IZMIR UNIVERSITY OF ECONOMICS

FACULTY OF ENGINEERING

COMPUTER, ELECTRIC & ELECTRONICS, INDUSTRIAL
and SOFTWARE ENGINEERING

FENG 498 PROJECT REPORT

Post-Disaster Needs and Logistics Management System

Author(s): Büşra Yılmaz, Murat Vermez, Mustafa İncu, Onat Filik, Yaren Gökçen Güdük

Supervisor: Ahmet Sermet Anagün

CONTENTS

1. Abstract

The project aimed to develop an AI-driven logistics and needs management system for post-disaster scenarios, incorporating advanced technologies to address critical challenges in aid distribution. The system combines real-time data analytics, AI-based route optimization, drone-assisted road condition analysis, and a centralized mobile application to streamline coordination between stakeholders. This ensures efficient and secure delivery of aid during crises.

One of the core motivations for this project was the inefficiencies observed during the February 2023 earthquake in Turkey, where generalized relief strategies resulted in resource mismatches and delays in critical supply distribution. These challenges highlighted the need for a more targeted, data-driven approach to aid delivery, one capable of addressing specific needs in a timely manner.

The project introduced several innovative solutions. The AI algorithms dynamically optimized routes, factoring in real-time road conditions and changes to ensure safe and efficient transportation. Additionally, the system enabled the strategic placement of temporary depots between neighborhoods, adapting locations based on the scale and geography of the disaster. A needs management component leveraged demographic data and AI models to identify suitable alternatives for unavailable medications. This approach also incorporated centralized stock control in warehouses, allowing for precise inventory management and demand forecasting, ensuring sufficient supplies for unforeseen demand surges.

These studies illustrate the transformative potential of integrating AI into disaster logistics. By providing a scalable and adaptable solution, the proposed system addresses the critical gaps in traditional relief strategies, offering a robust framework to improve disaster resilience and facilitate efficient recovery in vulnerable regions.

2. Introduction

Natural disasters, including earthquakes, floods, and forest fires, often expose critical gaps in traditional relief systems. The devastating earthquake in Turkey on February 6, 2023, highlighted significant inefficiencies in aid distribution, particularly in delivering essential supplies and medications to affected regions. Generalized strategies often lead to oversupply in some areas while critical shortages persist in others, exacerbating the challenges faced by disaster survivors. To address these issues, this project proposes a comprehensive solution leveraging artificial intelligence, real-time data analytics, and mobile technology for post-disaster logistics and needs management.

The proposed system combines dynamic AI-driven route optimization, demographic profiling for personalized aid distribution, and drone-assisted road condition analysis. The design includes the strategic placement of temporary depots between neighborhoods to enhance accessibility and adaptability during crises. The system also incorporates AI models to identify substitutes for unavailable medications and manage centralized stock control in warehouses, ensuring precise inventory management and rapid demand forecasting. These innovations aim to revolutionize disaster response systems by improving efficiency, reducing wastage, and ensuring aid reaches those in need promptly.

2.1. Problem Statement

In earthquake-prone regions like Turkey, existing disaster relief systems have shown significant limitations in addressing the specific needs of affected individuals and communities. Traditional models often fail to consider demographic variations, resulting in resource mismatches and delays in aid delivery. These inefficiencies are particularly evident in the distribution of medications and essential supplies, where the absence of a dynamic, personalized approach can lead to severe consequences for survivors. Additionally, the lack of real-time adaptation to road and infrastructure conditions further complicates the logistics of aid distribution.

2.2. Motivation

The urgency of addressing the inefficiencies in disaster relief systems became evident during the February 2023 earthquake in Turkey, where delays and mismatches in resource distribution had critical repercussions. Developing an AI-driven logistics and needs management system addresses these gaps by introducing a personalized, data-driven approach to aid delivery.

By leveraging AI technologies, the system optimizes routes dynamically, adapts to changing road conditions in real-time, and ensures that temporary depots are strategically positioned between neighborhoods for improved accessibility. Additionally, AI-powered medication substitution ensures that survivors receive necessary medical supplies even when primary options are unavailable. Centralized stock control and real-time demand forecasting further enhance the system's efficiency, reducing wastage and ensuring timely distribution of aid. This project represents a critical step toward improving disaster resilience and setting a new benchmark for humanitarian logistics.

3. Literature Review

A literature review for this study was conducted under three separate headings: **P-Median Model**, **Route Optimization**, and **Fine-Tuning of AI Models**. The P-Median Model is a critical method for determining the optimal facility placement in logistics and resource distribution. In this context, the studies reviewed shed light on the logistics management models to be utilized in the project. On the other hand, the fine-tuning of artificial intelligence models plays a crucial role in accurately identifying equivalent medications within the project. This section examines the relevant processes and methods used. Lastly, route optimization holds great significance in ensuring fast and efficient delivery of aid in post-disaster scenarios. The studies conducted in this area provide valuable insights into how route optimization can be integrated into the project.

3.1 P-Median Model

The study titled "P-median Facility Location Selection Problem and Solution Approaches" by Bastı (2012) examines the p-median problem, which has an important place among facility location selection problems, in detail. In the study, the main purpose of the p-median problem to minimize the distances of service points to demand points is emphasized and the mathematical models used in solving this problem are examined. In particular, the applicability of heuristic and meta-heuristic algorithms as well as exact solution methods for reducing logistics and transportation costs are discussed. The study supports theoretical knowledge with a practical example by showing how the p-median model can be applied to determine the warehouse locations of a food company in Düzce province in Turkey. In this context, the methods proposed for solving the p-median problem shed light on future studies on improving logistics processes and optimizing resource allocation.

The study titled "P-Median Problems and Solution Strategies" prepared by Siegel (2021) focuses on p-median problems and solution strategies in the field of operations planning, scheduling and control. The study emphasizes the importance of the p-median problem as an optimization model that aims to minimize total transportation costs by locating facilities in the most appropriate way to demand points. In addition to exact solution methods, the use of heuristic and meta-heuristic algorithms is discussed and how these approaches can be applied especially to large-scale problems is explained. The study reveals how effective use of mathematical models and heuristic strategies can provide critical improvements in resource allocation and logistics management. In this context, solution proposals for p-median problems make a significant contribution to the development of effective decision support systems in the field of operations management.

The study titled "Solution Approach to P-Median Facility Location Problem with Integer Programming and Genetic Algorithm" prepared by Ekin (n.d.) addresses the p-median facility location selection problem and details two different approaches to solving this problem: integer programming and genetic algorithm. The study emphasizes the critical role of the p-median problem in the context of locating facilities in the most appropriate way to demand points and minimizing transportation costs. While examining how the exact solution can be obtained with the integer programming method, it is shown that the genetic algorithm can provide an effective alternative for larger and more complex problems. In addition, a comparison is made to evaluate the applicability and effectiveness of both methods, and the problem solutions are supported with concrete examples. The study offers innovative and effective approaches for solving p-median problems in logistics and operations management.

The master's thesis titled "P-Median Facility Location Selection with Meta-Heuristic Approaches" prepared by Sümer (n.d.) investigates the effectiveness of meta-heuristic approaches in solving the p-median problem. The study emphasizes the importance of the p-median problem, especially in terms of optimally locating facilities at demand points and minimizing transportation costs. Meta-heuristic methods such as genetic algorithms, tabu search and particle swarm optimization are examined and it is shown that these methods are effective solution tools for large-scale and complex problems. The study compares the solution performances of different meta-heuristic approaches and provides a comprehensive assessment of which method is more suitable under which conditions.

3.2 Fine Tuning of AI Models

Goodfellow and others stated that fine-tuning is a critical technique in machine learning that involves adapting a pre-trained model to perform a specific task more effectively. In our project, fine-tuning was utilized to customize a language model for identifying equivalent medications, addressing the challenges of medication shortages in post-disaster scenarios.

