 0; transform: translateY(-20px); }
    to { opacity: 1; transform: translateY(0); }
}
.dashboard-container {
    max-width: 1400px;
    margin: 0 auto;
    background: rgba(0, 20, 40, 0.85);
    padding: 30px;
    border-radius: 12px;
    border: 2px solid #00c6ff;
    box-shadow: 0 0 25px rgba(0, 198, 255, 0.4);
    backdrop-filter: blur(8px);
}
table {
    border-collapse: collapse;
    width: 100%;
    background: rgba(0, 15, 30, 0.7);
    border-radius: 8px;
    overflow: hidden;
    box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);
}
th, td {
    border: 1px solid #00c6ff;
    padding: 12px 10px;
    text-align: center;
    color: #e0f7ff;
}
th {
    background: linear-gradient(135deg, #001020 0%, #002040 100%);
    font-weight: bold;
    font-size: 14px;
    text-transform: uppercase;
    letter-spacing: 0.5px;
    text-shadow: 0 0 8px rgba(0, 198, 255, 0.6);
    color: #00c6ff;
    position: sticky;
    top: 0;
    z-index: 10;
}
td {

```

```

background-color: rgba(0, 12, 26, 0.6);
font-size: 13px;
transition: background-color 0.3s ease;
}
tr:hover td {
background-color: rgba(0, 60, 100, 0.7);
}
select, button, input {
padding: 6px 10px;
margin: 2px;
background: rgba(0, 20, 50, 0.9);
color: #00c6ff;
border: 1px solid #00c6ff;
border-radius: 5px;
font-size: 13px;
outline: none;
transition: all 0.3s ease;
}
select {
cursor: pointer;
min-width: 120px;
}
::-webkit-scrollbar {
width: 10px;
height: 10px;
}
::-webkit-scrollbar-track {
background: rgba(0, 15, 40, 0.5);
}
::-webkit-scrollbar-thumb {
background: #00c6ff;
border-radius: 5px;
}
::-webkit-scrollbar-thumb:hover {
background: #00e5ff;
}
td:nth-child(7) {
font-weight: bold;
text-transform: uppercase;
font-size: 12px;
letter-spacing: 0.5px;
}
.button-container {
display: flex;
gap: 15px;
justify-content: center;
margin-top: 30px;
}
.update-button {
padding: 14px 30px;
font-size: 18px;
box-shadow: 0 0 20px rgba(0, 198, 255, 0.3);
}
.update-button:hover {
transform: translateY(-3px);
box-shadow: 0 0 30px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.6);
}
.back-button {
background: linear-gradient(135deg, rgba(0, 198, 255, 0.2) 0%, rgba(0, 147, 255, 0.2) 100%);
color: #00c6ff;
border: 2px solid #00c6ff;
padding: 14px 30px;
font-size: 18px;
text-decoration: none;
display: inline-block;
border-radius: 8px;
font-weight: bold;
}

```

```

transition: all 0.3s ease;
box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);
}
.back-button:hover {
background: linear-gradient(135deg, rgba(0, 198, 255, 0.4) 0%, rgba(0, 147, 255, 0.4) 100%);
color: #fff;
transform: translateY(-3px);
box-shadow: 0 0 25px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.5);
}
</style>
</head>
<body>
<h2>Welcome <%= request.getAttribute("officerName") %></h2>
<h3>Assigned Reports</h3>
<%
String officer = (String) request.getAttribute("officerName");

Class.forName("com.mysql.cj.jdbc.Driver");
Connection conn = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/harassment_reporting", "root", "root");

PreparedStatement ps = conn.prepareStatement(
"SELECT referenceCode, age, gender, description, date_of_incident, place, status FROM user_reports WHERE
officer=? ORDER BY created_at DESC"
);
ps.setString(1, officer);
ResultSet rs = ps.executeQuery();
%>
<form method="post" action="UpdateReportStatusServlet">
<div class="dashboard-container">
<table>
<tr>
<th>Ref Code</th>
<th>Age</th>
<th>Gender</th>
<th>Description</th>
<th>Date</th>
<th>Place</th>
<th>Status</th>
<th>Update Status</th>
</tr>
<%
while(rs.next()) {
String ref = rs.getString("referenceCode");
int age = rs.getInt("age");
String gender = rs.getString("gender");
String desc = rs.getString("description");
java.sql.Date date = rs.getDate("date_of_incident");
String place = rs.getString("place");
String status = rs.getString("status");
%>
<tr>
<td><%= ref %></td>
<td><%= age %></td>
<td><%= gender %></td>
<td><%= desc %></td>
<td><%= date %></td>
<td><%= place %></td>
<td><%= status %></td>
<td>
<input type="hidden" name="referenceCode" value="<%= ref %>">
<select name="status" required>
<option value="Submitted" <%= "Submitted".equals(status) ? "selected" : "" %>>Submitted</option>
<option value="In Progress" <%= "In Progress".equals(status) ? "selected" : "" %>>In Progress</option>
<option value="Resolved" <%= "Resolved".equals(status) ? "selected" : "" %>>Resolved</option>
<option value="Closed" <%= "Closed".equals(status) ? "selected" : "" %>>Closed</option>
</select>

```

```

</td>
</tr>
<%
}
rs.close();
ps.close();
conn.close();
%>
</table>
<div class="button-container">
<a href="index.html" class="back-button">← Go Back</a>
<button type="submit" class="update-button">Update</button>
</div>
</div>
</form>
</body>
</html>

```

Success.jsp:

```

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
<title>Report Submitted</title>
<style>
body {
    margin: 0;
    font-family: 'Segoe UI', sans-serif;
    background: url("https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg") no-repeat center center fixed;
    background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    padding: 20px;
    color: #e0f7ff;
}

.container {
    background: rgba(0, 20, 40, 0.95);
    padding: 40px;
    border-radius: 15px;
    border: 2px solid #00c6ff;
    box-shadow: 0 0 30px rgba(0, 198, 255, 0.5), 0 0 50px rgba(0, 198, 255, 0.3);
    backdrop-filter: blur(10px);
    text-align: center;
    width: 100%;
    max-width: 550px;
    animation: fadeInScale 0.6s ease;
}

@keyframes fadeInScale {
    from {
        opacity: 0;
        transform: scale(0.9) translateY(30px);
    }
    to {
        opacity: 1;
        transform: scale(1) translateY(0);
    }
}

.success-icon {
    font-size: 80px;
    color: #00c6ff;
}

```

```

margin-bottom: 20px;
animation: pulse 2s infinite;
text-shadow: 0 0 20px #00c6ff, 0 0 40px #00c6ff;
}

