UNIVERSITATEA “ALEXANDRU IOAN CUZA” IAȘI

**FACULTATEA DE INFORMATICĂ**

**Lucrare de Licenta**

Disaster Response Coordination Platform

propusa de

Tanase George

Coordonator

Ioniță Alexandru

**Cuprins**

1. Introducere . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
2. Alte aplicatii asemanatoare . . . . . . . . . . . . . . . . . . . . . . . . .
3. Specificatii . . . . . . . . . . . . . . . . . . . . . . . .
4. Arhitectura aplicatiei . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
5. Tehnologii utilizate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
6. Introducere  
     
    În ultimele decenii, frecvența și intensitatea dezastrelor naturale, cum ar fi cutremurele, inundațiile, uraganele și incendiile forestiere, au crescut semnificativ. De asemenea, provocările generate de om, cum ar fi accidentele industriale și atacurile teroriste, contribuie la complexitatea situațiilor de urgență. În acest context, este esențial să dezvoltăm soluții eficiente pentru gestionarea și coordonarea răspunsului la dezastre, astfel încât să minimizăm pierderile umane și materiale și să sprijinim recuperarea rapidă a comunităților afectate.

Aplicația Disaster Response Coordination Platform, își propune să abordeze aceste provocări oferind o platformă centralizată și scalabilă pentru gestionarea situațiilor de urgență. Aceasta facilitează comunicarea în timp real, monitorizarea situației, alocarea resurselor și coordonarea voluntarilor, toate într-o singură interfață intuitivă. Ideea principală a proiectului este de a reduce întârzierile în răspunsul la dezastre și de a îmbunătăți eficiența operațională prin utilizarea tehnologiilor moderne și a algoritmilor avansați.

La baza dezvoltării acestei aplicații stau câteva principii esențiale: accesibilitatea, flexibilitatea și scalabilitatea. Accesibilitatea se referă la ușurința de utilizare a platformei de către diverse categorii de utilizatori, de la profesioniști în gestionarea situațiilor de urgență la voluntari și cetățeni. Flexibilitatea implică capacitatea platformei de a se adapta la diferite tipuri de dezastre și scenarii de urgență, oferind un set variat de instrumente și funcționalități. Scalabilitatea se referă la abilitatea platformei de a gestiona volume mari de date și utilizatori, fără a compromite performanța și stabilitatea.

Într-un scenariu de urgență, coordonarea eficientă a resurselor și a voluntarilor este crucială. Prin integrarea diferitelor surse de date, cum ar fi rapoartele de incidente, postările de pe rețelele sociale și datele de la senzori, aplicația noastră permite o evaluare rapidă și precisă a situației. Utilizatorii pot vizualiza datele în timp real pe o hartă interactivă, pot aloca resurse și voluntari în funcție de nevoile identificate și pot comunica eficient pentru a asigura un răspuns prompt și bine coordonat.

Un alt aspect important al aplicației este utilizarea algoritmilor avansați de optimizare și inteligență artificială pentru a îmbunătăți procesul de luare a deciziilor. De exemplu, algoritmii de optimizare ajută la distribuirea eficientă a resurselor, minimizând costurile de transport și timpul de livrare. Algoritmii de procesare a limbajului natural (NLP) și de învățare automată (ML) sunt utilizați pentru a analiza și categoriza rapoartele de incidente, prioritizând intervențiile în funcție de gravitatea și urgența situației.

Utilizarea tehnologiei moderne în gestionarea dezastrelor aduce numeroase beneficii, inclusiv reducerea timpilor de răspuns, îmbunătățirea comunicării și coordonării, și optimizarea utilizării resurselor. De exemplu, platformele digitale permit monitorizarea în timp real a situației, identificarea rapidă a nevoilor și alocarea eficientă a resurselor. Aceste avantaje contribuie semnificativ la reducerea impactului dezastrelor asupra comunităților și la sprijinirea recuperării rapide.

1. Alte aplicatii asemanatoare  
     
    Sahana Eden

**Descriere:** Sahana Eden este o platformă open-source dedicată gestionării dezastrelor și a situațiilor de urgență. Aceasta oferă un set complet de funcționalități pentru gestionarea resurselor, monitorizarea situațiilor și coordonarea răspunsului la dezastre.

**Funcționalități:** Printre caracteristicile cheie se numără gestionarea inventarului de resurse, urmărirea voluntarilor, crearea și gestionarea rapoartelor de incidente, și cartografierea locațiilor afectate. Sahana Eden este extrem de flexibilă și poate fi adaptată pentru diverse tipuri de dezastre și scenarii de urgență.

