

Project Endpoints Summary

1) submit report by anyone:

-endpoint used : @router.post("/with-file") which is in routes/reports.py

we connected it in report.js like this:

```
const response = await fetch("http://127.0.0.1:8000/reports/with-file", {  
  method: "POST",  
  body: formData, // which is the form that the user submitted  
});
```

اعبط

=

مہان بج ف، یل دد یرف رزوی شمو یقطنم شم هأل ضرعو لوط طوطخ لخدی رزوی لا نم ان بل ط ام
نم:

OpenStreetMap API by city and country name.

2) there is no signup because it is sensitive data, but there is login and 2 roles: admin and investigator.

the admin can add users (investigators or admins).

-endpoint used for login: @router.post("/login") which is in routes/auth.py

in login.js:

```
const res = await fetch(`${baseUrl}/login`, {  
  method: "POST",  
  headers: { "Content-Type": "application/json" },  
  body: JSON.stringify({ email, password })  
});
```

it checks if the email exists in the DB, compares the password using verify_password in encryption_utils.py (bcrypt).

هرو د بسح دروبش ادلا یلع مدختسم لا لخدی، ؤحیحص تانای بل تاناک اذا

تایحالصل لا نم ققحتب دنإ کابل او، مدختسم لا تانای ب نینختل همدختسن ب localStorage لا
دروسابل او role_checker.py م ادختساب

Note on Security:

دروسابل او، check_role، لالخ نم ؤیحم ؤساسحل ات وارلا لک

Project Endpoints Summary

فشم

س

bcrypt. م ادخ تس اب ر

3) analytics (for admin and investigator):

Endpoints used in analytics.py:

- @router.get("/by-location"): returns number of reports per city using MongoDB aggregation.
- @router.get("/by-date"): returns number of reports per month.
- @router.get("/geodata"): returns lat/lon for each report.
- @router.get("/summary"): summary of reports by type, city, and date.

Displayed using Chart.js and Leaflet.js in JS.

4) manage cases (admin and investigator):

- POST /cases/: Add a new case.
- GET /cases/: Retrieve all cases.
- GET /cases/{case_id}: View case details.
- PATCH /cases/{case_id}: Update status and log to case_status_history_collection.
- DELETE /cases/{case_id}: Archive case (not delete forever).

5) add user (admin only):

- POST /add-user in auth.py

JS sends: JSON.stringify({ email, password, role, requester_email })

Checks requester is admin, ensures email isn't duplicated, hashes password using bcrypt, and inserts user.

6) manage victims (admin only):

- POST /victims/: Add new victim.
- GET /victims/: Retrieve all victims.
- GET /victims/{id}: Get victim data.
- PUT /victims/{id}: Update victim data.
- DELETE /victims/{id}: Permanent delete.
- PATCH /victims/{id}/risk-level: Update risk level.

Project Endpoints Summary

- GET /victims/{id}/risk-history: Get history of risk level (stored in risk_history_collection).
- GET /victims/{id}/risk-latest: Latest risk level.
- GET /victims/{id}/cases: Victim-related cases.
- GET /victims/case/{case_id}: Victims related to a case.