@keyframes pulse {
0%, 100% {
    transform: scale(1);
    opacity: 1;
}
50% {
    transform: scale(1.1);
    opacity: 0.8;
}
}

h2 {
color: #00c6ff;
font-size: 28px;
margin: 0 0 20px 0;
text-shadow: 0 0 10px #00c6ff, 0 0 20px #0093ff;
}

p {
font-size: 15px;
line-height: 1.7;
color: #e0f7ff;
margin-bottom: 20px;
}

.reference {
font-size: 32px;
font-weight: bold;
color: #00c6ff;
margin: 30px 0;
padding: 25px;
background: rgba(0, 40, 80, 0.7);
border-radius: 12px;
border: 2px solid #00c6ff;
box-shadow: 0 0 20px rgba(0, 198, 255, 0.4);
letter-spacing: 3px;
font-family: 'Courier New', monospace;
text-shadow: 0 0 10px #00c6ff;
animation: glow 2s ease-in-out infinite;
}

@keyframes glow {
0%, 100% {
    box-shadow: 0 0 20px rgba(0, 198, 255, 0.4);
}
50% {
    box-shadow: 0 0 30px rgba(0, 198, 255, 0.6), 0 0 40px rgba(0, 198, 255, 0.4);
}
}

.warning-note {
background: rgba(255, 193, 7, 0.1);
border: 1px solid #ffc107;
border-left: 4px solid #ffc107;
padding: 15px;
border-radius: 8px;
margin: 25px 0;
font-size: 14px;
color: #ffd54f;
display: flex;
align-items: center;
justify-content: center;
}

```

```

    gap: 10px;
}

.warning-note::before {
  content: '⚠';
  font-size: 20px;
}

.button-container {
  display: flex;
  gap: 15px;
  justify-content: center;
  margin-top: 30px;
  flex-wrap: wrap;
}


  Report Abuse



  Report Abuse


Report Abuse


```

<h2>Your Report Has Been Submitted Anonymously</h2>

<p>Thank you for helping us fight harassment. Please save your Reference Code below to track the status of your report:</p>

```

<div class="reference">
  <%= request.getAttribute("referenceCode") %>
</div>

```

```

<div class="warning-note">
  <strong>Note:</strong> Without this code, you will not be able to track your report.
</div>

<div class="divider"></div>

<div class="button-container">
  <a href="index.html" class="secondary">← Go Back Home</a>
  <a href="track.jsp">Track Report →</a>
</div>
</div>
</body>
</html>
Track.jsp:
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
<title>Report Submitted</title>
<style>
body {
  margin: 0;
  font-family: 'Segoe UI', sans-serif;
  background: url("https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg") no-repeat center center fixed;
  background-size: cover;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  padding: 20px;
  color: #e0f7ff;
}
.container {
  background: rgba(0, 20, 40, 0.95);
  padding: 40px;
  border-radius: 15px;
  border: 2px solid #00c6ff;
  box-shadow: 0 0 30px rgba(0, 198, 255, 0.5), 0 0 50px rgba(0, 198, 255, 0.3);
  backdrop-filter: blur(10px);
  text-align: center;
  width: 100%;
  max-width: 550px;
  animation: fadeInScale 0.6s ease;
}
@keyframes fadeInScale {
  from {
    opacity: 0;
    transform: scale(0.9) translateY(30px);
  }
  to {
    opacity: 1;
    transform: scale(1) translateY(0);
  }
}
.success-icon {
  font-size: 80px;
  color: #00c6ff;
  margin-bottom: 20px;
  animation: pulse 2s infinite;
  text-shadow: 0 0 20px #00c6ff, 0 0 40px #00c6ff;
}
@keyframes pulse {
  0%, 100% {
    transform: scale(1);
    opacity: 1;
  }
}

```

```

50% {
  transform: scale(1.1);
  opacity: 0.8;
}
}


## 


```

```

display: inline-block;
padding: 14px 28px;
background: linear-gradient(135deg, #00c6ff 0%, #0093ff 100%);
color: #000;
text-decoration: none;
border-radius: 8px;
font-weight: bold;
font-size: 15px;
transition: all 0.3s ease;
box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);
border: none;
}
a:hover {
background: linear-gradient(135deg, #fff 0%, #00e5ff 100%);
transform: translateY(-3px);
box-shadow: 0 0 25px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.5);
}
a.secondary {
background: linear-gradient(135deg, rgba(0, 198, 255, 0.2) 0%, rgba(0, 147, 255, 0.2) 100%);
color: #00c6ff;
border: 2px solid #00c6ff;
}
a.secondary:hover {
background: linear-gradient(135deg, rgba(0, 198, 255, 0.4) 0%, rgba(0, 147, 255, 0.4) 100%);
color: #fff;
}
.divider {
height: 1px;
background: linear-gradient(90deg, transparent, #00c6ff, transparent);
margin: 25px 0;
}
</style>
</head>
<body>
<div class="container">
<div class="success-icon">✓</div>
<h2>Your Report Has Been Submitted Anonymously</h2>
<p>Thank you for helping us fight harassment. Please <strong>save your Reference Code</strong> below to track the status of your report:</p>
<div class="reference">
<%= request.getAttribute("referenceCode") %>
</div>

<div class="warning-note">
<strong>Note:</strong> Without this code, you will not be able to track your report.
</div>
<div class="divider"></div>
<div class="button-container">
<a href="index.html" class="secondary">← Go Back Home</a>
<a href="track.jsp">Track Report →</a>
</div>
</div>
</body>
</html>
AdminLoginServlet.java:
package report;

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/AdminLoginServlet")
public class AdminLoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

```

```

String username = request.getParameter("username");
String password = request.getParameter("password");
if ("admin".equals(username) && "admin123".equals(password)) {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/harassment_reporting", "root", "root");

        String action = request.getParameter("action");

        if ("assign".equals(action)) {
            String referenceCode = request.getParameter("referenceCode");
            String officerName = request.getParameter("officer");

            String sql = "UPDATE user_reports SET status=? WHERE referenceCode=?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, "Assigned to " + officerName);
            stmt.setString(2, referenceCode);
            stmt.executeUpdate();
            stmt.close();
        }
        request.setAttribute("conn", conn);
        RequestDispatcher rd = request.getRequestDispatcher("adminDashboard.jsp");
        rd.forward(request, response);

        conn.close();
    } catch (Exception e) {
        e.printStackTrace();
        response.getWriter().println("Error: " + e.getMessage());
    }
} else {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<meta charset='UTF-8'>");
    out.println("<meta name='viewport' content='width=device-width, initial-scale=1.0'>");
    out.println("<title>Login Failed</title>");
    out.println("<style>");
    out.println("body {");
    out.println(" margin: 0;");
    out.println(" font-family: 'Segoe UI', sans-serif;");
    out.println(" background: url('https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg') no-repeat center center fixed;");
    out.println(" background-size: cover;");
    out.println(" display: flex;");
    out.println(" justify-content: center;");
    out.println(" align-items: center;");
    out.println(" min-height: 100vh;");
    out.println(" padding: 20px;");
    out.println(" color: #e0f7ff;");
    out.println("}");
    out.println(".error-container {");
    out.println(" background: rgba(0, 20, 40, 0.95);");
    out.println(" padding: 40px;");
    out.println(" border-radius: 15px;");
    out.println(" border: 2px solid #f44336;");
    out.println(" box-shadow: 0 30px rgba(244, 67, 54, 0.5), 0 0 50px rgba(244, 67, 54, 0.3);");
    out.println(" backdrop-filter: blur(10px);");
    out.println(" text-align: center;");
    out.println(" width: 100%;");
    out.println(" max-width: 500px;");
    out.println(" animation: fadeInScale 0.6s ease;");
    out.println("}");
}

```