The Fine-Tuning Process

1. **Selection of a Pre-Trained Model**: We began with a large language model that had been pre-trained on extensive general text corpora. This provided a strong foundational understanding of language patterns and semantics.
2. **Domain-Specific Data Collection**: A specialized dataset was assembled, comprising medication names, active ingredients, usage indications, dosages, and known equivalents. This dataset was crucial for teaching the model the specific relationships and nuances within pharmaceutical data.
3. **Data Preprocessing**: The dataset was cleaned and formatted to ensure consistency. Steps included handling missing values, normalizing text (e.g., converting all text to lowercase), and removing duplicates.
4. **Fine-Tuning Procedure**:
    - **Hyperparameter Optimization**: Key hyperparameters such as learning rate, batch size, and the number of epochs were carefully selected. A lower learning rate was chosen to prevent drastic updates to the model weights, preserving valuable pre-trained features.
    - **Training**: The model was trained on the domain-specific dataset, adjusting its parameters to minimize the loss function associated with predicting correct medication equivalents.
    - **Validation**: A portion of the data was set aside for validation to monitor the model's performance and prevent overfitting. Performance metrics such as accuracy, precision, recall, and F1 score were used to evaluate the model.
5. **Integration into the System**: The fine-tuned model was integrated into our application, enabling real-time suggestions for medication substitutes based on user input.

Application in the Project

- **Accurate Medication Substitution**: The fine-tuned model can suggest appropriate alternative medications when a specific drug is unavailable, considering factors like active ingredients and therapeutic class.
- **Personalized Recommendations**: By incorporating user-specific medical data (e.g., allergies, chronic conditions), the model ensures that the suggested alternatives are safe and suitable for each individual.

- **Efficiency in Emergencies**: The model's ability to provide quick and accurate recommendations is particularly valuable in disaster settings, where time and resources are limited.

Advantages of Using Fine-Tuning

- **Resource Efficiency**: Fine-tuning leverages existing pre-trained models, reducing the computational resources and time required compared to training a model from scratch.
- **Enhanced Performance**: Adapting the model to domain-specific data improves its accuracy and reliability in making predictions relevant to our application.
- **Flexibility**: The model can be updated with new data, allowing it to remain current with the latest medical guidelines and medication availability.

Challenges and Solutions

- **Data Quality**: The effectiveness of fine-tuning is highly dependent on the quality of the domain-specific dataset. To address this, data was sourced from reputable medical databases and underwent thorough validation.
- **Ethical Considerations**: Ensuring patient safety is paramount. The model's suggestions are designed to support, not replace, professional medical judgment. Users are advised to consult healthcare professionals before making decisions based on the model's output.
- **Hyperparameter Selection**: Finding the optimal hyperparameters required experimentation. Techniques such as grid search were employed to systematically evaluate different configurations.

Impact on Disaster Response Efforts

By implementing a fine-tuned AI model, our project significantly enhances the ability to manage medication shortages during disasters:

- **Improved Accessibility**: Aids healthcare providers in quickly finding viable medication alternatives, ensuring continuity of care.
- **Reduced Burden on Medical Staff**: Automating the identification of substitutes allows medical professionals to focus on other critical tasks.
- **Adaptability**: The system can be updated as new medications become available or as guidelines change, maintaining its relevance over time.

3.3 Route Optimization

In route optimization and artificial intelligence applications, several works have been identified that show how AI can change logistics, especially under complex and dynamic conditions. This section gives an overview of some of the important works that form the backbone for AI-based route optimization in the project.

AI-Based Route Optimization: A Logistical Process Software Example

Ünal, Kazan, and Çuhadar (2022) present a study on the application of AI-based route optimization in logistical processes. This research underlines the potential of AI to realize cognitive functions like perception, reasoning, learning, and problem-solving in the logistics domain. Based on these capabilities, AI systems can support decision-making processes, especially under dynamically changing conditions. It assists in the attainment of resource optimality and also shortens the duration it takes to get something delivered to someone's doorstep; therefore, these AI logisticians ensure that its attainment meets project aims for finding routes efficiently and adaptively.

AI and ML for Transportation Route Optimization

Krishna Vaddy (2023) discusses in detail how AI and ML can be exploited for improving transportation route optimization. The study highlights the incorporation of AI algorithms for real-time data processing, which can make adjustments dynamically due to changes in ground conditions such as heavy congestion of traffic, accidents, or closure of roads. Consequently, with the use of predictive analytics, the system can forecast any disruption and reroute the vehicles proactively to maintain efficiencies. This will reduce travel time and operational cost, hence improving the overall service reliability. The results have shown that AI and ML mechanisms need to be added to modern transportation to develop adaptive and resilient route optimization solutions.

An Elementary Approach to the Vehicle Routing Problem via Python and Google API

Lian, K.Q. and Tribello, G.A. (2024) presents methods for solving the Vehicle Routing Problem using techniques for mixed integer linear programming. The inputs for unique time windows and locations of each customer to take them to their final destination with the use of a fleet are optimized by Python and the Google Maps API. Salient features of the approach include automated data integration through the API to reduce manual inputs and troubleshooting methods to debug infeasibilities in routing plans. This is supported, with a node partitioning strategy in order to reduce computational times and enhance scalability towards real-world applications.

Application to the Project

These studies will directly inform the development of the AI-supported route optimization framework in this project. This is particularly true for Python and Google Maps API, as noted by Lian and Tribello (2024), which offer pragmatic and highly scalable methods to automate data integration, hence reducing the effort needed from a human expert during the routing process. Such APIs enable the system to dynamically fetch real-time traffic information, road conditions, and location coordinates that will be put to work in effective route optimization.

This, added to the emphasis on adaptability of AI in dynamic conditions, performing real-time problem solving, is also consistent with the project objective that plans to optimize routes by updating conditions in real time in roads and depots. In that direction, the integration of such principles will ensure the system can be robust and adaptive enough to effectively handle complex logistical problems.

The essence is that the integration of AI and ML into transportation route optimization can bring about great efficiency, adaptability, and reliability. In summary, the reviewed studies will provide useful insights and methodologies that can be applied to further enhance the effectiveness of the AI-based route optimization system proposed in this project.

4. Methodology

This study aims to establish an integrated system for addressing post-disaster aid and needs. In the first stage, the P-Median model was utilized to determine the most suitable locations for warehouses in disaster-affected regions. Subsequently, data such as medication usage, height, and weight were collected from individuals in the region through a mobile application and stored in a database. In this process, artificial intelligence methods were employed for needs analysis based on the collected data; body size metrics were used for clothing size estimation, and artificial intelligence applications were utilized for determining medication alternatives. Furthermore, a detailed method was provided for managing the stock levels of the warehouses via datasets. Finally, an optimal route analysis was conducted for individuals accessing warehouses based on their locations in the event of a disaster, and the methods for this analysis were elaborated. This methodology comprehensively addresses all phases of the study and explains the integration process.

4.1 Determining the Most Suitable Locations For Warehouses

4.1.1 Basic Structure and Classification of Facility Location Problem

The facility location problem is an important optimization problem in the field of logistics and operations management and generally aims to place facilities (such as warehouses, production centers, service points) in the most appropriate way to certain demand points. The basic structure of the problem is based on minimizing transportation costs, shortening service times and optimizing resource allocation. Facility location problems are divided into different categories depending on the determined goals and constraints: p-median problems, p-maximum coverage problems and capacitated facility location problems are some of them. They can also be classified as deterministic and stochastic models. While deterministic models work with fixed data sets, stochastic models take into account uncertainties and variable demands. The facility location problem stands out as a critical research area to increase the efficiency of both industrial and humanitarian logistics processes.

4.1.2 Warehouse Location Problem in Natural Disaster Cases

Effective aid distribution after natural disasters requires fast and efficient access to demand points in disaster areas. Warehouses play a critical role in this process, and correct placement ensures that relief supplies can be delivered to the right points on time. The warehouse placement problem refers to the strategic placement of warehouses in order to optimize post-disaster logistics processes. This placement aims to ensure that warehouses are close to the disaster area and that every demand point can be reached quickly. In emergencies such as natural disasters, correct placement of warehouses aims to ensure that people in the disaster area can reach the nearest warehouse and that relief supplies are distributed quickly and effectively. This model includes strategic locations where people can easily access the closest warehouses to ensure efficiency in post-disaster logistics processes.

4.1.3 Coverage-Based Warehouse Layout Models

Coverage-based warehouse layout models aim to strategically place warehouses in order to ensure that people can reach the closest warehouse after a disaster. In these models, each warehouse has a certain coverage area, and all demand points within this coverage area, i.e. disaster victims, can reach the closest warehouse. In the maximum coverage model, the maximum possible demand points are covered with a certain number of warehouses, while in the full coverage model, the aim is to cover all demand points. These approaches increase efficiency in post-disaster logistics processes, provide people with fast and easy access to the nearest warehouses, and ensure that relief supplies are delivered to the right points at the right time.

