CS-359 Project Report

E-199

(Handling incidents by firefighting services)

Nikoleta Arvaniti (csd4844)

28 January 2025

1. Introduction

The project is divided in two main parts: webapp file that contains the front end (HTML, CSS, JavaScript) and the java file that contains the back end (Java Servlets, REST APIs, AJAX).

No changes were made in the DataBase, the tables are:

- incidents
- messages
- participants
- users
- volunteers

(The project runs on a noCORS browser)

2.Back End

1. User Management Servlets

- **LoadUser.java**: Fetches user details (e.g., registered users) from the database.
- **RegistrationUser.java**: Handles new user registration and adds them to the database.
- **UpdateUser.java**: Updates existing user details (e.g., profile information).
- **Login.java**: Authenticates users using their credentials.
- **Logout.java**: Manages user session termination.

2. Volunteer Management Servlets

- **LoadVolunteer.java**: Retrieves details of registered volunteers.
- **RegistrationVolunteer.java**: Handles volunteer registration.
- **UpdateVolunteer.java**: Updates volunteer profile information.
- **AcceptIncidentParticipation.java**: Allows volunteers to accept participation in an incident.

3. Incident Management Servlets

- **LoadIncidents.java**: Fetches a list of all incidents from the database.
- **LoadIncidentsUser.java**: Retrieves incidents related to a specific user.
- **LoadRunningIncidents.java**: Lists ongoing active incidents.
- **AddIncident.java**: Handles new incident submissions.
- **UpdateIncident.java**: Updates information about a specific incident.

- **GetIncidentDetails.java**: Fetches detailed information for a specific incident.

4. Communication Servlets

- **SendPublicMessage.java**: Sends public messages related to incidents.
- **SendUserMessage.java**: Sends private messages from users to administrators.
- **SendVolunteerMessage.java**: Sends private messages from volunteers to administrators or other volunteers.
- **GetMessages.java**: Retrieves all messages related to a specific incident. (admin)
- **GetMessagesUser.java**: Retrieves messages specific to a user.
- **GetMessagesVolunteer.java**: Retrieves messages specific to a volunteer

5. Administrator/Statistics Servlets

- **GetStatistics.java**: Provides statistics on incidents, volunteers, and user activity.
- **CreateSeats.java**: Handles the creation of volunteer positions for incidents.
- **GetActiveIncidentsForVolunteers.java**: Fetches active incidents that require volunteer participation.

6. Location and Proximity Servlets

- **GetUserLocation.java**: Retrieves the geographic location of a user.
- **CalculateNearbyIncidents.java**: Computes incidents near a user based on their location.
- **GetVolunteerHistory.java**: Provides a volunteer's participation history.

7. Database Initialization and Maintenance(default)

- **InitDB.java**: Initializes the database with necessary tables and default data.
- **DeleteDB.java**: Handles the deletion or cleanup of database records.

8. Utility Servlets

- **AddMessage.java**: Adds new messages to the message queue(DB).

3. FRONT END

1. HTML Files

- **index.html**: The main entry point for the application. Contains general information and links to other pages, including login, register, report incident.
- **admin.html**: Specific page for the admin role to manage incidents, users, and volunteers. Includes forms and interfaces for admin functionalities.
- **user.html**: Dedicated page for regular users to view incidents, report new ones, and track nearby incidents.
- **volunteer.html**: A page for volunteer firefighters to register, view, and participate in incidents.

2. CSS Files

- **index.css**: Handles the general styling for the main page and shared components.
- **admin.css**: Contains specific styles for admin pages, such as forms, tables, and statistics.
- **user.css**: Focused on styling the user interface for registered users, including maps and incident notifications.

3. JavaScript Files

- **ajax.js**: Implements AJAX functionality for real-time updates between the front-end and back-end (e.g., submitting forms, retrieving incident data).

- **script.js**: A script file for all front-end functionalities, such as handling user interactions, validation, or UI updates. Methods used in script.js:

Map-Related Functions

createMap()

Initializes an OpenLayers map and adds layers (OSM for map tiles and markers for incidents).Returns the map and markers objects.

setPosition(lat, lon)

Converts geographical coordinates (latitude and longitude) to the map's projection system.

addMarker(lat, lon, message)

Adds a marker on the map at the specified position with a popup message.

clearMarkers()

Clears all markers and popups from the map.

centerMap(lat, lon)

Centers the map on the provided coordinates.