```

out.println("@keyframes fadeInScale {");
out.println(" from { opacity: 0; transform: scale(0.9) translateY(30px); }");
out.println(" to { opacity: 1; transform: scale(1) translateY(0); }");
out.println("}");
out.println(".error-icon {");
out.println(" font-size: 70px;");
out.println(" color: #f44336;");
out.println(" margin-bottom: 20px;");
out.println(" animation: shake 0.5s ease;");
out.println(" text-shadow: 0 0 20px #f44336;");
out.println("}");
out.println("@keyframes shake {");
out.println(" 0%, 100% { transform: translateX(0); }");
out.println(" 25% { transform: translateX(-10px); }");
out.println(" 75% { transform: translateX(10px); }");
out.println("}");
out.println("h3 {");
out.println(" color: #f44336;");
out.println(" font-size: 26px;");
out.println(" margin: 0 0 15px 0;");
out.println(" text-shadow: 0 0 10px rgba(244, 67, 54, 0.6);");
out.println("}");
out.println("p {");
out.println(" color: #e0f7ff;");
out.println(" font-size: 15px;");
out.println(" margin-bottom: 30px;");
out.println(" line-height: 1.6;");
out.println("}");
out.println(".divider {");
out.println(" height: 1px;");
out.println(" background: linear-gradient(90deg, transparent, #f44336, transparent);");
out.println(" margin: 25px 0;");
out.println("}");
out.println("a {");
out.println(" display: inline-block;");
out.println(" padding: 14px 30px;");
out.println(" background: linear-gradient(135deg, #00c6ff 0%, #0093ff 100%);");
out.println(" color: #000;");
out.println(" text-decoration: none;");
out.println(" border-radius: 8px;");
out.println(" font-weight: bold;");
out.println(" font-size: 15px;");
out.println(" transition: all 0.3s ease;");
out.println(" box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);");
out.println("}");
out.println("a:hover {");
out.println(" background: linear-gradient(135deg, #fff 0%, #00e5ff 100%);");
out.println(" transform: translateY(-3px);");
out.println(" box-shadow: 0 0 25px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.5);");
out.println("}");
out.println("</style>");
out.println("</head>");
out.println("<body>");
out.println("<div class='error-container'>");
out.println(" <div class='error-icon'>X</div>");
out.println(" <h3>Login Failed</h3>");
out.println(" <p>Invalid credentials. Please check your username and password and try again.</p>");
out.println(" <div class='divider'></div>");
out.println(" <a href='index.html'>← Go Back Home</a>");
out.println("</div>");
out.println("</body>");
out.println("</html>");

}

}

}

AssignOfficerServlet.java:
package report;

```

```

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;
import java.util.*;

@WebServlet("/AssignOfficerServlet")
public class AssignOfficerServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/harassment_reporting", "root", "root");

            Enumeration<String> paramNames = request.getParameterNames();

            boolean anyAssigned = false;

            while (paramNames.hasMoreElements()) {
                String paramName = paramNames.nextElement();

                if (paramName.startsWith("officer_")) {
                    String officer = request.getParameter(paramName);
                    if (officer != null && !officer.isEmpty()) {
                        String referenceCode = paramName.substring(8);

                        PreparedStatement stmt = conn.prepareStatement(
                            "UPDATE user_reports SET officer=? WHERE referenceCode=?"
                        );
                        stmt.setString(1, officer);
                        stmt.setString(2, referenceCode);
                        stmt.executeUpdate();
                        stmt.close();

                        anyAssigned = true;
                    }
                }
            }

            conn.close();
            if (anyAssigned) {
                request.setAttribute("confirmation", "Officer(s) assigned successfully!");
            } else {
                request.setAttribute("confirmation", "No officer was selected for assignment.");
            }
        }

        RequestDispatcher rd = request.getRequestDispatcher("adminDashboard.jsp");
        rd.forward(request, response);
    } catch (Exception e) {
        e.printStackTrace();
        response.getWriter().println("Error: " + e.getMessage());
    }
}

```

OfficerLoginServlet.java:

```

package report;

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

```

```

@WebServlet("/OfficerLoginServlet")
public class OfficerLoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String username = request.getParameter("username");
        String password = request.getParameter("password");
        if ((username.equals("officer1") && password.equals("pass1")) ||
            (username.equals("officer2") && password.equals("pass2")) ||
            (username.equals("officer3") && password.equals("pass3"))) {

            request.setAttribute("officerName", username);
            RequestDispatcher rd = request.getRequestDispatcher("officerDashboard.jsp");
            rd.forward(request, response);

        } else {
            response.setContentType("text/html;charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<meta charset='UTF-8'>");
            out.println("<meta name='viewport' content='width=device-width, initial-scale=1.0'>");
            out.println("<title>Login Failed</title>");
            out.println("<style>");
            out.println("body {");
            out.println("margin: 0;");
            out.println("font-family: 'Segoe UI', sans-serif;");
            out.println("background: url('https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg') no-repeat center center fixed;");
            out.println("background-size: cover;");
            out.println("display: flex;");
            out.println("justify-content: center;");
            out.println("align-items: center;");
            out.println("min-height: 100vh;");
            out.println("padding: 20px;");
            out.println("color: #e0f7ff;");
            out.println("}");
            out.println(".error-container {");
            out.println("background: rgba(0, 20, 40, 0.95);");
            out.println("padding: 40px;");
            out.println("border-radius: 15px;");
            out.println("border: 2px solid #f44336;");
            out.println("box-shadow: 0 0 30px rgba(244, 67, 54, 0.5), 0 0 50px rgba(244, 67, 54, 0.3);");
            out.println("backdrop-filter: blur(10px);");
            out.println("text-align: center;");
            out.println("width: 100%;");
            out.println("max-width: 500px;");
            out.println("animation: fadeInScale 0.6s ease;");
            out.println("}");
            out.println("@keyframes fadeInScale {");
            out.println("from { opacity: 0; transform: scale(0.9) translateY(30px); }");
            out.println("to { opacity: 1; transform: scale(1) translateY(0); }");
            out.println("}");
            out.println(".error-icon {");
            out.println("font-size: 70px;");
            out.println("color: #f44336;");
            out.println("margin-bottom: 20px;");
            out.println("animation: shake 0.5s ease;");
            out.println("text-shadow: 0 0 20px #f44336;");
            out.println("}");
            out.println("@keyframes shake {");
            out.println("0%, 100% { transform: translateX(0); }");
            out.println("25% { transform: translateX(-10px); }");
            out.println("75% { transform: translateX(10px); }");
            out.println("}");
        }
    }
}

```