4.1.4 Median-Based Warehouse Location Model

The median-based warehouse location model is another common approach to the placement of warehouses. In this model, warehouse locations are optimized to minimize the distance to all demand points. Warehouses are placed in a way that provides the shortest distance to each demand point, which minimizes transportation costs and response times. The median-based model is used to ensure that post-disaster relief distribution is fast and effective. Correct location of warehouses increases the efficiency of logistics processes and ensures the most efficient use of resources.

4.1.5 P-Median Model

The P-median model is a mathematical optimization model frequently used in logistics and facility location problems. This model aims to locate the facilities closest to the demand points in a certain region. The P-median model is a very effective tool, especially in scenarios such as post-disaster aid distribution, healthcare location, and emergency management. Basically, it seeks a solution to minimize the distance between demand points in a region where a certain number of facilities need to be located. The model optimizes logistics processes, ensures more efficient use of resources and shortens response times. The P-median model, which can be solved with various optimization techniques, increases operational efficiency by offering the most appropriate solution to decision makers in such problems.

In our project, we aimed to use the P-median model to determine the strategic locations of warehouses in the disaster area in order to optimize the placement of relief materials at the points where they will be stored in advance after a natural disaster. This model optimizes the locations of warehouses to minimize the distance to the demand points and to ensure that people can reach the nearest warehouses in the fastest way. Thus, disaster victims will be able to access the nearest warehouses quickly and effectively.

Our project aims to create safe gathering areas for people in natural disaster situations. For this purpose, we focused on determining warehouse locations to ensure that people can easily reach the nearest warehouse after a disaster. We chose the Güzelbahçe district as the project area to facilitate settlement and analysis. Güzelbahçe is a district with 12 neighborhoods and occupies a smaller area compared to other districts, which facilitated our project process.

In this context, we focused on determining warehouse locations using the P-median model in order to optimize the location of relief centers after a natural disaster. First, we examined the population density of each neighborhood and evaluated the locations of existing collection areas. This data played an important role in our warehouse layout planning. Using the P-Median model, we initially analyzed the possibility of serving post-disaster demand points with 5 warehouses. However, as a result of our calculations, we observed that we could not cover 19,113 people with 5 warehouses. Therefore, we decided that the most appropriate solution would be to increase the number of warehouses and start with 8 warehouses.

```
                  KAHRAMAND~     MALTEPE  MUSTAFAKE~       YALI       YELKI

ATATURK                1.000
CAMLI                                          1.000
CELEBI                 1.000
KAHRAMANDERE           1.000
KUCUKKAYA                                      1.000
MALTEPE                            1.000
MUSTAFAKEMALPASA                               1.000
PAYAMLI                                        1.000
SITELER                1.000
YAKA                               1.000
YALI                                                       1.000
YELKI                                                                  1.000


----     74 VARIABLE Y.L  eger j noktasinda bir istasyon acilmis ise

KAHRAMANDERE     1.000,    MALTEPE         1.000,    MUSTAFAKEMALPASA 1.000
YALI             1.000,    YELKI           1.000


----     74 VARIABLE z.L                  =    19113.230
```

Figure 4.1.5.1 (Using 5 warehouses with P-Median Model)

```
                      ATATURK       CAMLI  KAHRAMAND~     MALTEPE  MUSTAFAKE~

ATATURK                 1.000
CAMLI                                1.000
CELEBI                                           1.000
KAHRAMANDERE                                     1.000
KUCUKKAYA               1.000
MALTEPE                                                      1.000
MUSTAFAKEMALPASA                                                         1.000
PAYAMLI                                                                  1.000
YAKA                                                         1.000

              +       SITELER        YALI       YELKI

SITELER                 1.000
YALI                                 1.000
YELKI                                            1.000


----     74 VARIABLE Y.L  eger j noktasinda bir istasyon acilmis ise

ATATURK          1.000,    CAMLI           1.000,    KAHRAMANDERE     1.000
MALTEPE          1.000,    MUSTAFAKEMALPASA 1.000,    SITELER          1.000
YALI             1.000,    YELKI           1.000


----     74 VARIABLE z.L                  =     5531.270
```

Figure 4.1.5.2 (Using 8 warehouses with P-Median Model)

By applying the P-Median model, we ensured that the 8 warehouse locations we determined were placed as close as possible to each neighborhood. The locations of these warehouses were as follows: Kahramandere, Atatürk, Çamlı, Maltepe, Mustafa Kemal, Siteler, Yalı and Yelki neighborhoods. The warehouse locations we determined for each neighborhood were optimized to serve these neighborhoods as well as neighboring neighborhoods. For example, the warehouse to be opened in Atatürk Neighborhood was planned to serve both Atatürk and Küçükkaya Neighborhoods; the warehouse to be opened in Kahramandere Neighborhood was planned to serve Kahramandere and Çelebiye.

With this solution, approximately 32,469 of the 38,000 people will be covered in 8 of the 12 neighborhoods in the Güzelbahçe district. This settlement plan aims to optimize the distribution of relief supplies and the rapid meeting of needs by allowing people to reach the nearest warehouse after a natural disaster.

The formulation of the model is as follows:

Sets:

i: Set of demand regions, $i \in$ {Atatürk, Camlı, Celebi...}

j: Regions where facilities can be established, $j \in$ {Atatürk, Camlı, Celebi...}

Parameters:

a_i: Demand (population) at point i

d_{ij}: Distance between demand region i and facility j (km)

p: Number of warehouses to be opened

S: Maximum distance constraint

Decision Variables

X_{ij} = {1, if neighborhood i is assigned to warehouse j.
0, otherwise.

Y_j = {1, if a warehouse is opened at point j.
0, otherwise.

Objective Function

The objective function of the P-median model is to minimize the total distance between demand points and facilities:

$$\text{Minimize } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i d_{ij} X_{ij} \tag{5.5.1}$$

Constraints:

$$\sum_{j \in J} X_{ij} = 1 \qquad \forall i \in I \tag{5.5.2}$$

$$X_{ij} \leq Y_j \qquad \forall i \in I, \ \forall j \in J \tag{5.5.3}$$

$$\sum_{j \in J} Y_j = P \tag{5.5.4}$$

$$d_{ij} X_{ij} \leq S \qquad \forall i \in I, \ \forall j \in J \tag{5.5.5}$$

$$X_{ij}, Y_j \in \{0,1\} \qquad \forall i \in I, \ \forall j \in J \tag{5.5.6}$$

The objective function (5.5.1) ensures that the total cost between demand points and facilities is minimized. Constraint (5.5.2) ensures that each demand point is assigned to only one facility, thus ensuring demand-fulfillment consistency. Constraint (5.5.3) states that only active facilities can be assigned demand. (5.5.4) optimizes resource usage by limiting the number of facilities to be opened to P. (5.5.5) ensures that the distance between demand points and facilities does not exceed the maximum distance limit (S). (5.5.6) indicates that the decision variables are binary (0 or 1).