**Puncte forte:** Platforma este modulară și extensibilă, permițând adăugarea de noi funcționalități pe măsură ce nevoile utilizatorilor evoluează. De asemenea, fiind open-source, beneficiază de o comunitate activă de dezvoltatori care contribuie la îmbunătățirea continuă a software-ului.

**Limitări:** Implementarea și configurarea platformei poate fi complexă și necesită cunoștințe tehnice avansate. De asemenea, interfața de utilizator poate fi percepută ca fiind mai puțin intuitivă în comparație cu soluțiile comerciale.

**Crisis Cleanup**

**Descriere:** Crisis Cleanup este o platformă colaborativă dedicată coordonării eforturilor de curățenie și reconstrucție în urma dezastrelor. Aceasta permite organizațiilor și voluntarilor să colaboreze eficient, evitând duplicarea eforturilor și asigurând o alocare optimă a resurselor.

**Funcționalități:** Platforma oferă instrumente pentru gestionarea sarcinilor, urmărirea progresului, și comunicarea între echipe. Utilizatorii pot raporta nevoile de curățenie, iar voluntarii pot prelua aceste sarcini în funcție de capacitățile și disponibilitatea lor.

**Puncte forte:** Crisis Cleanup se concentrează pe colaborarea între diverse organizații, facilitând partajarea informațiilor și resurselor. Platforma este ușor de utilizat și nu necesită cunoștințe tehnice avansate.

**Limitări:** Funcționalitățile sunt orientate în principal către activitățile de curățenie și reconstrucție, limitându-se astfel la un anumit tip de răspuns la dezastre. Nu oferă un set complet de instrumente pentru gestionarea resurselor sau monitorizarea situațiilor în timp real.

**RapidSOS**

**Descriere:** RapidSOS este o platformă avansată de gestionare a apelurilor de urgență, care integrează date de locație și informații suplimentare pentru a îmbunătăți răspunsul la apelurile de urgență. Aceasta colaborează cu serviciile de urgență pentru a oferi date precise și relevante în timp real.

**Funcționalități:** Platforma furnizează date de locație exacte, informații medicale, și alte date critice care pot ajuta la salvarea vieților. RapidSOS integrează date din diverse surse, inclusiv aplicații mobile, dispozitive conectate, și senzori IoT.

**Puncte forte:** Integrarea avansată a datelor și capacitatea de a furniza informații precise și relevante în timp real sunt punctele forte ale RapidSOS. Aceasta contribuie semnificativ la îmbunătățirea răspunsului la apelurile de urgență și la coordonarea intervențiilor.

**Limitări:** Platforma este orientată în principal către gestionarea apelurilor de urgență și poate să nu ofere toate instrumentele necesare pentru o gestionare completă a dezastrelor și a resurselor.

**One Concern**

**Descriere:** One Concern utilizează inteligența artificială și învățarea automată pentru a oferi analize predictive și evaluări ale impactului dezastrelor. Aceasta ajută orașele și organizațiile să se pregătească mai bine pentru dezastre și să răspundă eficient în situații de criză.

**Funcționalități:** Platforma oferă previziuni detaliate ale impactului dezastrelor, inclusiv daunele anticipate și nevoile de resurse. One Concern utilizează date geospațiale și modele predictive pentru a furniza informații precise și acționabile.

**Puncte forte:** Capacitatea de a oferi analize predictive detaliate și evaluări ale impactului dezastrelor reprezintă un avantaj major al One Concern. Aceasta ajută la luarea deciziilor informate și la planificarea eficientă a răspunsului.

**Limitări:** Platforma poate fi costisitoare și necesită acces la date de înaltă calitate și la expertiză tehnică pentru a implementa și utiliza pe deplin funcționalitățile sale.

1. Sepecificatii

#### 3.1 Funcționalități Principale

1. **Gestionarea Voluntarilor**
   * **Înregistrare și Autentificare**: Voluntarii pot crea conturi și se pot autentifica în aplicație.
   * **Profil de Voluntar**: Fiecare voluntar are un profil care include informații despre competențe, disponibilitate și locație.
   * **Alocare Automată**: Algoritmul de potrivire a voluntarilor cu sarcini pe baza competențelor, disponibilității și locației.
2. **Gestionarea Resurselor**
   * **Inventar de Resurse**: Baza de date centralizată pentru gestionarea stocurilor de resurse (alimente, apă, materiale medicale etc.).
   * **Monitorizarea Resurselor**: Actualizarea în timp real a stocurilor și locațiilor resurselor.
   * **Alocare Optimizată**: Algoritm de optimizare a distribuției resurselor în funcție de cerere și disponibilitate, utilizând tehnici de programare liniară.
3. **Raportarea Incidentelor**
   * **Colectarea Datelor**: Agregarea rapoartelor de incidente din diverse surse (rețele sociale, rapoarte directe, senzori).
   * **Analiza și Categorizarea Incidentelor**: Utilizarea procesării limbajului natural (NLP) și a algoritmilor de clusterizare pentru a analiza și categoriza incidentele.
   * **Mapare Incidentelor**: Vizualizarea în timp real a incidentelor pe o hartă interactivă.
4. **Rute Optimizate**
   * **Algoritmi de Rutare**: Utilizarea algoritmilor Dijkstra sau A\* pentru determinarea celor mai optime rute pentru voluntari și resurse.
   * **Integrarea Datelor de Trafic**: Actualizarea rutelor în funcție de datele de trafic și condițiile drumurilor.
5. **Comunicare în Timp Real**
   * **Mesagerie Instantanee**: Canal de comunicare în timp real pentru coordonarea echipelor și distribuirea informațiilor critice.
   * **Notificări**: Alarme și notificări pentru incidente noi, nevoi de resurse și modificări ale rutelor.