Validation Functions

validatePasswords()

Validates the user's password for match, strength, and prohibited words.
Displays errors or success messages.

checkPasswordStrength()

Evaluates password strength based on length, character variety, and prohibited content.
Outputs feedback on the password's strength.

checkAgeForVolunteerFirefighter()

Ensures the user's age is between 18 and 55 if the user is registering as a volunteer firefighter.

Form Submission

handleFormSubmission()

Runs validations for the form fields and prepares data for submission.
Sends data as JSON to the appropriate backend servlet endpoint (RegistrationVolunteer or RegistrationUser).

submitIncident() / submitIncidentAdmin()

Sends a new incident's data to the server for creation.
Specifically used by users and admins.

updateIncident()

Updates existing incident details by sending a POST request with modified data.

updateUserInfo() / updateVolunteerInfo()

Sends updated user or volunteer information to the backend.

Data Retrieval and Display

loadUser() / loadVolunteer()

Fetches user or volunteer data and populates the form or display area with it.

loadIncidents() / loadIncidentsUser() / loadIncidentsVolunteer() / loadIncidentsGuest()

Fetches incidents from the server and displays them in a table format. Handles data for admins, users, volunteers, and guests.

viewStatistics()

Fetches and displays statistical charts using Google Charts (e.g., incident distribution, user/volunteer participation).

viewMessages() / viewUserMessages() / loadVolunteerMessages()

Retrieves and displays messages related to incidents, users, or volunteers.
viewUserMessages()

Filters and displays public messages for regular users.

selectIncident() / selectIncidentUser() / selectIncidentVolunteer()

Retrieves specific incident details and populates a form for review or update.

loadVolunteerHistory()

Retrieves participation history for volunteers and displays it in a table format.

loadActiveIncidents()

Fetches active incidents and displays cards for volunteers to accept participation.

Geocoding

verifyAddress()

Uses a third-party geocoding API to verify and validate the user's address. Displays errors or updates the map based on the location's validity.

showNearbyIncidents()

Fetches incidents and filters them based on proximity to the user’s location using the TrueWay Matrix API.

Messaging

sendPublicMessage() / sendUserMessage() / sendVolunteerMessage()

Sends messages between users, volunteers, and public recipients. Allows interaction related to incidents.

toggleRecipientOptions()

Toggles additional fields for specifying message recipients based on user selection.

Admin-Specific Functions

showAddIncidentForm()

Displays the form for adding a new incident.

createParticipantSeats()

Creates participant seats (e.g., vehicles, personnel) for an incident.

Login/Logout

loginPOST()

Handles login by sending user credentials to the server and redirecting to the appropriate dashboard.

logout()

Logs the user out and redirects to the login page.

DOM Manipulation and UI

togglePasswordVisibility()

Toggles the visibility of the password fields.

toggleVolunteerFields(show)

Shows or hides additional form fields for volunteer firefighters.

populateForm(user)

Automatically fills form fields with the data retrieved from the server.

drawIncidentChart(), drawUserVolunteerChart(), drawParticipantChart()

Renders Google Charts to visualize data like incidents, users, volunteers, and participants.

Utility Functions

createUserTableJSON(data) / createVolunteerCardJSON(data)

Converts user or volunteer data into HTML table or card format.

loadIncidentTableJSON(data) / Similar Functions

Converts incident data into a table format for display.

createEditableTableFromJSON(data)

Creates an editable table from JSON data for user or volunteer modifications.

AJAX was used in every method that linked the front end (script.js) with the back end(e.g. Login.java , Logout.java, UpdateIncident.java. UpdateUser.java , UpdateVolunteer.java)

REST was used in API.java that handles new incidents submissions

APIS that were used :
OpenLayers

- **Purpose:** For managing and displaying maps, adding markers, and handling geospatial data.
- **Key Responsibilities:**
  Initializes and renders the map.
  Handles transformations between coordinate systems.
  Adds and manages markers and popups on the map.

Google Charts

- **Purpose:** For visualizing data in charts (e.g., pie charts, bar charts).
- **Key Responsibilities:**
  Displays statistical data, such as the number of incidents, users, volunteers, and participants.
  Generates charts dynamically to provide insights into the data.