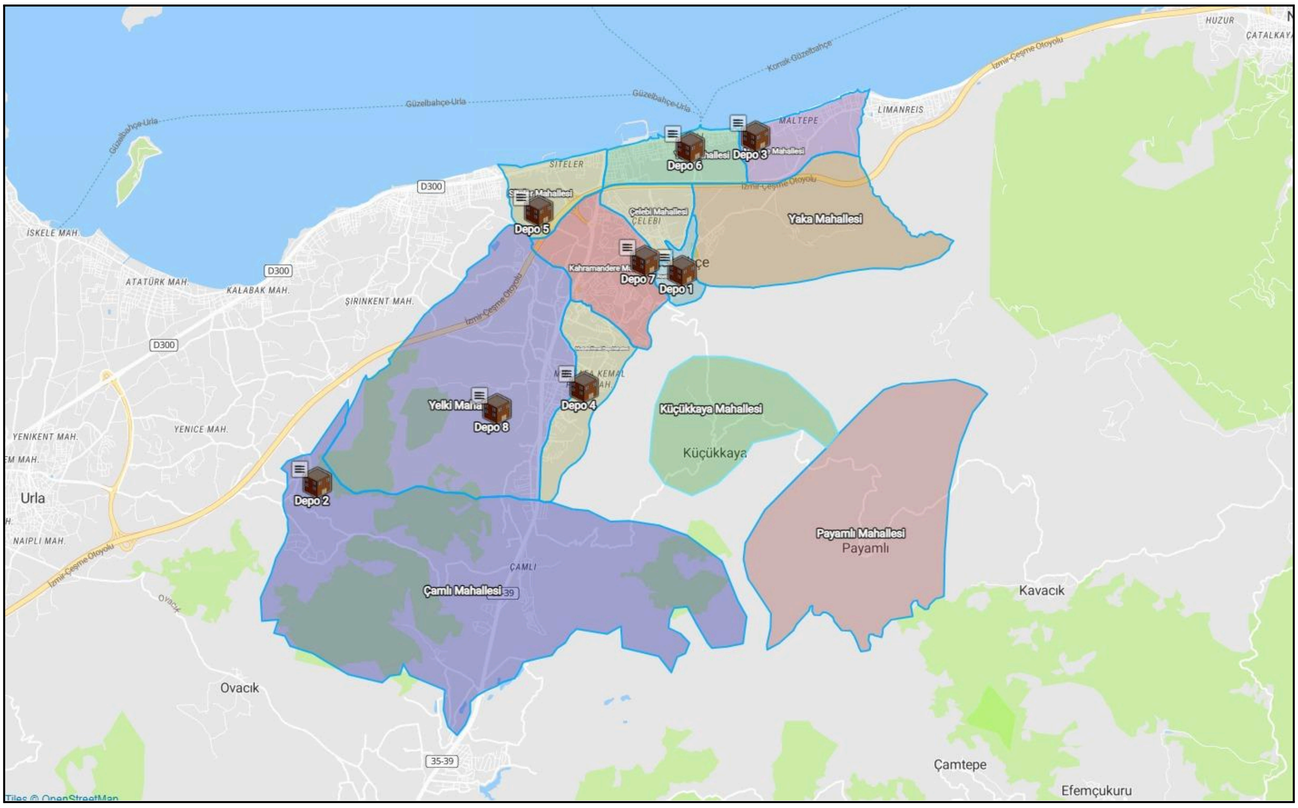

Figure 4.1.5.3 (Optimal location points of 8 warehouses with P-Median Model)

4.2 Integrated Data and Warehouse Management for Disaster

To enhance the sustainability and effectiveness of the system, the use of two separate databases, which form the foundation of a personalized and efficient disaster relief system, has been deemed appropriate. The first dataset plays a critical role in addressing the specific needs of affected populations during disasters by comprehensively collecting individual demographic, geographic, and medical data. The second important dataset is a database that allows for the efficient management of stock levels in warehouses. The analysis of these two datasets will help in understanding the key data components in disaster management and how these components can be effectively utilized in practice.

4.2.1 Personalized and Efficient Disaster Relief System

The information specified below will be collected from users via the mobile application and stored in the database.

4.2.1.1 User Demographics

The dataset includes attributes such as user name, surname, birthdate, gender, and height/weight, which are critical for profiling affected individuals. These details allow the system to:

- Prioritize aid based on vulnerability (e.g., children, elderly, or those with critical medical needs).
- Calculate specific resource needs, such as nutritional or medical requirements, tailored to individual body metrics.

The detailed calculation methodology for body metrics is provided below.

4.2.1.1.1 The Importance of Body Point Metrics in Disaster Relief

In disaster relief operations, efficiency and accuracy are paramount in ensuring that resources are appropriately allocated to meet the diverse needs of affected populations. Among the various metrics employed, the concept of a "body point" stands out as a composite measure that leverages physical dimensions such as chest and waist sizes to streamline aid distribution. By providing a weighted and practical score, the body point system exemplifies a data-driven approach to humanitarian logistics, aligning resources with individual requirements while minimizing waste.

The body point is calculated using a formula that combines chest size and waist size in a weighted manner, where chest size contributes 70% and waist size accounts for 30%. This prioritization reflects the primary influence of the chest size on overall body frame, while waist size adds supplementary detail. The formula is structured as follows:

Body Point=Rounded Value of ((Chest Size×0.7)+(Waist Size×0.3))

The chest size and waist size are derived based on the individual's height and weight using the following calculation:

Chest/Waist Size=Rounded Value of ((Weight (kg)+Height (cm))/2.5)

This method ensures that the body point accurately represents the physical build of an individual in a manner that is both scalable and easy to implement. For instance, if a person has a chest size of 100 cm and a waist size of 90 cm, their body point is calculated as:

Body Point$=((100\times0.7)+(90\times0.3))=97$

The simplicity of this calculation allows it to be automated, making it ideal for integration into a centralized data system that handles large datasets.

The utility of the body point metric lies in its ability to enhance personalization in resource allocation. In disaster scenarios, aid packages often include clothing, medical equipment, and other essential supplies. By leveraging the body point, relief organizations can ensure that distributed items, such as garments and mobility aids, are appropriately sized. For example, a person with a higher body point may require larger clothing or more robust medical support, whereas individuals with lower scores might need smaller or more specialized items.

Moreover, the weighted approach in the body point calculation underscores the importance of subtle data interpretation. Assigning a higher weight to the chest size acknowledges its greater impact on overall sizing decisions, while incorporating the waist size as a secondary factor adds precision without overcomplicating the metric. This balance between simplicity and accuracy makes the body point a versatile tool for both rapid response and long-term planning in disaster relief.

Beyond its practical applications, the body point system reflects the broader shift toward data-driven decision-making in humanitarian efforts. By quantifying physical attributes into actionable metrics, the system enables organizations to move away from generic resource allocation, instead fostering a more tailored and efficient approach. This not only enhances the effectiveness of aid distribution but also builds trust and satisfaction among beneficiaries, as their unique needs are visibly acknowledged and addressed.

The body point metric is a testament to the power of integrating analytics into disaster relief. Its weighted, composite structure provides a reliable and practical measure of physical dimensions, facilitating personalized aid distribution. By bridging the gap between data and action, the body point system sets a benchmark for precision and efficiency in humanitarian logistics, ensuring that relief efforts are not only impactful but also equitable.

4.2.1.2 Geographic Data

There are two different data fields for user location. The first is the current session information during the user's registration process, and the second is the current GPS location. The user GPS location field determines the exact coordinates of individuals and provides the following advantages:

- Accurate mapping of disaster impact zones.
- Optimized route planning for aid delivery using geographic clustering.
- Strategic placement of temporary depots between affected neighborhoods, minimizing transportation time.

4.2.1.3 Contact Information

Details like user phone numbers and emergency contact numbers ensure communication between aid coordinators and beneficiaries. This is vital for:

- Real-time updates on aid status.
- Emergency contact for priority medical evacuations or resource distribution.
- In case the individual cannot be reached, contact will be established with the emergency contact provided by the user.

4.2.1.4 Medical Data

Medical details, including chronic diseases, allergies, prescribed medications, and equivalents, are pivotal for personalized healthcare support:

- **Chronic Diseases:** Ensure that individuals with conditions like diabetes or asthma receive critical care supplies (e.g., insulin or inhalers) without delays.
- **Allergies:** Prevent the distribution of aid items (e.g., food or medications) that could trigger allergic reactions.
- **Medication Substitution:** The inclusion of medication equivalence data, such as **{Metformin: "850 mg once daily"}**, empowers an AI model to recommend safe alternatives when specific drugs are unavailable.

Detailed information about the AI model methodology is provided below.

4.2.1.4.1 Post-Disaster Drug Substitute Identification

The aim of the mobile application is to identify substitute drugs for medications reported by users that are unavailable in pharmaceutical warehouses after a disaster. To achieve this, an AI model trained with a relevant drug dataset will be employed, and the substitute drug information will be stored in a designated database. Therefore, the AI model to be developed needs to be trained and optimized specifically for this task.

Model Training and Fine-Tuning

For the AI model intended to identify drug substitutes, the use of the Fine-Tuning method has been deemed appropriate. Fine-Tuning is a transfer learning-based approach that optimizes the model for a specific task. This process begins with the Pre-Training phase, where the model is trained on a broad dataset. In the Pre-Training phase, neural networks are used to help the model understand the fundamental structure of the language, grammar rules, and semantic relationships. Once this phase is completed, the model reaches a level where it can comprehend general language structures and concepts.

Transition to Specific Training

After the Pre-Training phase, the model must be retrained on a specific domain, in this study drugs and their compositions. This phase, referred to as Fine-Tuning, is carried out using datasets tailored to the field. During Fine-Tuning, correctly configuring hyperparameter settings is crucial for the model's success.

Hyperparameter Settings

- **Tuning Epochs:** Specifies how many times the model will iterate over the training dataset. A high value may cause the model to overfit the dataset, resulting in poor performance on real-world data. Overfitting occurs when the model becomes excessively tailored to the training data.
- **Learning Rate Multiplier:** Indicates the learning speed of the model. A very low learning rate might require numerous epochs for the model to reach good performance, while a very high rate can hinder learning, increase loss values, and raise the risk of NaN (not a number) values.
- **Batch Size:** Refers to the number of samples processed simultaneously during training. While a high batch size increases memory usage, a low batch size may prolong training time and introduce inconsistencies.