#### 3.2 Specificații Tehnice

1. **Backend**
   * **Limbaj de Programare**: Python
   * **Framework**: Flask pentru gestionarea API-urilor și logica aplicației.
   * **Baza de Date**: MySQL pentru gestionarea datelor persistente.
2. **Frontend**
   * **Limbaj de Programare**: JavaScript
   * **Framework**: React pentru construirea interfeței utilizatorului.
   * **Hărți**: Google Maps API pentru vizualizarea hărților și datelor geografice.
3. **Algoritmi**
   * **Potrivirea Voluntarilor**: Algoritm de potrivire bazat pe scoruri calculate în funcție de competențe, disponibilitate și locație.
   * **Optimizarea Resurselor**: Algoritmi de programare liniară pentru optimizarea distribuției resurselor.
   * **Analiza Incidentelor**: Algoritmi de NLP și clusterizare pentru analiza și clasificarea rapoartelor de incidente.
   * **Rutare Optimizată**: Algoritmii Dijkstra și A\* pentru calcularea celor mai eficiente rute.
4. Arhitectura Aplicatiei

 **Înregistrare și Autentificare Utilizatori**

* Utilizatorul trimite cererea de înregistrare/autentificare din interfața React.
* Cererea este procesată de endpoint-ul corespunzător din Flask.
* Baza de date MySQL verifică și stochează informațiile utilizatorilor.
* Răspunsul este trimis înapoi către frontend pentru a actualiza starea aplicației.

 **Raportarea și Monitorizarea Incidentelor**

* Utilizatorul raportează un incident din interfața React.
* Incidentul este trimis la backend și stocat în baza de date.
* Algoritmii de NLP și clusterizare procesează datele și actualizează prioritățile.
* Datele sunt afișate în timp real pe harta interactivă.

 **Alocarea Resurselor și Voluntarilor**

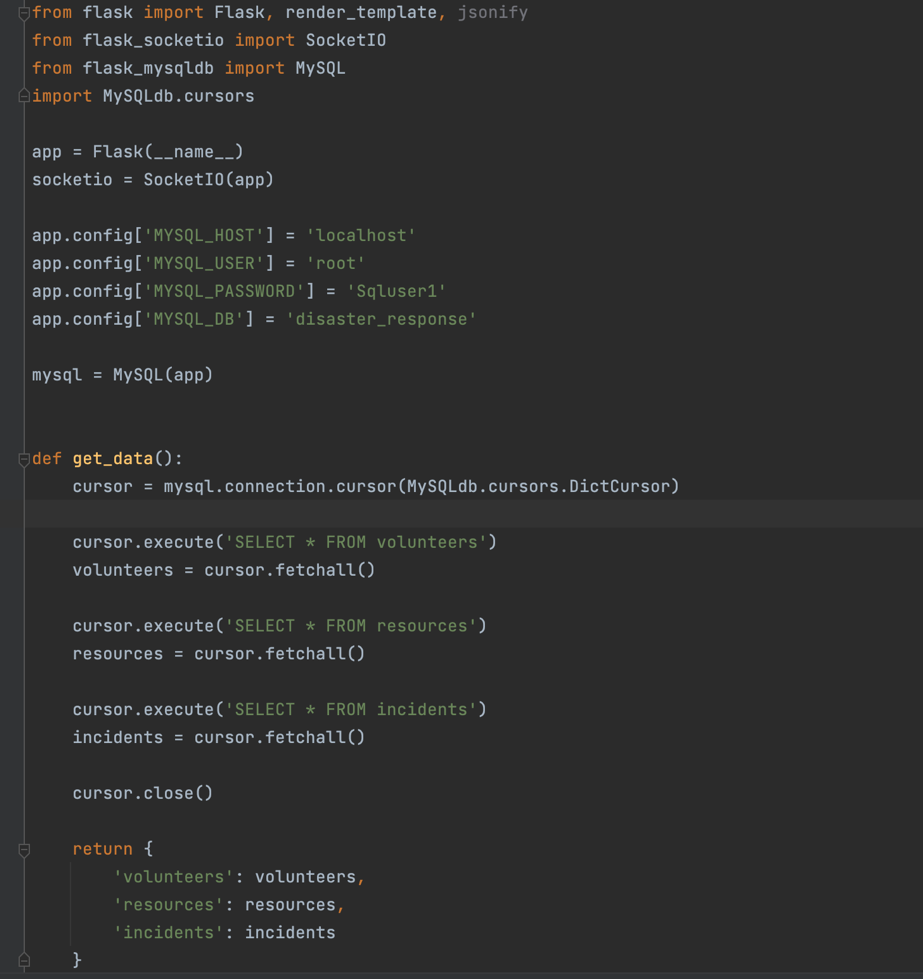
* Cererea de resurse sau voluntari este trimisă din interfața de administrare.
* Algoritmii de potrivire și optimizare procesează cererea.
* Alocările sunt stocate în baza de date și vizualizate în interfața utilizatorului.

