BIRZEIT UNIVERSITY

COMPUTER SCIENCE DEPARTMENT

COMP4382 (Second 2024/2025)

Human Rights Monitor - Final Project Guidelines

Project Overview

The **Human Rights Monitor Management Information System (MIS)** is a secure, data-driven platform designed to **document, track, and analyze human rights violations**. It enables organizations to:

Record cases of abuses (arbitrary detention, forced displacement, etc.)
Manage victim/witness data securely (with anonymity options)
Process incident reports from multiple sources (web, mobile, NGOs)
Generate visual analytics (trends, hotspots, violation types)
S**upport legal actions** with structured evidence and documentation

Key Features

Case Management – Track investigations from report to resolution
Incident Reporting – Secure submission with media attachments
Victim Database – Protected records with risk assessment
Data Dashboard – Interactive charts, maps, and exportable reports

Goal: Improve documentation efficiency, evidence preservation, and advocacy through data-driven insights.

Students will develop a **Human Rights Monitor Management Information System (MIS)** using **FastAPI (Python) + MongoDB**, with a frontend of their choice (React, Vue.js, or Streamlit for Python-based dashboards).

The system will consist of **four modules**, and each of the three students will select **one module** to develop (the last module will be implemented by the group).

Project Modules & Responsibilities

1 Case Management System (Student 1)

Responsibilities:

- Design and implement CRUD operations for human rights cases.
- Add search/filter functionality (by date, location, violation type).
- Ensure proper case status tracking (e.g., `new`, `under_investigation`, `resolved`).
- Implement file/document attachments (PDFs, images, videos).

Key API Endpoints:

- `POST /cases/` – Create a new case
- `GET /cases/{case_id}` – Retrieve a specific case
- `GET /cases/` – List all cases (with filters)
- `PATCH /cases/{case_id}` – Update case status
- `DELETE /cases/{case_id}` – Archive a case

MongoDB Collections:

- `cases` (main case records)
- `case_status_history` (track status changes)

2 Incident Reporting Module (Student 2)

Responsibilities:

- Develop a secure form for submitting human rights violations.
- Support anonymous reporting (optional personal details).
- Allow media uploads (photos, videos, documents).
- Implement geolocation tagging (Google Maps API or OpenStreetMap).

Key API Endpoints:

- `POST /reports/` – Submit a new incident report
- `GET /reports/` – List reports (filter by status, date, location)
- `PATCH /reports/{report_id}` – Update report status
- `GET /reports/analytics` – Count reports by violation type

MongoDB Collections:

- `incident_reports` (reported incidents)
- `report_evidence` (attached media)

3 Victim/Witness Database Module (Student 3)

Responsibilities:

- Design a secure database for victims/witnesses.
- Implement role-based access (only authorized users can view sensitive data).
- Add risk assessment tracking (low/medium/high risk).
- Support pseudonyms for anonymity.

Key API Endpoints:

- `POST /victims/` – Add a new victim/witness
- `GET /victims/{victim_id}` – Retrieve victim details (restricted access)
- `PATCH /victims/{victim_id}` – Update risk level
- `GET /victims/case/{case_id}` – List victims linked to a case

MongoDB Collections:

- `victims` (personal details, encrypted if sensitive)
- `victim_risk_assessments` (track risk levels)

4 Data Analysis & Visualization (Student 1 or 2)

Responsibilities:

- Build dashboards showing trends (violation types, locations, time periods).
- Generate PDF/Excel reports.
- Implement charts (bar, pie, maps) using **Plotly, Matplotlib, or D3.js**.
- Add filters (date range, region, violation type).

Key API Endpoints:

- `GET /analytics/violations` – Count violations by type
- `GET /analytics/geodata` – Map visualization data
- `GET /analytics/timeline` – Cases/reports over time

MongoDB Aggregations Needed:

- Group by violation type, location, date
- Time-series analysis

Project Deliverables

Working FastAPI backend (with MongoDB integration)
Frontend (React/Vue/Streamlit)
GitHub repository (with proper branching & commits)
Postman/OpenAPI documentation (API endpoints)
Final presentation (demo + code walkthrough)

Evaluation criteria:

1. Coverage of the above requirements.
2. Implementing the best practices and constraints.
3. Overall design and understanding.