Applying these processes correctly will help the model achieve high efficiency and accuracy in identifying drug substitutes.

4.2.1.5 Clothing and Accessibility Needs

The dataset highlights special clothing or accessibility requirements such as wheelchairs or prosthetic legs. This is essential for:

- Delivering customized aid items, like mobility aids or size-specific clothing.
- Ensuring that the system accounts for logistical challenges related to individuals with mobility constraints.

4.2.1.6 Inventory and Demand Forecasting

The body size, shoe size, chest size, waist size, and other metrics allow for precise stock management in warehouses. These parameters ensure:

- Optimal inventory levels tailored to the population's needs, reducing wastage.
- Accurate forecasting of aid item requirements, such as clothing or medical supplies.

4.2.1.7 Data Processing and Consent

Fields like data processing approval and contract approval are vital for ethical data handling and compliance with privacy laws. This ensures:

- Secure management of sensitive personal data.
- Transparency and trust between the aid system and its beneficiaries.

4.2.2 Warehouse Stock Management

The warehouse information will be integrated with the number of users and their needs in the region, as indicated in the application. Warehouse locations will be accessed using the methodology outlined in P-median model which is explained in 4.1 and the relevant datasets will be provided as specified below.

4.2.2.1 Product Management in Warehouses

Storing product names along with other relevant information (such as category, quantity, status) in a database provides significant convenience in inventory management, quick access, and traceability. Storing product names and other related information in the database enables accurate tracking of inventory. This way, stock levels can be monitored in real-time, and issues like inventory shortages or excess can be identified in advance.

Additionally, regular reporting can be done on the identity and status of each product in the system. Information such as the category of products and their current stock status allows quick access to the required products in warehouses and optimizes distribution processes. Such an arrangement increases the efficiency of aid operations, while making storage and distribution processes more transparent. Since product information is regularly updated in the database, stock movements and product data can be consistently tracked, ensuring that the correct resources are available in the right place and in the right amount during crisis times.

4.2.2.1.1 Product Categorization

Categorizing the products in the warehouses is critical for quick access, inventory management, identifying priority needs, traceability, and reporting. The system includes four distinct categories:

- **Clothing**: This category includes various types of clothing essential for survivors after a disaster, such as coats, jackets, shirts, and pants.
- **Footwear**: This category includes shoes, boots, and other types of footwear necessary for disaster victims.
- **Special Needs**: This category contains products designed to meet the specific needs of vulnerable groups. These may include items such as diapers, mobility aids, and specialized medical equipment for infants, elderly people, or individuals with disabilities.
- **Medications**: The medications category includes essential drugs and medical supplies needed to treat injuries, diseases, or infections that may arise after a disaster.

4.2.2.2 Warehouse Location Management

In emergency situations or crises, inventory procurement or delivery is of great importance. In such situations, products may need to be quickly and accurately obtained or delivered from warehouses. Users or aid organizations must be able to reach the relevant warehouse locations as quickly as possible. This can be achieved by properly mapping the warehouses and monitoring inventory in real-time.

Stock levels for specific needs have been predefined, and it has been determined which warehouse location contains which products and the current stock levels of these products. This information can be accessed quickly in an emergency. Especially during crises, having the inventory in the correct quantities and at the right locations ensures that aid processes are managed swiftly and efficiently.

Therefore, an effective inventory management system should establish a strong connection between warehouse locations, inventory status, and needs. Additionally, the latitude and longitude information of the warehouses, when visualized on a map, enables managers or decision-makers to quickly identify the nearest warehouse locations and the required products. This speeds up material procurement and ensures a more efficient aid operation without disruptions in the supply chain.

4.3 AI-Driven Multi-Depot Route Optimization Framework for Disaster Response and Resource Allocation

4.3.1 Data Collection and Preprocessing

For AI-enabled route optimization, multi-source data will be acquired and processed to create an effective framework. The multi-source data sources include:

- **GPS Data**: Real time location information of users provided through GPS modules installed on mobile devices. It shall be used for determining the origins for the users and proximity to depots. GPS data accuracy will be enhanced by using error correction techniques like DGPS to reduce inconsistencies in the location of users.
- **Inventory Data**: Real-time stock availability at depots is updated with APIs linked to the inventory management system. This shall be fetched at regular intervals so as to have users with updated information on item availability, therefore reducing the chances of the user being routed to a depot that has low or no stock of the items required.
- **Road Network Data**: Massive data about the road conditions such as congestion, blockage, and distance covered will be provided via APIs like Google Maps and OpenStreetMap. Further, the road network data will be complemented with historical traffic flow that allows the system to predict the congestion in advance and suggest an alternative route.
- **Environmental Data**: Road/ Weather condition, warnings of a possible natural disaster, from meteorological services. This might contain specific details about flood or landslide

zones, among others. Integration into route optimization methods needs to take these to enhance the user's safety.

Preprocessing: Preprocessed cleaning and normalization for feeding them in an encoded compatible form. Examples may be :

- Perform the interpolation of missing values in inventory or street network data using the k-NN method or mean imputation approach in any statistical imputation. Outliers are usually due to signal interference in GPS data; detect and correct them using clustering, for example, with DBSCAN.
- The road network will be represented as a weighted directed graph. The nodes will be the intersections, and the roads will be the edges between these nodes. The weight of the edges shall be determined by a composite scoring of traffic density, road quality, and predicted delays. This representation shall be the very base on which all routing computation is done.
- To guarantee the robustness of the pre-processing step, real-time validation techniques will be put in place. For example, inventory data inconsistencies will result in warnings, which would raise an alarm for the depot managers to correct it immediately. Similarly, the anomalies in traffic data-odd sudden closure of a road-will be detected, validated with secondary data sources, and then fed into the model.

4.3.2 Model Selection

Model selection is an important step in the proposed system, since it guarantees a balance between computational efficiency and real-time adaptability. It gives a base by the combination of classical graph-based algorithms with modern methods of reinforcement learning to handle various challenges related to route optimization.

- **Graph algorithms**: Dijkstra's algorithm and A*, will be implemented as the baseline computation of routes, since both are really efficient in static graph environments. On one hand, Dijkstra's Algorithm is effective at finding the shortest path between nodes in a weighted graph. A*, on the other hand, incorporates heuristic functions to guide the exploration and is often considerably faster in many cases. These algorithms are the basis for primary route calculation and hence can form a sound basis on which other factors may be integrated dynamically.
- **Reinforcement Learning (RL)**: In this respect, for adapting to dynamic road conditions, one will implement a Deep Q-Learning Network-DQN. The model of RL learns the

optimal policies through trial and error; this simulates the movements of agents across a graph representation of the environment. Unlike other conventional algorithms, RL does well with real-time updates, say, a road suddenly getting closed or changes in traffic density. The Q-value function in the DQN model will approximate a neural network so that it can make complex routing decisions efficiently.

4.3.2.1 Advanced Model Selection Techniques

1. Transfer Learning: Leverage transfer learning for fast convergence: Pretrain on smaller subgraphs and scale to the full size of data to save time for optimal performance.

2. Hybrid Integration: Graph algorithms and reinforcement learning will be integrated dynamically into the system. For example, graph algorithms can statically compute the routes, while the RL system dynamically adjusts those at runtime, taking as input in real-time over stock or road disruption.

3. Design of Cost Function: Design one composite cost function for route assessment. It will take into consideration a number of factors like travel time, distance, density of traffic, and road quality. These factors will be given appropriate weights to reflect user priority so that the system yields routing solutions optimized according to users' needs.

4.3.2.2 Training and Validation

This work describes the training of the DQN model on a synthetic dataset enriched by traffic patterns, inventory scenarios, and road blockages. Using advanced simulation tools, this research will devise scenarios related to peak-hour traffic in an urban environment or supply shortage conditions at depots. Further, the model has to be tested on real grounds using real-time data obtained from APIs like Google Maps or OpenStreetMap for its accuracy and reliability.

Further robustness could be achieved by resorting to an ensemble learning approach: multiple RL models may be trained on different data subsets. This would surely enhance the generalization capability of an ensemble strategy and reduce overfitting risks when complex routing environments are faced.