 **Planificarea și Executarea Rutelor**

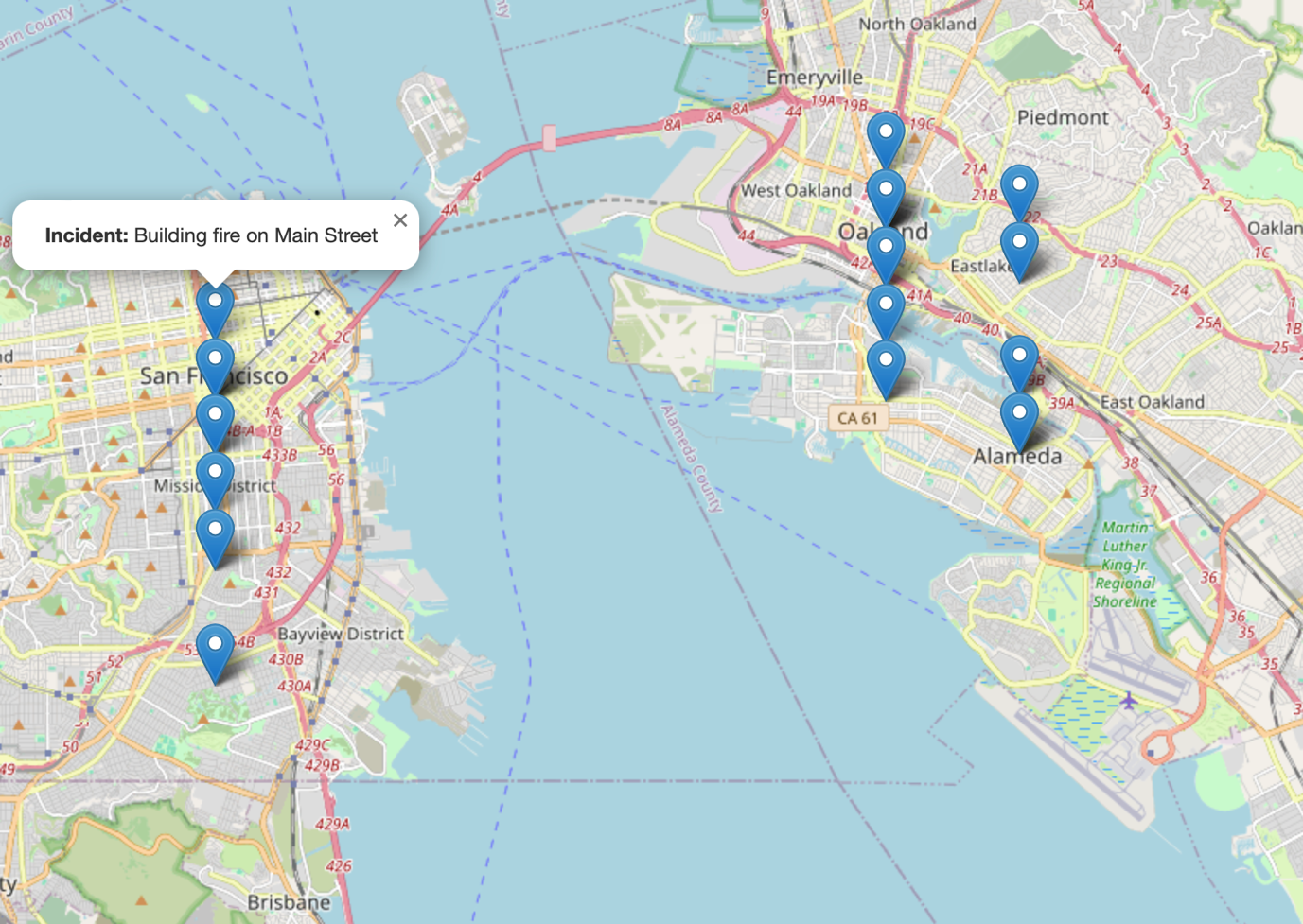
* Datele de trafic și condițiile drumurilor sunt trimise către backend.
* Algoritmii de rutare calculează rutele optime.
* Rutele sunt actualizate în timp real în interfața de utilizator.