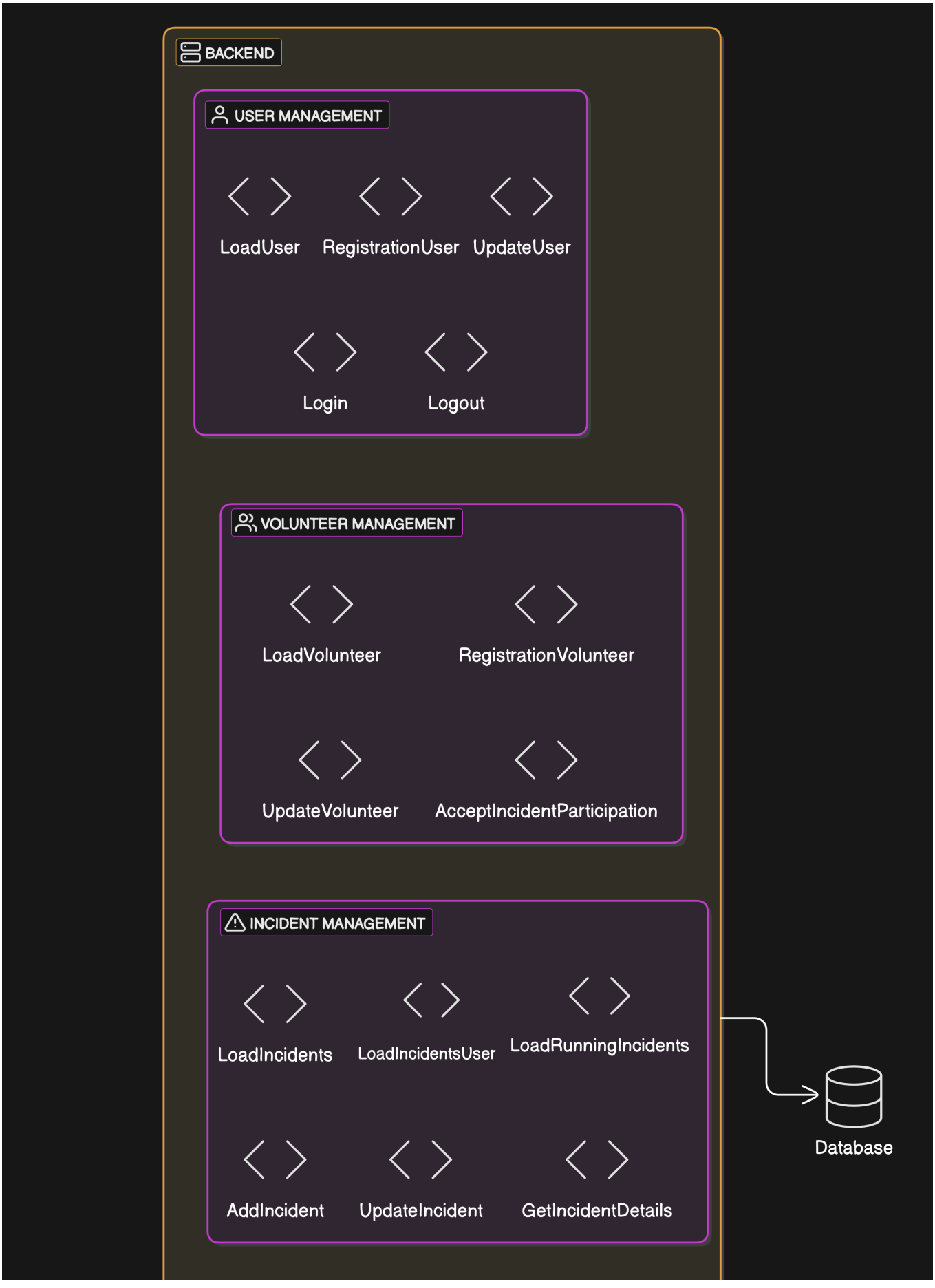
RAPID API Services (TrueWay Matrix and Forward-Reverse Geocoding)