```

        out.println("h3 {" );
        out.println(" color: #f44336;" );
        out.println(" font-size: 26px;" );
        out.println(" margin: 0 0 15px 0;" );
        out.println(" text-shadow: 0 0 10px rgba(244, 67, 54, 0.6);" );
        out.println("}" );
        out.println("p {" );
        out.println(" color: #e0f7ff;" );
        out.println(" font-size: 15px;" );
        out.println(" margin-bottom: 30px;" );
        out.println(" line-height: 1.6;" );
        out.println("}" );
        out.println(".divider {" );
        out.println(" height: 1px;" );
        out.println(" background: linear-gradient(90deg, transparent, #f44336, transparent);" );
        out.println(" margin: 25px 0;" );
        out.println("}" );
        out.println("a {" );
        out.println(" display: inline-block;" );
        out.println(" padding: 14px 30px;" );
        out.println(" background: linear-gradient(135deg, #00c6ff 0%, #0093ff 100%);" );
        out.println(" color: #000;" );
        out.println(" text-decoration: none;" );
        out.println(" border-radius: 8px;" );
        out.println(" font-weight: bold;" );
        out.println(" font-size: 15px;" );
        out.println(" transition: all 0.3s ease;" );
        out.println(" box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);" );
        out.println("}" );
        out.println("a:hover {" );
        out.println(" background: linear-gradient(135deg, #fff 0%, #00e5ff 100%);" );
        out.println(" transform: translateY(-3px);" );
        out.println(" box-shadow: 0 0 25px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.5);" );
        out.println("}" );
        out.println("</style>" );
        out.println("</head>" );
        out.println("<body>" );
        out.println("<div class='error-container'>" );
        out.println(" <div class='error-icon'>X</div>" );
        out.println(" <h3>Login Failed</h3>" );
        out.println(" <p>Invalid credentials. Please check your username and password and try again.</p>" );
        out.println(" <div class='divider'></div>" );
        out.println(" <a href='index.html'>← Go Back Home</a>" );
        out.println("</div>" );
        out.println("</body>" );
        out.println("</html>" );
    }
}
}

SubmitReportServlet.java:

```

```

package report;

import java.io.*;
import java.sql.*;
import java.util.UUID;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;
import java.nio.file.Files;
import java.nio.file.StandardCopyOption;

```

```

@WebServlet("/SubmitReportServlet")
@MultipartConfig(
    fileSizeThreshold = 1024 * 1024 * 1,
    maxFileSize = 1024 * 1024 * 10,
    maxRequestSize = 1024 * 1024 * 15
)

```

```

public class SubmitReportServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String age = request.getParameter("age");
        String gender = request.getParameter("gender");
        String description = request.getParameter("details");
        String dateOfIncident = request.getParameter("incident_date");
        String place = request.getParameter("location");
        Part filePart = request.getPart("proofFile");
        String filePath = null;
        if (filePart != null && filePart.getSize() > 0) {
            String uploadDir = getServletContext().getRealPath("") + File.separator + "uploads";
            File uploadFolder = new File(uploadDir);
            if (!uploadFolder.exists()) uploadFolder.mkdirs();
            String fileName = UUID.randomUUID().toString() + "_" + filePart.getSubmittedFileName();
            filePath = uploadDir + File.separator + fileName;
            try (InputStream input = filePart.getInputStream()) {
                Files.copy(input, new File(filePath).toPath(), StandardCopyOption.REPLACE_EXISTING);
            }
        }
        String referenceCode = "REF-" + UUID.randomUUID().toString().substring(0, 6).toUpperCase();
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/harassment_reporting", "root", "root");

            String sql = "INSERT INTO user_reports " +
                "(referenceCode, age, gender, description, date_of_incident, place, filePath, status) " +
                "VALUES (?, ?, ?, ?, ?, ?, ?, 'Submitted')";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, referenceCode);
            stmt.setInt(2, Integer.parseInt(age));
            stmt.setString(3, gender);
            stmt.setString(4, description);
            stmt.setDate(5, java.sql.Date.valueOf(dateOfIncident));
            stmt.setString(6, place);
            stmt.setString(7, filePath);
            stmt.executeUpdate();
            stmt.close();
            conn.close();
            request.setAttribute("referenceCode", referenceCode);
            RequestDispatcher rd = request.getRequestDispatcher("success.jsp");
            rd.forward(request, response);
        } catch (Exception e) {
            e.printStackTrace();
            response.getWriter().println("Error: " + e.getMessage());
        }
    }
}

```

TrackReportServlet.java:

```

package report;

import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/TrackReportServlet")

public class TrackReportServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String referenceCode = request.getParameter("referenceCode");

```

```

try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/harassment_reporting", "root",
"root");

    String sql = "SELECT status FROM user_reports WHERE referenceCode=?";
    PreparedStatement stmt = conn.prepareStatement(sql);
    stmt.setString(1, referenceCode);

    ResultSet rs = stmt.executeQuery();
    if (rs.next()) {
        request.setAttribute("status", rs.getString("status"));
        request.setAttribute("referenceCode", referenceCode);
        RequestDispatcher rd = request.getRequestDispatcher("track.jsp");
        rd.forward(request, response);
    } else {
        response.getWriter().println("Invalid Reference Code");
    }

    stmt.close();
    conn.close();
} catch (Exception e) {
    e.printStackTrace();
    response.getWriter().println("Error: " + e.getMessage());
}
}

}

UpdateReportStatusServlet.java:
package report;

import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/UpdateReportStatusServlet")
public class UpdateReportStatusServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        int updatedCount = 0;
        List<String> updatedReports = new ArrayList<>();

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection conn = DriverManager.getConnection(
                "jdbc:mysql://localhost:3306/harassment_reporting", "root", "root");
            Enumeration<String> paramNames = request.getParameterNames();
            while (paramNames.hasMoreElements()) {
                String paramName = paramNames.nextElement();
                if (paramName.startsWith("status_")) {
                    String referenceCode = paramName.substring(7);
                    String newStatus = request.getParameter(paramName);
                    String sql = "UPDATE user_reports SET status=? WHERE referenceCode=?";
                    PreparedStatement stmt = conn.prepareStatement(sql);
                    stmt.setString(1, newStatus);
                    stmt.setString(2, referenceCode);
                    int rowsUpdated = stmt.executeUpdate();

                    if (rowsUpdated > 0) {
                        updatedCount++;
                        updatedReports.add(referenceCode);
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        }
        stmt.close();
    }
}
conn.close();
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<meta charset='UTF-8'>");
out.println("<meta name='viewport' content='width=device-width, initial-scale=1.0'>");
out.println("<title>Status Updated</title>");
out.println("<style>");
out.println("body {");
out.println(" margin: 0;");
out.println(" font-family: 'Segoe UI', sans-serif;");
out.println(" background: url('https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg') no-repeat center center fixed;");
out.println(" background-size: cover;");
out.println(" display: flex;");
out.println(" justify-content: center;");
out.println(" align-items: center;");
out.println(" min-height: 100vh;");
out.println(" padding: 20px;");
out.println(" color: #e0f7ff;");
out.println("}");
out.println(".");
success-container {
background: rgba(0, 20, 40, 0.95);
padding: 40px;
border-radius: 15px;
border: 2px solid #00c6ff;
box-shadow: 0 0 30px rgba(0, 198, 255, 0.5), 0 0 50px rgba(0, 198, 255, 0.3);
backdrop-filter: blur(10px);
text-align: center;
width: 100%;
max-width: 600px;
animation: fadeInScale 0.6s ease;
}
.out.println("}");