It can exploit synergy in achieving the best of both worlds: computational efficiency and real-time adaptability by merging traditional and state-of-the-art methods. This hybrid model will enable any user to be always offered optimal routing advice even under dynamic, quite unpredictable conditions.

4.3.3 Multi-Depot Routing Strategy

The multi-depot routing strategy is demonstrated in the İzmir Güzelbahçe district, which is the focus area of this project. Eight depots have been strategically selected within the district in view of population density, easy access, and suitability to serve as gathering points. The map below, Figure X, shows the divisions of the district and the location of the depots, including how such factors drive the routing strategy.

Consider a scenario: A user from Yelki needs a "Thermal Blanket" and "Nutritious Food Pack." First, the system enquiries with the inventory database for the location of such depots that have these items. Let's say Depot 7 has the blankets and Depot 1 has the food packs. It then probes all possible routes as in the map below, Figure 4.4.1 with the routing algorithm based on road conditions, traffic density, and distance to arrive at the best route.

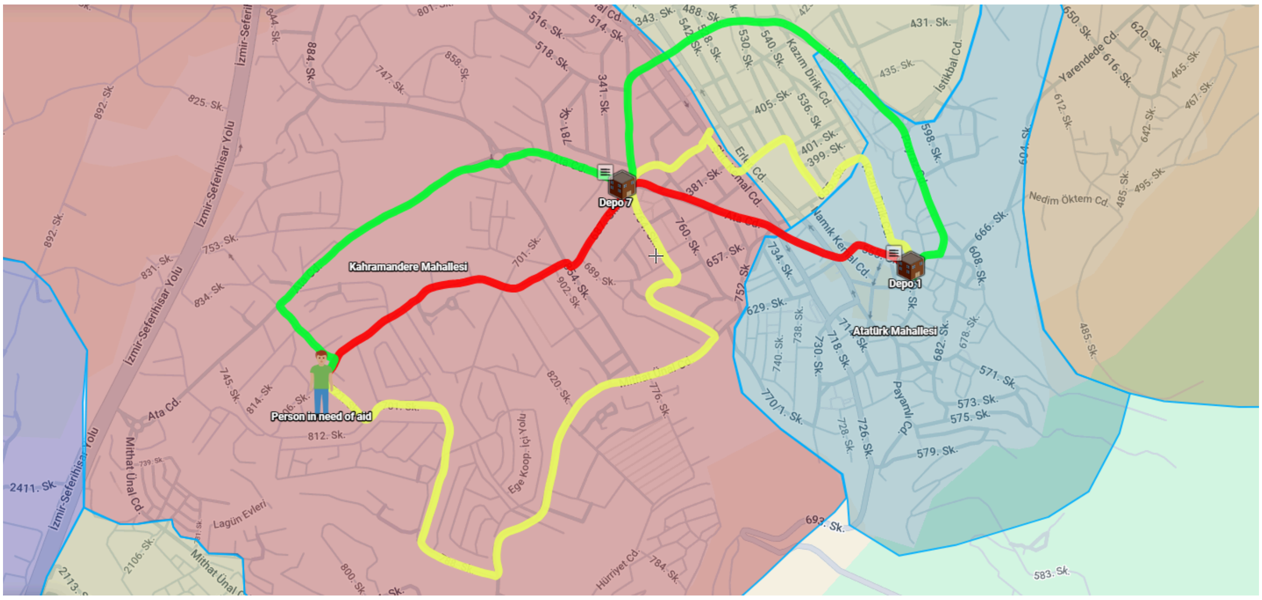

Figure 4.3.3.1 (Possible Routes for Scenario)

The system might recommend the following optimal route:

1. Go to Depot 7 to pick up the blankets by using a primary road with very little traffic flow.
2. Head toward Depot 1, adjusting the route dynamically in case of any unforeseen blockages.

Real-time traffic updates will enable the recalibration of the route through reinforcement learning to ensure that the user navigates the most efficient path to meet their needs with the least amount of time and effort. The system can dynamically incorporate new data-such as temporary depot closures or sudden surges in traffic-into its routing calculations.

Handling requests across multiple depots involves solving an advanced Vehicle Routing Problem. The steps include:

- A query-based filter on the inventory database to identify all depots holding the requested items.
- Geographically and item-priority-based clustering of depots by the K-means technique so that depots are categorized in an efficient manner for routing.
- A Genetic Algorithm can be applied to find the best possible sequence of the depot visit. It needs to define the cost function, balancing travel time and distance, with the urgency of the requested items. Each chromosome here represents one possible route to which the algorithm encodes potential depot visit sequences into chromosomes. The GA evaluates these sequences using a fitness function, aiming for solutions with the minimum total cost of travel while ensuring that all depots with required items are in the route. The GA improves this routing solution through iterative processes of selection, crossover, and mutation, converging toward an optimal or near-optimal sequence of depot visits.
- Integrating the optimized sequence into the RL model, with the possibility of dynamic reoptimization in response to real-time changes in road or depot conditions. This makes the solution adaptive and robust against changing scenarios.

4.3.4 Implementation

Implementation of the conceptual system will include the following:

- Backend Development: Python is to be selected because it has a wide range of libraries. For AI models, TensorFlow and PyTorch will be used; NetworkX for graph-based calculations; and for solving combinatorial optimization problems, Google OR-Tools will be availed. As a matter of fact, they provide the technical basis for all algorithmic/computational operations.
- Frontend Interface: The user interface can be designed using Flutter for cross-platform compatibility and user friendliness. It will provide an interface to the user for inputting their requirements and observing optimized routes in real time.
- Cloud Integration: It can be done with any cloud service provider, for example, Amazon Web Services, which would serve to host the backend in an easily scalable way. It will enable real-time updates through WebSocket connections so that user interactions become really smooth.

4.3.5 Validation and Testing

The reliability, robustness, and efficiency of the proposed system have to be guaranteed with validation and testing. In this respect, all components of the system will be tested in a structured manner under various scenarios.

4.3.5.1 Simulation Environments

Firstly, the system will be validated in a controlled simulated environment before its actual deployment. The simulations will be based on real-world scenarios, including:

- High traffic density during peak hours.
- Road closure or detours at short notice due to construction work or accidents
- Stock in depots reaching a stock-out level.
- Flooding and/or a blizzard.

Advanced traffic and routing would include the use of the SUMO and AnyLogic software to make these traffic and routing scenarios realistic. It provides the functionality to model even the complex set of interactions between vehicles, users, and depot locations, which offers a strong platform to test route optimization algorithms.

4.3.5.2 Performance Metrics

KPIs are to be defined for system effectiveness. The metrics that come within its ambit are:

- **Accuracy**: The ratio of the number of routes very near to the optimum.
- **Efficiency**: Time taken to travel, averaged out, as compared to baseline routes.
- **Adaptability**: How efficiently the system updates routes within 1-2 seconds in cases of dynamic updates.
- **Scalability**: System performance when faced with up to 10,000 concurrent requests.

4.3.5.3 Testing Phases

1. **Unit Testing**: The system will be tested for correctness and reliability on the individual components, like the GPS module, inventory API, and traffic data parser. Automated testing frameworks for Python, such as pytest, will be employed in the testing process.
2. **Integration Testing**: It will test the interoperability between different components for the proper flow of data. In particular, the integration of traffic data with route optimization algorithms will be tested.
3. **Stress Testing**: It involves testing under extreme conditions, including very high loads of users and many updates on dynamically changed data, among other aspects. Based on

these analyses, several bottlenecks will emerge on the basis of which appropriate optimization needs to be targeted.

4. **User Testing**: In the case of system testing by users in a small pilot deployment, test results regarding user friendliness, accuracy, and prompt responses will also be aggregated for further application refinement.

4.3.5.4 Dynamic Update Validation

Testing will also include scenarios where sudden changes on the road or at the depot inventories take place. It would be expected that the system recalibrates its routes in seconds with the view of maintaining accuracy and minimizing user disruption.

Error Handling and Redundancy: Security validation shall create preparation in case something goes wrong, such as:

- API downtime will trigger fallback systems to use cached or secondary data sources.
- GPS signal loss will be mitigated by switching to alternate localization methods, such as Wi-Fi triangulation.

Iterative Refinement: The performed iterative refinements will involve algorithms and system architecture, which will be derived from test results. If tests of stress reveal latency, for example, then caching mechanisms or parallel processing techniques shall be used to boost performance.

This comprehensive validation and testing methodology will ensure that the system provides a robust, user-oriented solution that can meet a wide range of operational challenges.