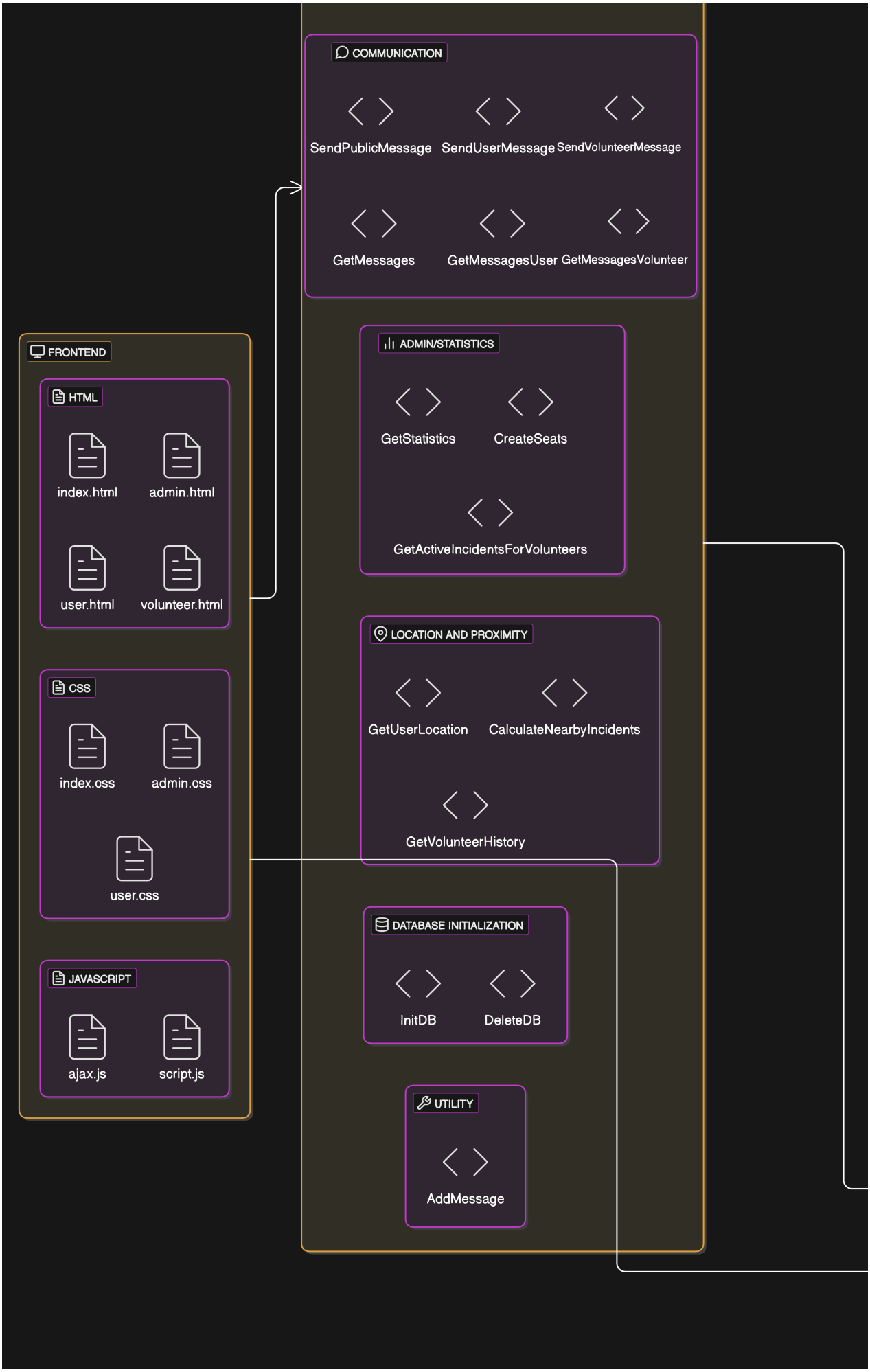
- **Purpose:** For geocoding addresses and calculating driving distances.
- **Key Responsibilities:**
  Validates and verifies user addresses.
  Filters incidents based on proximity to the user's location.