Feature-uri Implementate:

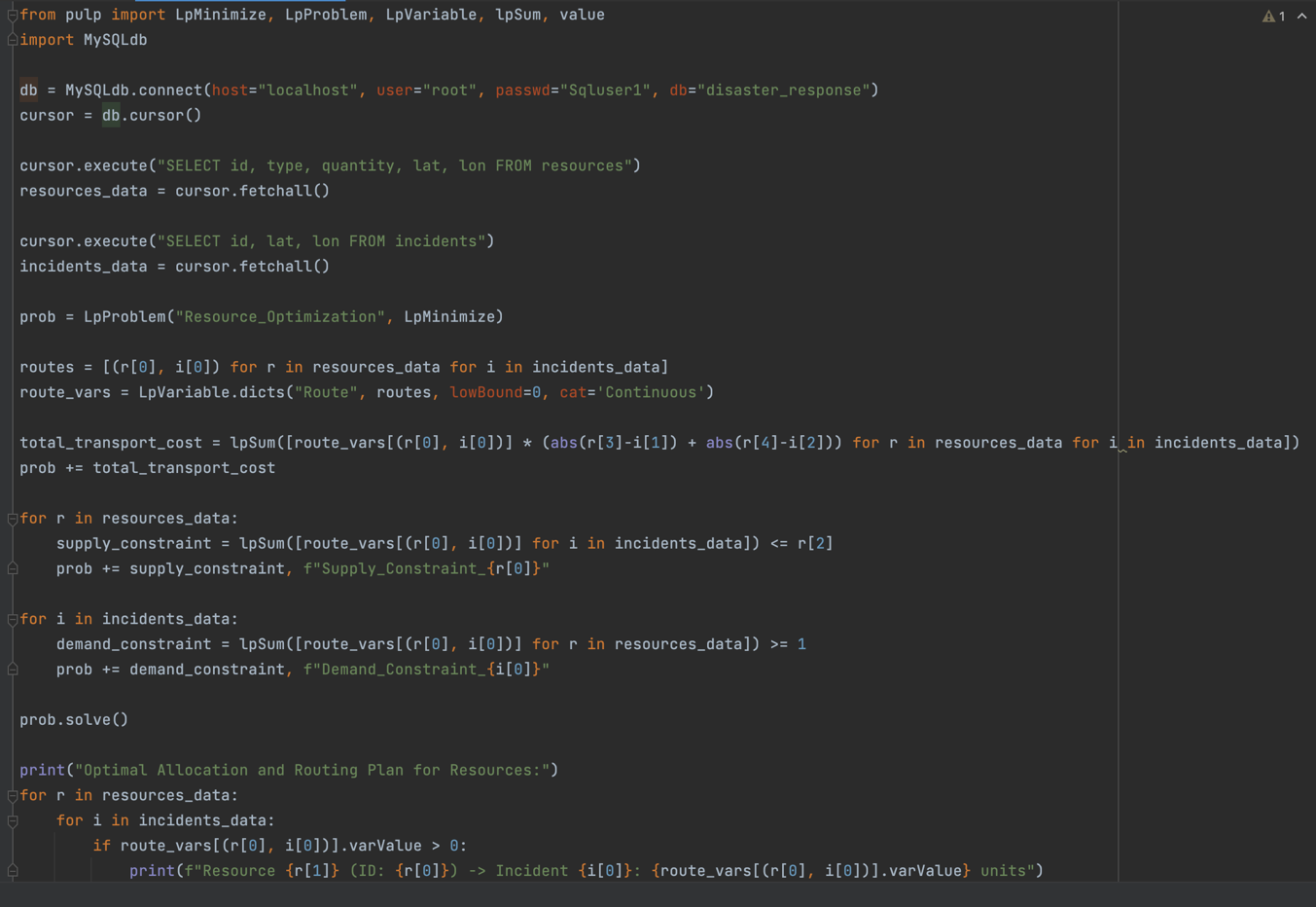
Real-Time Mapping:

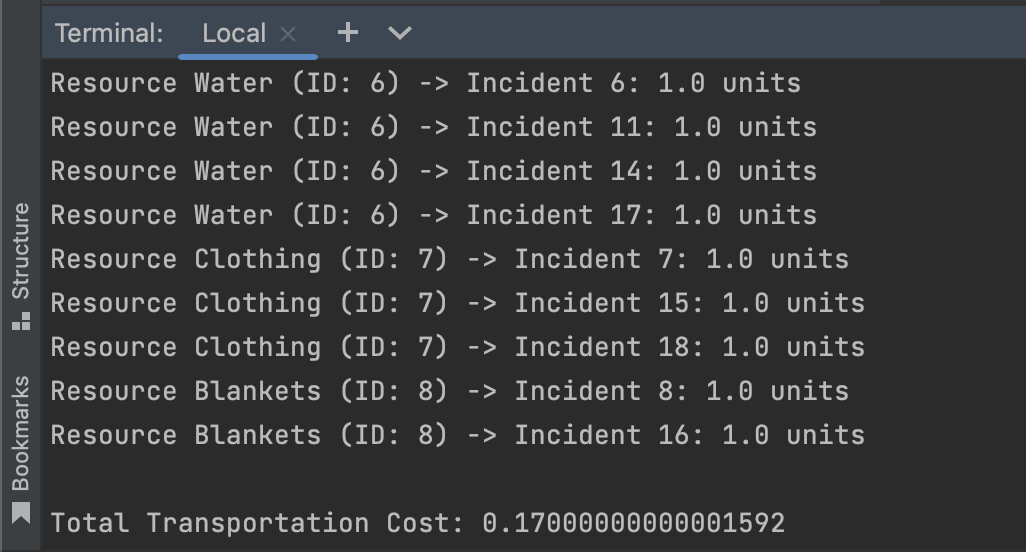






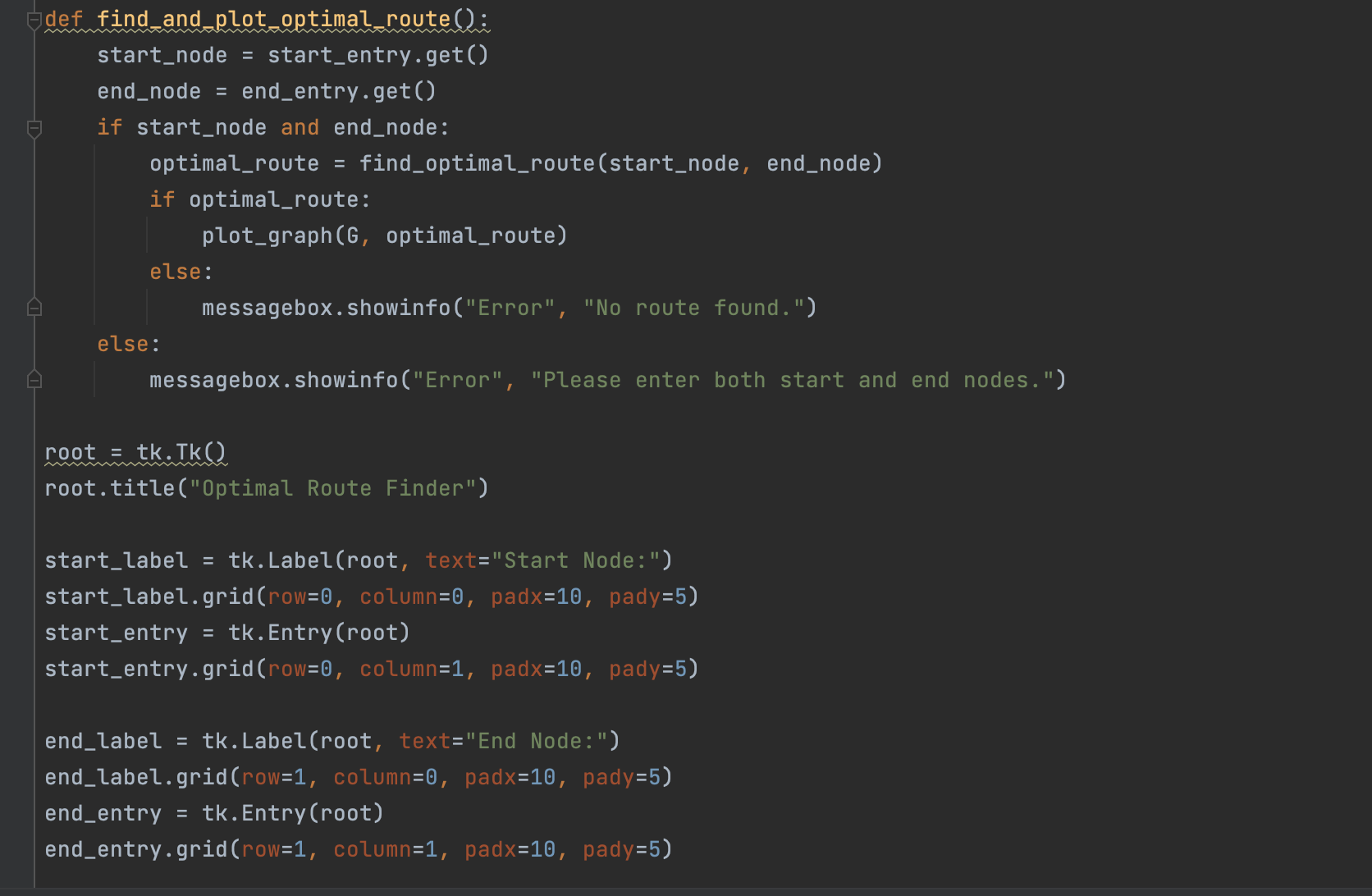
Resource Optimization:

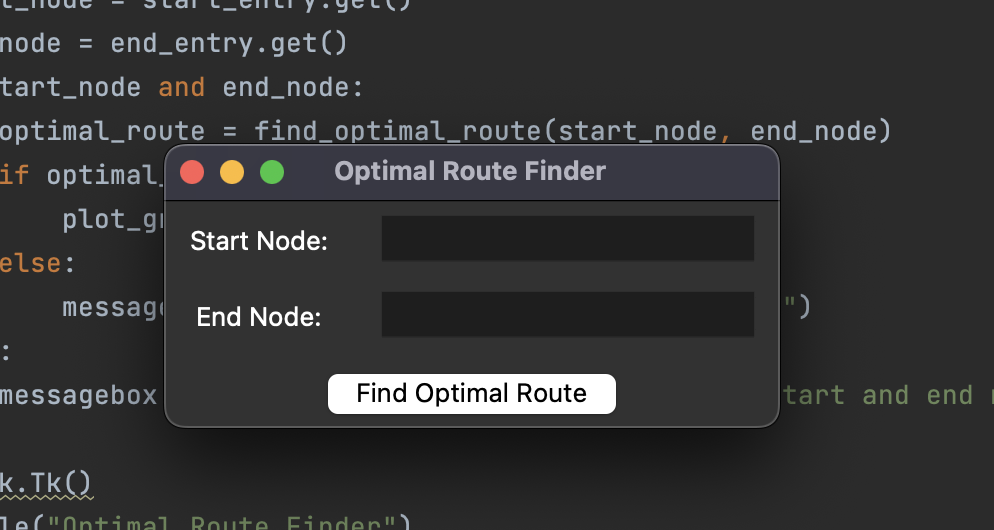


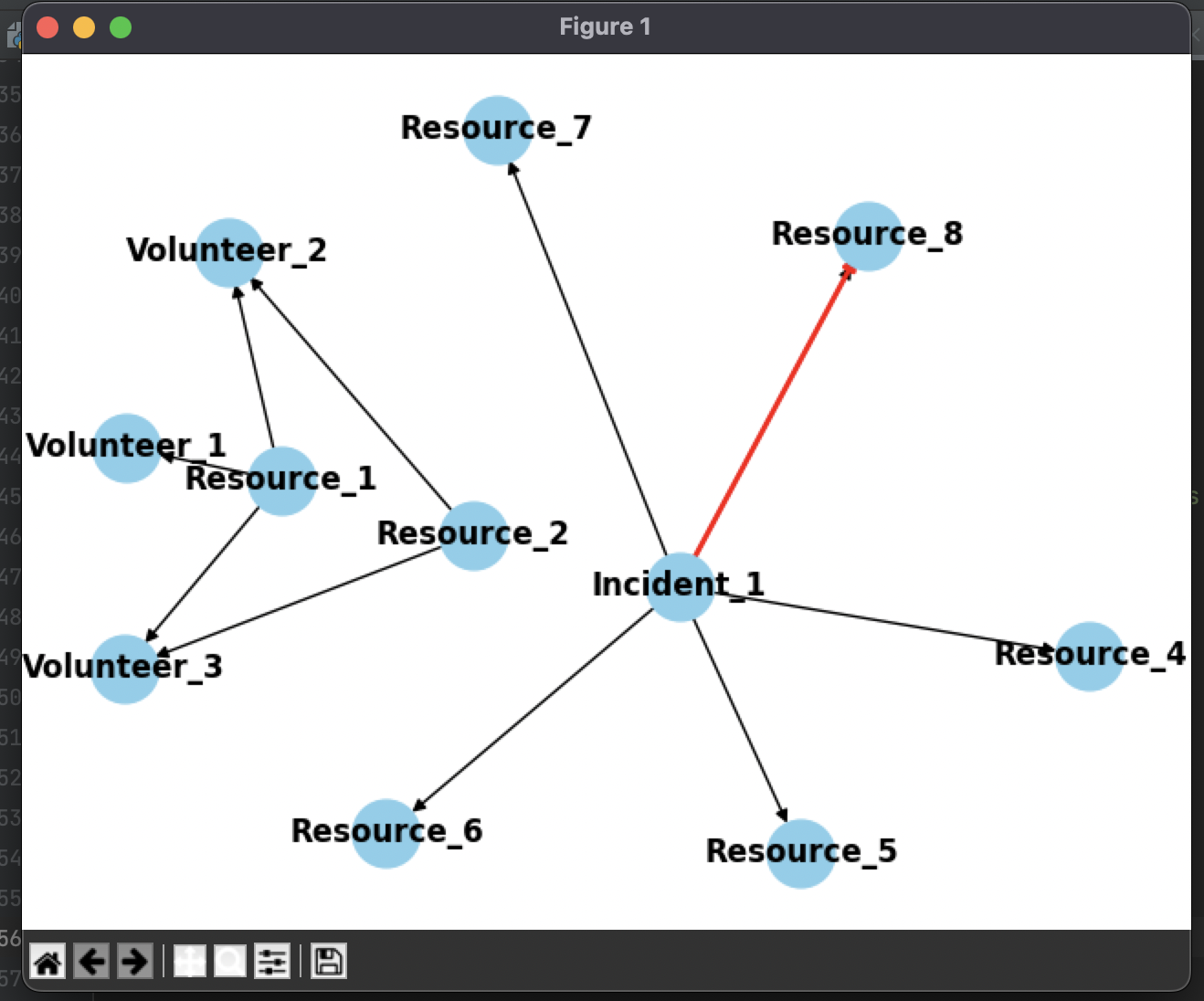


Routing-algorithm:









1. Tehnologii utilizate  
     
   5.1 Tehnologii Utilizate Până în Momentul de Față  
     
    **Flask**: Framework Python pentru dezvoltarea aplicațiilor web. Flask este utilizat pentru gestionarea logicii de afaceri și a endpoint-urilor API RESTful.

####  **MySQL**: Sistem de gestionare a bazelor de date relaționale. MySQL este utilizat pentru stocarea și gestionarea datelor persistente ale aplicației.

 **NLP (Natural Language Processing)**: Tehnologii și biblioteci precum NLTK și spaCy sunt utilizate pentru analiza și procesarea textelor din rapoartele de incidente și postările de pe rețelele sociale.

####  **Clustering Algoritms**: Algoritmi precum K-means sau DBSCAN pentru categorisirea și prioritizarea incidentelor.

 **Scipy și PuLP**: Biblioteci Python pentru optimizare și programare liniară. Acestea sunt utilizate pentru a rezolva problemele de optimizare a distribuției resurselor.

####  **NetworkX**: Utilizat pentru implementarea algoritmilor de rutare precum Dijkstra sau A\* pentru găsirea celor mai scurte trasee.

#### 5.2 Tehnologii ce urmeaza a fi implementate

 **Celery**: Sistem de task queue distribuit pentru a gestiona sarcini asincrone și programate. Celery va fi utilizat pentru sarcini de fond, precum procesarea datelor în loturi și actualizările periodice.

 **Docker**: Platformă pentru containerizarea aplicațiilor. Docker va facilita dezvoltarea, testarea și implementarea aplicației în medii izolate și replicabile.

####  **Kubernetes**: Orchestrator pentru containere Docker. Kubernetes va asigura scalabilitatea și gestionarea eficientă a resurselor în producție.

 **Machine Learning pentru Potrivirea Voluntarilor**: Algoritmi de învățare automată, precum Random Forests sau Gradient Boosting, pentru a îmbunătăți potrivirea voluntarilor cu sarcinile pe baza unui set de caracteristici extins.

 **Optimizare Avansată a Rutelelor**: Algoritmi precum Genetic Algorithms sau Particle Swarm Optimization pentru găsirea soluțiilor optime în probleme complexe de rutare.

####  **Deep Learning pentru Analiza Incidentelor**: Rețele neuronale convoluționale (CNN) pentru analiza imaginilor sau rețele neuronale recurente (RNN) pentru analiza secvențelor de text din rapoartele de incidente.

 **Slack API**: Integrarea aplicației cu Slack pentru notificări și actualizări în timp real către echipele de răspuns la dezastre.

####  **Twilio**: API pentru trimiterea de mesaje SMS și notificări telefonice către voluntari și echipe de intervenție.