4.3.6 Ethical Considerations

The system will have priorities regarding user data privacy, anonymizing GPS and other personal data so that individuals cannot be identified throughout different steps of data gathering or processing. Secure encryption protocols should be applied at all instances of data transmission and storage: AES-256 should preferably be used to avoid related risks of unauthorized access and breaches. The system will also be compatible with international data protection regulations like GDPR, thus ensuring adherence to high standards of privacy.

The system will embed fairness and transparency in AI decision-making. In ensuring fairness to the outcomes for all users, bias detection will be set up during the training phase of the system. Periodic auditing of the AI models will be carried out to avoid biases or mistakes that are unintended, which secures trust in the system even further. This falls in line with measures for ethical use, promoting accountability and fairness.

In disaster scenarios, these will also extend to prioritization protocols to make sure vulnerable populations receive a fair share of whatever is available. It would thus provide mechanisms for flagging high-priority cases like medical emergencies without compromising the fairness for other users.
In the proposed system, user data privacy will be paramount. GPS and personal data shall be anonymized before processing; on top of that, secure encryption protocols will be applied to all communications. Ethical guidelines on designing AI guarantee fairness and transparency of decision-making processes.

5. Results and Discussion

The Results and Discussion section will be structured based on the implementation of the P-median model and provide valuable insights into the optimal placement of aid depots within the Güzelbahçe district, which has a population of 38,000 distributed across 12 neighborhoods, datasets and route optimization outcomes.

5.1 P-Median Model Analysis for Optimal Aid Depot Placement in Güzelbahçe District

The implementation of the P-median model provided valuable insights into the optimal placement of aid depots within the Güzelbahçe district, which has a population of 38,000 distributed across 12 neighborhoods. The primary results and discussions arising from this analysis include:

5.1.1 Determined Locations for Aid Depots

- Based on the P-median model's analysis, the optimal strategy suggested establishing a minimum of **8 aid depots** within targeted neighborhoods to ensure effective reach and accessibility during emergencies.
- The selected neighborhoods for depot establishment include:
    - **Atatürk**
    - **Çamlı**
    - **Kahramandere**
    - **Maltepe**
    - **Mustafa Kemal**

- **Siteler**
- **Yalı**
- **Yelki**

The cost comparison between container and reinforced concrete construction for a 100 square meter structure is shown in the table. These figures do not include land costs.

Reinforced Concrete Structure	**Cost (TL)**	**Container Structure**	**Cost (TL)**
Foundation and Flooring Work	350,000	Container Body	660.000
Wall Work (Gas concrete)	50,000	Floor Covering (Tiles or Laminate)	30.000
Roof Work (Sandwich panel roof)	75,000	Wall Covering (Panel or plasterboard)	60.000
Doors and Windows Installation	40,000	Doors and Windows Installation	40.000
Electrical and Plumbing Systems	60,000	Electrical and Plumbing Systems	60.000
Kitchen Area (Countertop, cabinets, sink)	80,000	Kitchen Area (Countertop, cabinets, sink)	80.000
Restroom Work	30,000	Restroom Work	30.000
Solar Panel System	140,000	Solar Panel System	140.000
Total	**825,000**	**Total**	**1.100.000TL**

Table 5.1.1.1 (comparison between container and reinforced concrete construction)

As a result of these calculations, it was decided that the structure to be built would be reinforced concrete, as it is less costly and more durable than containers. Our warehouse has 4 stock rooms, 1 kitchen, 1 toilet for men and women, and 1 dressing room.

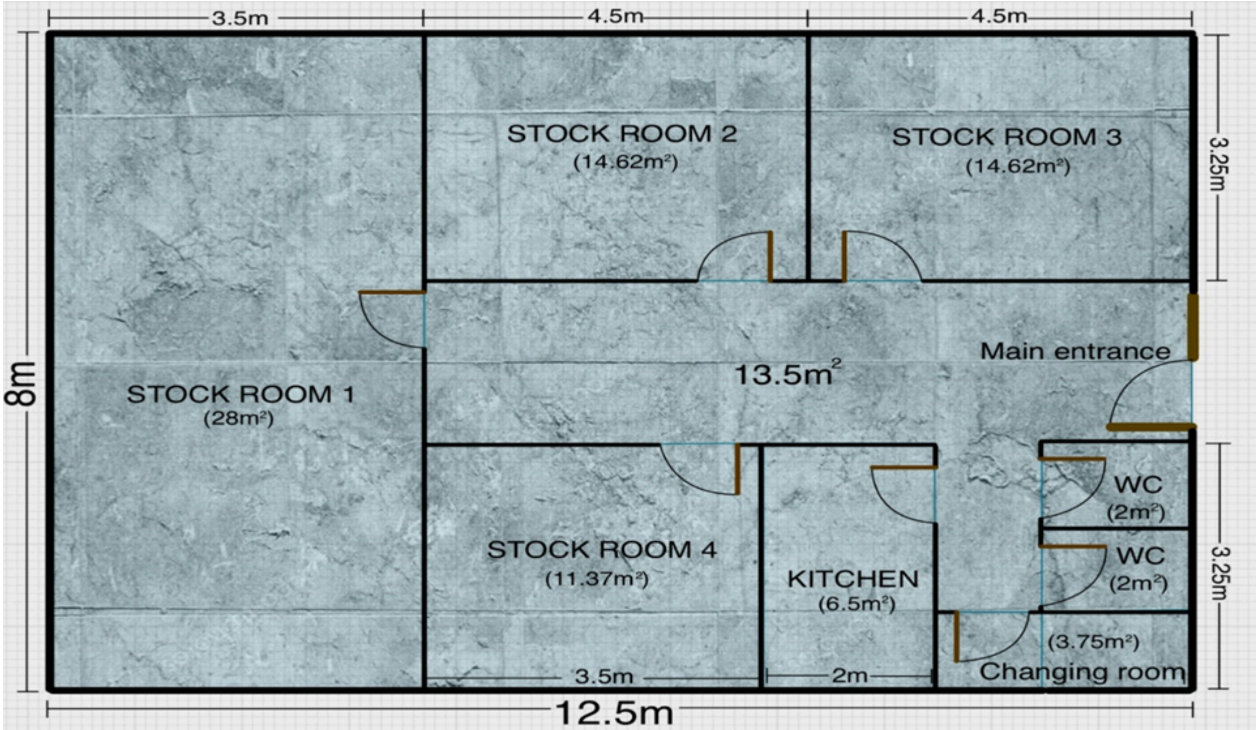

Figure 5.1.1.1 (optimal placement of aid depots)

5.1.1.1 Coverage and Efficiency

- Each depot will cover a **5 km radius**, which ensures that a significant portion of the population remains within access to aid facilities in case of a disaster.
- The distribution of depots across these neighborhoods allows for a more balanced and efficient response to emergencies, reducing the average distance residents need to travel to access aid.

5.1.1.2 Cost and Resource Optimization

- The analysis highlighted potential cost savings associated with optimal depot placement, minimizing transportation and logistical costs involved in delivering aid.
- By strategically positioning depots close to high-density population areas, resource allocation can be optimized, ensuring faster response times during emergencies.

5.1.1.3 Stakeholder Engagement

- Collaboration with local stakeholders, including municipal authorities and community leaders, is crucial for the successful implementation of this plan. Engagement efforts can enhance community buy-in and ensure the depots are tailored to meet the specific needs and concerns of residents.

5.1.1.4 Limitations and Future Research

- While the results indicate a robust model for establishing aid depots, limitations such as the availability of land, potential bureaucratic hurdles, and changes in population dynamics should be considered.
- Future research may focus on integrating broader data sets, including socioeconomic factors and historical emergency response effectiveness, as well as exploring different models or scenarios for optimal aid distribution.

5.2 Artificial Intelligence Integration for Drug Alternatives and Disaster Relief Resource Management

After the earthquake, artificial intelligence support has been utilized in finding drug alternatives. At the point where the project is practically implemented, the goal was to create an artificial intelligence model from scratch, from the pre-training phase to the fine-tuning phase, to meet the required specifications. However, in order to demonstrate feasibility, the existing Gemini model has been used. The applicability of artificial intelligence in this context has been tested using the "Tune a Model" feature provided by Google AI Studio which is free. For this purpose, datasets related to drug names, active ingredients, and usage areas were researched, and then a CSV file named Medicine_Details.csv was found. This file, with more than 10,000 rows, is highly suitable for training the model, as it meets the minimum data limit of 500 items. A snippet from this file is shown in Figure 5.1.

