

# Programming Project / C++ – Incident Reporting System

---

Student 1: Andreea Ungur

Student 2: Ilinca Stanescu

## I. Task Description

Student 1 is responsible for managing the incident reporting functionality, which includes:

- Adding new incident reports with validated input for area, type, and date
- Viewing a list of all recorded incidents in a structured tabular format
- Saving new incidents to a persistent file storage named "incidents.txt"

Student 2 is responsible for providing interactive tools to manage and analyze the stored incidents:

- Filtering incidents by area, type, or date using case-insensitive substring search
- Editing specific fields (area, type, date) of selected incidents with validation
- Deleting unwanted or obsolete incident entries from the system
- Sorting the incident list by area (alphabetically) or by date (chronologically)

## II. Data Structures Used by the Team

The system leverages two main C++ classes:

### 1. Incident

- Attributes:

- `std::string area`
- `std::string type`
- `std::string date`

- Methods:

- Getters and setters for each attribute
- A static function to validate the date format and logical consistency

### 2. IncidentManager

- Attributes:

- `std::vector<Incident> incidents`
- `std::string dataFile = "incidents.txt"`

- Methods:
  - loadFromFile, saveToFile for data persistence
  - addIncident, viewIncidents for data entry and display
  - editIncident, deleteIncident for modification and removal
  - filterIncidents, sortIncidents for searching and ordering data

### III. File Structure

- incidents.txt

A CSV file used to store the incident records persistently, with one incident per line in the format:

<area>,<type>,<date>

Example:

Timisoara,Power Failure,01.05.2025

Bucharest,Fire Alert,02.06.2025

### IV. Interacting with Executables

Application 1 will offer the following options:

- ./main1.exe

To launch the incident reporting system interface for adding and viewing incidents.

Simulated Terminal Output:

=== Incident Reporting - Main 1 ===

1. Add Incident
2. View Incidents
3. Exit

Choose an option: \_

Application 2 will offer the following options:

- ./main2.exe

To launch the incident management interface for filtering, editing, deleting, and sorting incidents.

Simulated Terminal Output:

=== Incident Reporting - Main 2 ===

1. Filter Incidents
2. Edit Incident
3. Delete Incident
4. Sort Incidents
5. Exit

Choose an option: \_