@keyframes fadeInScale {
from { opacity: 0; transform: scale(0.9) translateY(30px); }
to { opacity: 1; transform: scale(1) translateY(0); }
}
.success-icon {
font-size: 70px;
color: #00c6ff;
margin-bottom: 20px;
animation: pulse 2s infinite;
text-shadow: 0 0 20px #00c6ff;
}
.out.println("}");


@keyframes pulse {
0%, 100% { transform: scale(1); opacity: 1; }
50% { transform: scale(1.1); opacity: 0.8; }
}
.h2 {
color: #00c6ff;
font-size: 28px;
margin: 0 0 15px 0;
text-shadow: 0 0 10px #00c6ff, 0 0 20px #0093ff;
}
.out.println("}");


p {
color: #e0f7ff;
font-size: 16px;
margin-bottom: 25px;
line-height: 1.6;
}
.divider {
height: 1px;
}

```

```

out.println(" background: linear-gradient(90deg, transparent, #00c6ff, transparent');");
out.println(" margin: 25px 0;");
out.println("}");
out.println("button {");
out.println(" padding: 14px 30px;");
out.println(" font-size: 16px;");
out.println(" cursor: pointer;");
out.println(" background: linear-gradient(135deg, #00c6ff 0%, #0093ff 100%);">
out.println(" border: none;");
out.println(" color: #000;");
out.println(" font-weight: bold;");
out.println(" border-radius: 8px;");
out.println(" transition: all 0.3s ease;");
out.println(" box-shadow: 0 0 15px rgba(0, 198, 255, 0.3);");
out.println("}");
out.println("button:hover {");
out.println(" background: linear-gradient(135deg, #ffff 0%, #00e5ff 100%);">
out.println(" transform: translateY(-3px);");
out.println(" box-shadow: 0 0 25px #00c6ff, 0 0 40px rgba(0, 198, 255, 0.5);");
out.println("}");
out.println(".count-badge {");
out.println(" display: inline-block;");
out.println(" background: rgba(0, 198, 255, 0.2);");
out.println(" border: 1px solid #00c6ff;");
out.println(" padding: 8px 20px;");
out.println(" border-radius: 20px;");
out.println(" font-size: 18px;");
out.println(" font-weight: bold;");
out.println(" margin: 15px 0;");
out.println(" color: #00c6ff;");
out.println("}");
out.println("</style>");
out.println("</head>");
out.println("<body>");
out.println("<div class='success-container'>");
out.println(" <div class='success-icon'>✓</div>");
out.println(" <h2>Status Updated Successfully</h2>");
out.println(" <div class='count-badge'>" + updatedCount + " report(s) updated</div>");
out.println(" <p>The status has been successfully updated for the selected reports.</p>");
out.println(" <div class='divider'></div>");
out.println(" <button onclick=\"window.location.href='index.html'\">← Go Back Home</button>");
out.println("</div>");
out.println("</body>");
out.println("</html>");

} catch (Exception e) {
    e.printStackTrace();
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<meta charset='UTF-8'>");
    out.println("<meta name='viewport' content='width=device-width, initial-scale=1.0'>");
    out.println("<title>Update Failed</title>");
    out.println("<style>");
    out.println("body {");
    out.println(" margin: 0;");
    out.println(" font-family: 'Segoe UI', sans-serif;");
    out.println(" background: url('https://image.freepik.com/vecteurs-libre/concept-cybersecurite-element-hud-fond-ton-bleu_99087-130.jpg') no-repeat center center fixed;");
    out.println(" background-size: cover;");
    out.println(" display: flex;");
    out.println(" justify-content: center;");
    out.println(" align-items: center;");
    out.println(" min-height: 100vh;");
    out.println(" padding: 20px;");
    out.println(" color: #e0f7ff;");
    out.println("}");
}

```

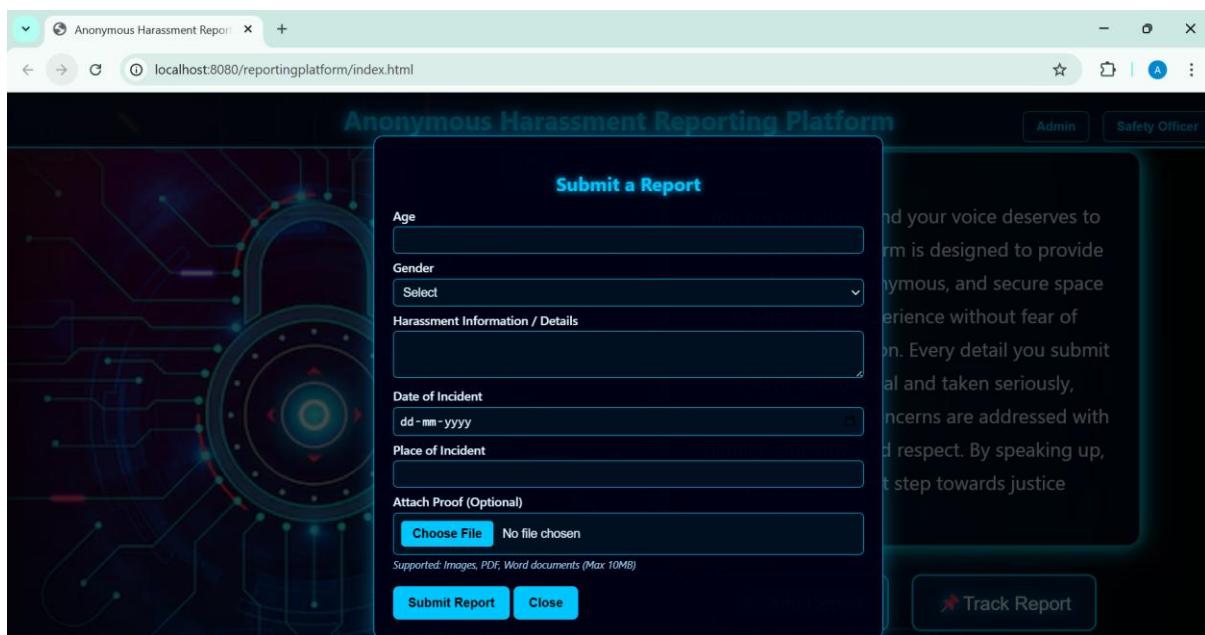
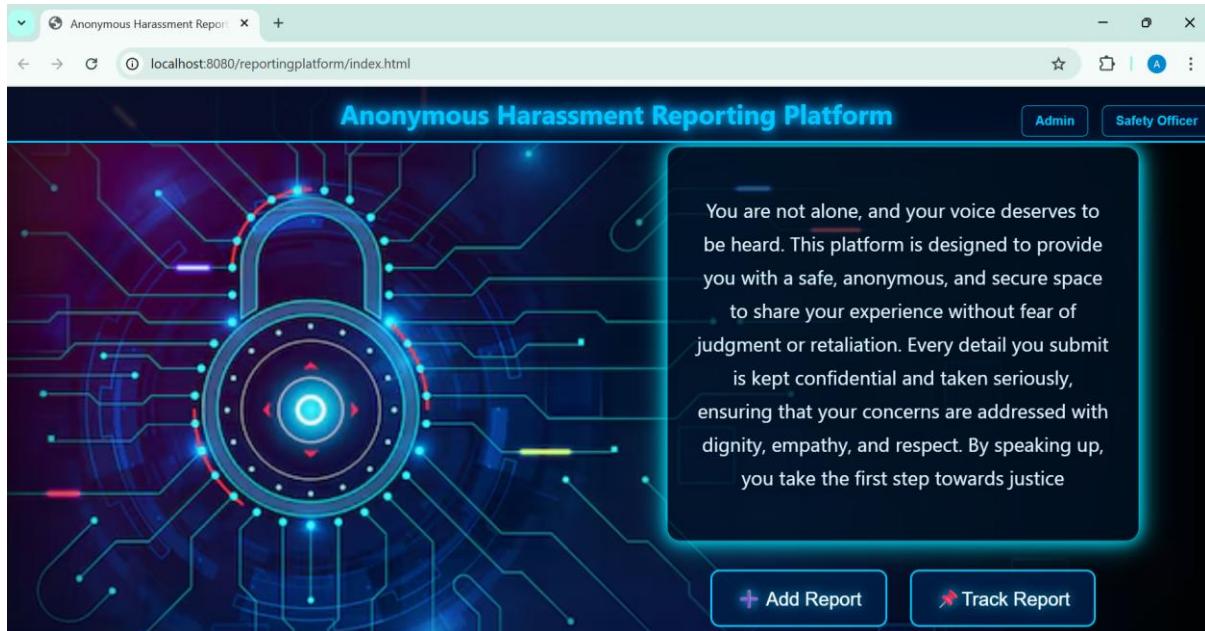
```