Submission:

1. Due date: June 10, 2025, 00:00. You have to reply to the assignment message on Ritaj with your GitHub repository link only. Then download the project from GitHub (.zip format) and upload it on ITC.
2. No late submissions or extensions will be accepted.
3. You might be asked to design and apply code changes during the discussion.

Sample MongoDB Schema Design & Sample Data for Human Rights MIS (FastAPI Backend)

1. Case Management Collection

Schema: `cases`


```
{
  "_id": {"$oid": "507f1f77bcf86cd799439011"},
  "case_id": "HRM-2023-0425",
  "title": "Forced Displacement in Northern Region",
  "description": "Mass displacement of villagers following military
operations",
  "violation_types": ["forced_displacement", "property_destruction"],
  "status": "under_investigation",
  "priority": "high",
  "location": {
    "country": "Syria",
    "region": "Northern Governorate",
    "coordinates": {"type": "Point", "coordinates": [36.5078, 37.0954]}
  },
  "date_occurred": {"$date": "2023-04-15T00:00:00Z"},
  "date_reported": {"$date": "2023-04-20T00:00:00Z"},
  "victims": [{"$oid": "507f1f77bcf86cd799439012"}, {"$oid":
"507f1f77bcf86cd799439013"}],
  "perpetrators": [{"name": "35th Infantry Division", "type":
"military_unit"}],
  "evidence": [
    {
      "type": "photo",
      "url": "/evidence/hr0425-1.jpg",
      "description": "Destroyed homes in village",
```

```
      "date_captured": {"$date": "2023-04-16T00:00:00Z"}
    }
  ],
  "created_by": {"$oid": "507f1f77bcf86cd799439021"},
  "created_at": {"$date": "2023-04-20T14:30:00Z"},
  "updated_at": {"$date": "2023-04-25T09:15:00Z"}
}
```

2. Incident Reports Collection

Schema: `incident_reports`

```
{
  "_id": {"$oid": "607f1f77bcf86cd799439014"},
  "report_id": "IR-2023-0789",
  "reporter_type": "victim",
  "anonymous": false,
  "contact_info": {
    "email": "reporter@example.com",
    "phone": "+963912345678",
    "preferred_contact": "email"
  },
  "incident_details": {
    "date": {"$date": "2023-05-10T00:00:00Z"},
    "location": {
      "country": "Yemen",
      "city": "Taiz",
      "coordinates": {"type": "Point", "coordinates": [44.0333,
13.5833]}
    },
    "description": "Arbitrary detention of 15 civilians at checkpoint",
    "violation_types": ["arbitrary_detention", "torture"]
  },
  "evidence": [
    {
      "type": "video",
      "url": "/evidence/ir0789-1.mp4",
      "description": "Checkpoint footage"
    }
  ],
  "status": "new",
  "assigned_to": {"$oid": "507f1f77bcf86cd799439022"},
  "created_at": {"$date": "2023-05-12T08:45:00Z"}
}
```

3. Victims/Witnesses Collection

Schema: `individuals`

```
{
  "_id": {"$oid": "507f1f77bcf86cd799439012"},
  "type": "victim",
  "anonymous": false,
  "demographics": {
    "gender": "female",
    "age": 34,
    "ethnicity": "Kurdish",
    "occupation": "teacher"
  },
  "contact_info": {
    "email": "safecontact@example.org",
    "phone": "+963987654321",
    "secure_messaging": "signal"
  },
  "cases_involved": [{"$oid": "507f1f77bcf86cd799439011"}],
  "risk_assessment": {
    "level": "medium",
    "threats": ["intimidation", "surveillance"],
    "protection_needed": true
  },
  "support_services": [
    {
      "type": "legal",
      "provider": "HRM Legal Team",
      "status": "active"
    }
  ],
  "created_at": {"$date": "2023-04-21T10:00:00Z"},
  "updated_at": {"$date": "2023-05-15T14:20:00Z"}
}
```