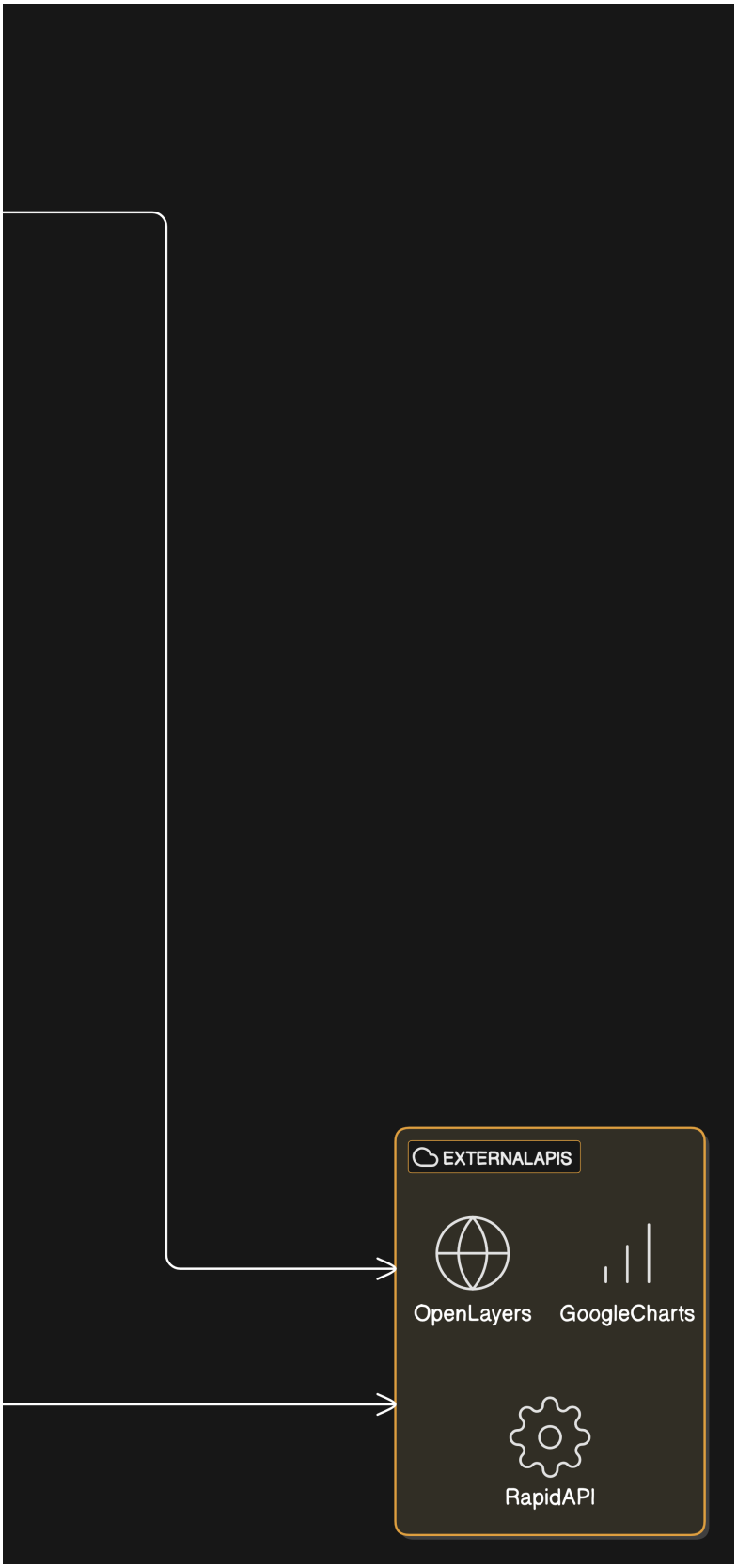
jQuery (used via $() calls)

- **Purpose:** Simplifies DOM manipulation and AJAX handling (although only lightly used in the file).
- **Key Responsibilities:**
  Updates or modifies DOM elements dynamically.

Handles success/error messages for server responses.




COMMUNICATION
SendPublicMessage
SendUserMessage
SendVolunteerMessage
GetMessages
GetMessagesUser
GetMessagesVolunteer
FRONTEND
HTML
index.html
admin.html
user.html
volunteer.html
CSS
index.css
admin.css
user.css
JAVASCRIPT
ajax.js
script.js
ADMIN/STATISTICS
GetStatistics
CreateSeats
GetActiveIncidentsForVolunteers
LOCATION AND PROXIMITY
GetUserLocation
CalculateNearbyIncidents
GetVolunteerHistory
DATABASE INITIALIZATION
InitDB
DeleteDB
UTILITY
AddMessage


EXTERNALAPIS
OpenLayers
GoogleCharts
RapidAPI