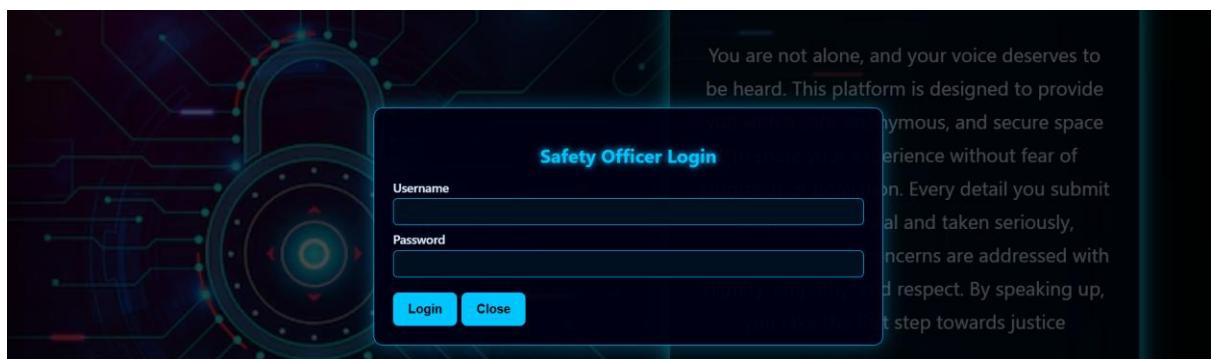
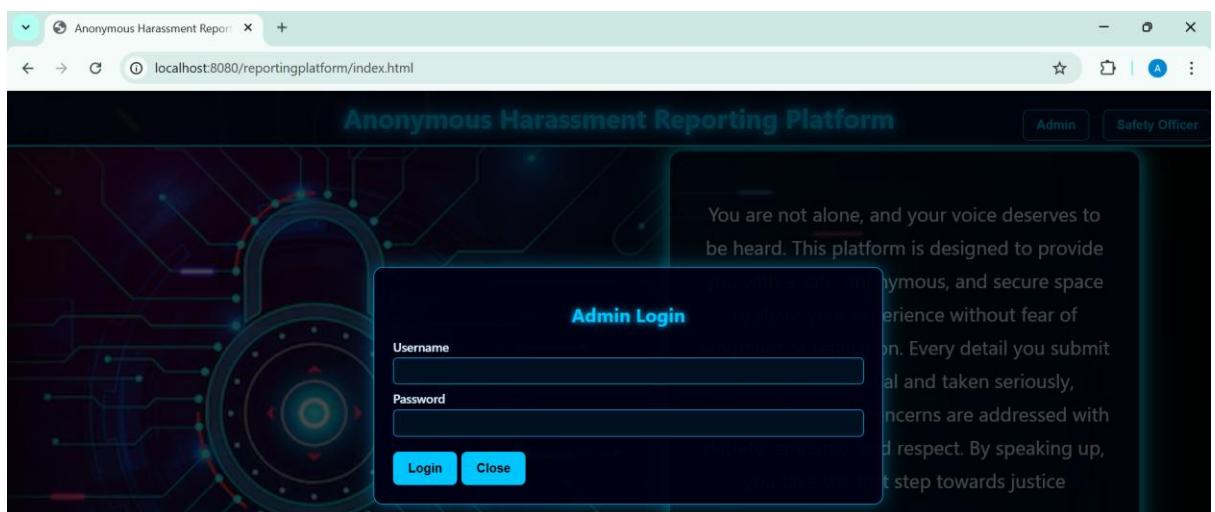
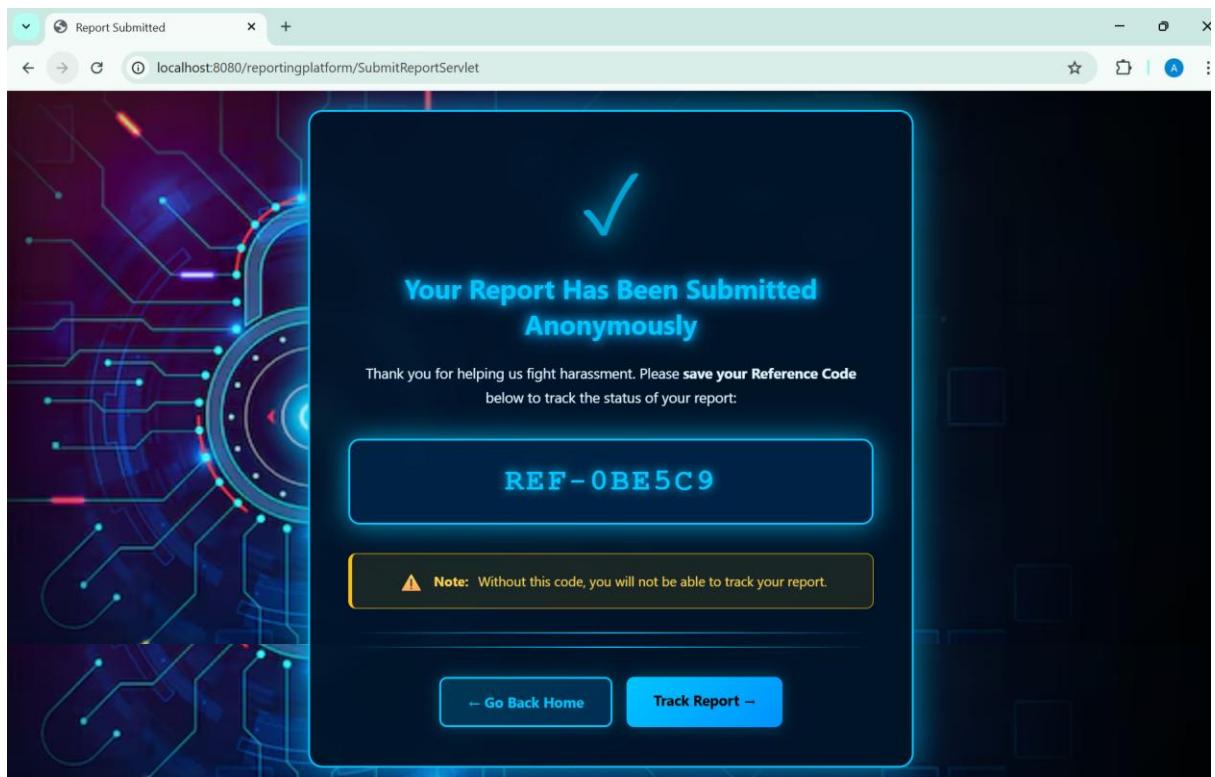
        out.println(".error-container {");
        out.println(" background: rgba(0, 20, 40, 0.95);");
        out.println(" padding: 40px;");
        out.println(" border-radius: 15px;");
        out.println(" border: 2px solid #f44336;");
        out.println(" box-shadow: 0 0 30px rgba(244, 67, 54, 0.5);");
        out.println(" backdrop-filter: blur(10px);");
        out.println(" text-align: center;");
        out.println(" max-width: 500px;");
        out.println(" animation: fadeInScale 0.6s ease;\"");
        out.println("}");
        out.println("@keyframes fadeInScale {");
        out.println(" from { opacity: 0; transform: scale(0.9) translateY(30px); }\"");
        out.println(" to { opacity: 1; transform: scale(1) translateY(0); }\"");
        out.println("}");
        out.println(".error-icon {");
        out.println(" font-size: 70px;");
        out.println(" color: #f44336;\"");
        out.println(" margin-bottom: 20px;\"");
        out.println(" animation: shake 0.5s ease;\"");
        out.println("}");
        out.println("@keyframes shake {");
        out.println(" 0%, 100% { transform: translateX(0); }\"");
        out.println(" 25% { transform: translateX(-10px); }\"");
        out.println(" 75% { transform: translateX(10px); }\"");
        out.println("}");
        out.println("h2 { color: #f44336; font-size: 26px; margin: 0 0 15px 0; text-shadow: 0 0 10px rgba(244, 67, 54, 0.6); }\"");
        out.println("p { color: #e0f7ff; font-size: 15px; margin-bottom: 30px; }\"");
        out.println("button {");
        out.println(" padding: 14px 30px;\"");
        out.println(" font-size: 16px;\"");
        out.println(" cursor: pointer;\"");
        out.println(" background: linear-gradient(135deg, #00c6ff 0%, #0093ff 100%);\"");
        out.println(" border: none;\"");
        out.println(" color: #000;\"");
        out.println(" font-weight: bold;\"");
        out.println(" border-radius: 8px;\"");
        out.println(" transition: all 0.3s ease;\"");
        out.println("}");
        out.println("button:hover {");
        out.println(" background: linear-gradient(135deg, #fff 0%, #00e5ff 100%);\"");
        out.println(" transform: translateY(-3px);\"");
        out.println("}");
        out.println("</style>\"");
        out.println("</head>\"");
        out.println("<body>\"");
        out.println("<div class='error-container'>\"");
        out.println(" <div class='error-icon'>X</div>\"");
        out.println(" <h2>Update Failed</h2>\"");
        out.println(" <p>Error: " + e.getMessage() + "</p>\"");
        out.println(" <button onclick=\"window.location.href='index.html'\">← Go Back Home</button>\"");
        out.println("</div>\"");
        out.println("</body>\"");
        out.println("</html>\"");
    }
}

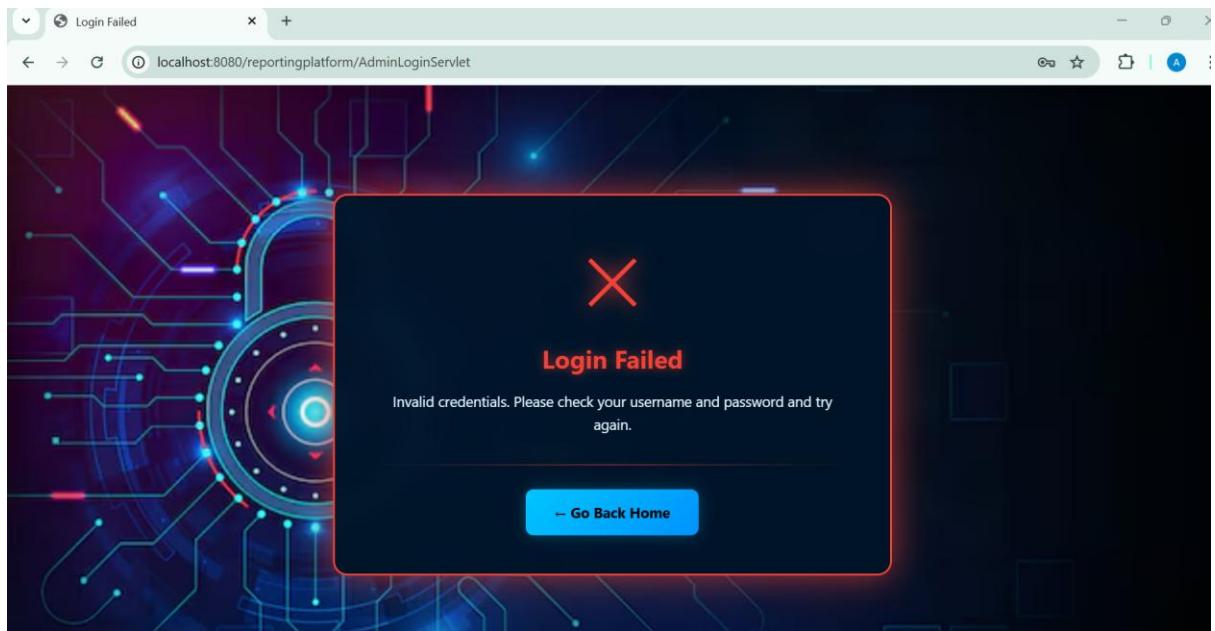
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.sendRedirect("index.html");
}
}

```

DEMONSTRATION:







The screenshot shows a web browser window with the title "Admin Dashboard". The URL in the address bar is "localhost:8080/reportingplatform/AdminLoginServlet". The main content is titled "Admin Dashboard" in large blue letters. Below it is a table with the following data:

| REF CODE | AGE | GENDER | DESCRIPTION | DATE | PLACE | STATUS | ASSIGN OFFICER |
|-------------|-----|--------|------------------|------------|------------|-------------|------------------|
| REF-OBEC5C9 | 16 | Female | In my school.... | 2022-10-27 | Salem | Submitted | Officer2 ▾ |
| REF-C6640D | 17 | Other | In my area | 2020-07-14 | Trichy | Submitted | Officer3 ▾ |
| REF-0DA7F4 | 17 | Female | that happened | 2025-01-01 | Chennai | In Progress | Officer1 ▾ |
| REF-6C623E | 19 | Male | this happened | 2015-11-12 | Coimbatore | In Progress | Officer2 ▾ |
| REF-EFA544 | 20 | Female | so and so | 2018-12-12 | Cuddalore | Resolved | Select Officer ▾ |
| REF-BCC08A | 20 | Female | so and so | 2012-12-12 | Cuddalore | Resolved | Officer2 ▾ |

At the bottom left is a blue button labeled "Go Back Home". At the bottom right is a blue button labeled "Assign Officers".

Admin Dashboard

Success!
Officer(s) assigned successfully!

| REF CODE | AGE | GENDER | DESCRIPTION | DATE | PLACE | STATUS | ASSIGN OFFICER |
|------------|-----|--------|------------------|------------|--------|-----------|----------------|
| REF-0BE5C9 | 16 | Female | In my school.... | 2022-10-27 | Salem | Submitted | Officer2 |
| REF-C6640D | 17 | Other | In my area | 2020-07-14 | Trichy | Submitted | Officer3 |

Welcome officer2

Assigned Reports

| REF CODE | AGE | GENDER | DESCRIPTION | DATE | PLACE | STATUS | UPDATE STATUS |
|------------|-----|--------|------------------|------------|------------|-------------|---------------|
| REF-0BE5C9 | 16 | Female | In my school.... | 2022-10-27 | Salem | SUBMITTED | Submitted |
| REF-6C623E | 19 | Male | this happened | 2015-11-12 | Coimbatore | IN PROGRESS | In Progress |
| REF-BCC0BA | 20 | Female | so and so | 2012-12-12 | Cuddalore | RESOLVED | Resolved |

[Go Back](#) [Update](#)

Anonymous Harassment Reporting Platform

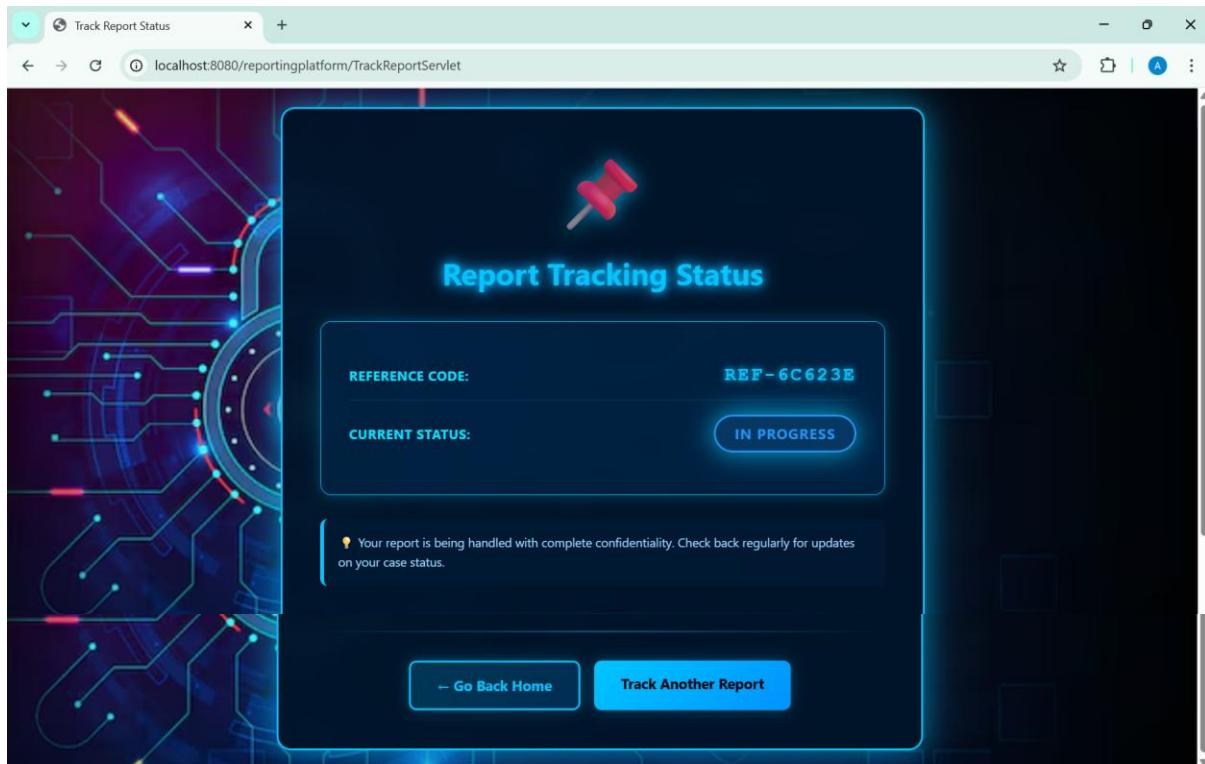
[Admin](#) [Safety Officer](#)

You are not alone, and your voice deserves to be heard. This platform is designed to provide you with a safe, anonymous, and secure space to share your experience without fear of judgment or retaliation. Every detail you submit will be handled with care and taken seriously, and your concerns are addressed with dignity, empathy, and respect. By speaking up, you take the first step towards justice.

Track Your Report

Reference Code

[Check Status](#) [Close](#)



```
mysql> use harassment_reporting;
Database changed
mysql> describe user_reports;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int | NO | PRI | NULL | auto_increment |
| referenceCode | varchar(100) | NO | | NULL |
| age | int | YES | | NULL |
| gender | varchar(20) | YES | | NULL |
| description | text | NO | | NULL |
| date_of_incident | date | NO | | NULL |
| place | varchar(150) | NO | | NULL |
| filePath | varchar(255) | YES | | NULL |
| status | varchar(50) | YES | | Submitted |
| created_at | timestamp | YES | | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| officer | varchar(20) | YES | | NULL |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```