	A	B	C	D	E	F	G	H	I
1	Medicine Name	Composition	Uses	Side_effects	Image URL	Manufacturer	Excellent Review %	Average Review %	Poor Review %
2	Avastin 400mg Injection	Bevacizumab (400mg)	Cancer of colon and rectum Non-small cell lung	Rectal bleeding	https://onemg.gumlet.io/l	Roche Products India Pvt Ltd	22	56	22
3	Augmentin 625 Duo Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	Treatment of Bacterial infections	Vomiting Nausea	https://onemg.gumlet.io/l	Glaxo SmithKline Pharmaceutica	47	35	18
4	Azithral 500 Tablet	Azithromycin (500mg)	Treatment of Bacterial infections	Nausea Abdomi	https://onemg.gumlet.io/l	Alembic Pharmaceuticals Ltd	39	40	21
5	Ascoril LS Syrup	Ambroxol (30mg/5ml) + Levosalbutamol (1mg/5ml) + Guaifenesin (50mg/5ml)	Treatment of Cough with mucus	Nausea Vomiting	https://onemg.gumlet.io/l	Glenmark Pharmaceuticals Ltd	24	41	35
6	Aciloc 150 Tablet	Ranitidine (150mg)	Treatment of Gastroesophageal reflux disease (	Headache Diarr	https://onemg.gumlet.io/l	Cadila Pharmaceuticals Ltd	34	37	29
7	Allegra 120mg Tablet	Fexofenadine (120mg)	Treatment of Sneezing and runny nose due to al	Headache Drow	https://onemg.gumlet.io/l	Sanofi India Ltd	35	42	23
8	Avil 25 Tablet	Pheniramine (25mg)	Treatment of Allergic conditionsTreatment of Re	Sedation	https://onemg.gumlet.io/l	Sanofi India Ltd	40	34	26
9	Aricep 5 Tablet	Donepezil (5mg)	Alzheimer's disease	Common cold U	https://onemg.gumlet.io/l	Eisai Pharmaceuticals India Pvt L	43	28	29
10	Amoxyclav 625 Tablet	Amoxycillin (500mg) + Clavulanic Acid (125mg)	Treatment of Bacterial infections	Vomiting Nausea	https://onemg.gumlet.io/l	Abbott	36	43	21
11	Atarax 25mg Tablet	Hydroxyzine (25mg)	Treatment of AnxietyTreatment of Skin condition	Sedation Nausea	https://onemg.gumlet.io/l	Dr Reddy's Laboratories Ltd	35	41	24

Figure 5.2.1 (A Part From the Medicine_Details.csv File)

As shown in Figure 5.1, the "Medicine Name" column was assigned as the input column for the model, while the "Composition," "Uses," "Side Effects," "Image URL," and "Manufacturer" columns were assigned as the output. The model then underwent training, and after 5 hours and 29 minutes of training, the model was fine-tuned specifically for drug alternatives. This training consisted of 11,826 examples, with the hyperparameter values set as follows: Epochs: 5, Batch size: 16, Learning rate: 0.0002. After the model training, the outputs can be seen in Figure 5.2. As shown in the graph, the model underwent linear training and did not experience any extreme jumps, either too high or too low. This indicates that the model continued to learn in a stable manner.

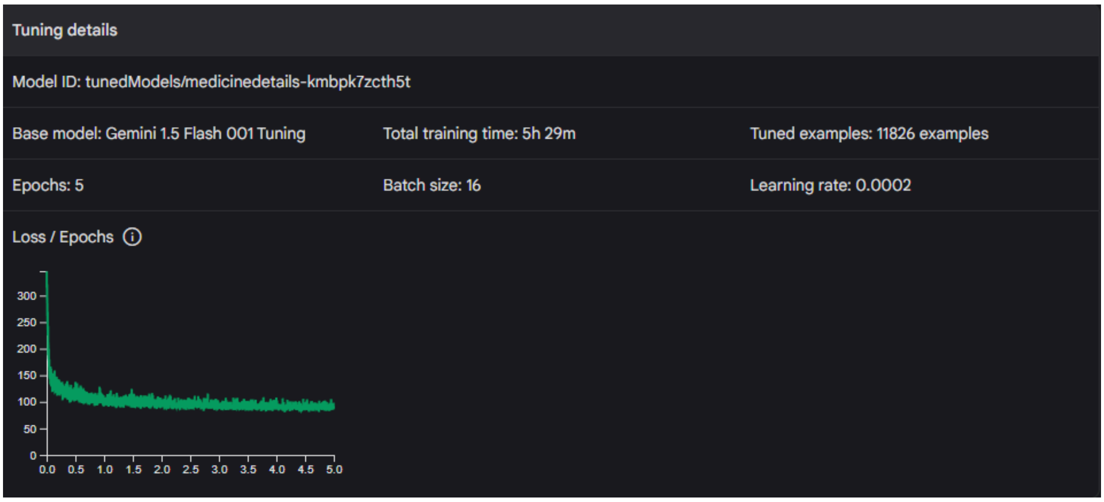

Figure 5.2.2 (Medicine_Details Model Results After Training)

Upon completion of the model's training, an application was developed using Python, JavaScript, HTML, and CSS technologies to serve this purpose, utilizing the API key obtained from the model. Details of this application are provided in the Appendix section. Figures 5.3 present two different drug substitute information obtained from the application.

In the first step, the drug whose substitute is to be found will be entered into the relevant text field. Then, the prompt "List drugs that have the same composition and uses as {medicine_name}. If {medicine_name} is not found, write 'the medicine is not found.' This is not medical advice. Only list the names of medicines with the same composition and uses as {medicine_name}." will be used to list substitute drugs based on the dataset trained by the model.

An important point to note here is that this application was developed to test the applicability of the model and the effectiveness of AI support. This revision clarifies that Google AI Studio was used for testing. In real-life implementation of the project, the large language model (LLM) will be specifically created for this project, and the relevant dataset will be trained using the Fine Tuning method on the generated LLM.

In real-life scenarios, the drug information entered by the user during registration will be sent to the specialized AI model. Subsequently, the listed substitute drugs will be saved in the relevant database.

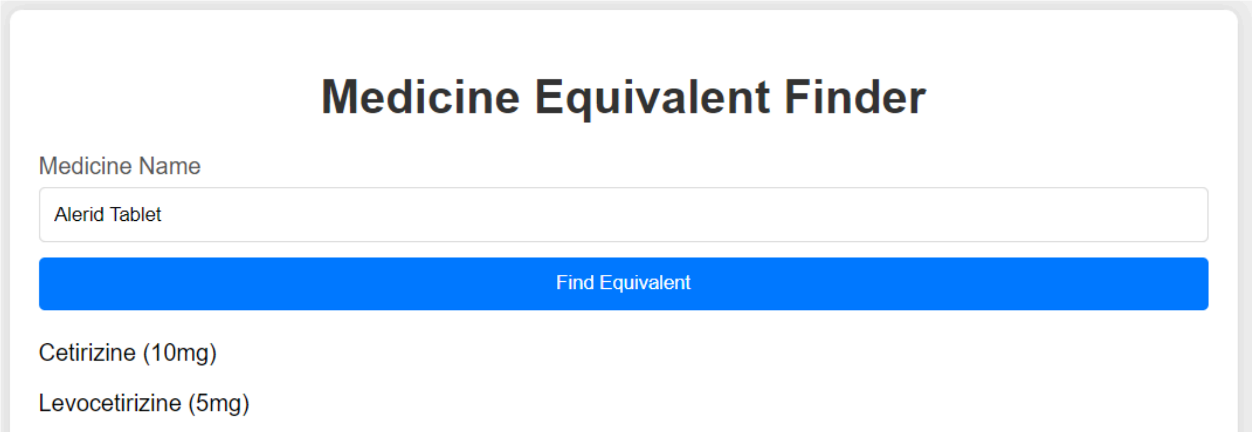

Figure 5.2.3 (Equivalent Medicine List of Alerid Tablet Medicine)

In Figure 5.3, the drug Alerid Tablet is listed along with its equivalent drugs, Cetirizine and Levocetirizine. Cetirizine is the active ingredient of Alerid Tablet and is an antihistamine used for treating allergic reactions. Levocetirizine is a purer form of Cetirizine and is also used for allergy treatment in the same way.

5.2.1 Dataset Management and Example Data

Below are table details and sample data for two different databases: **post_disaster_app** and **post_disaster_stock**. Both databases are integrated with each other. The goal is to combine user data coming from the application with related data from the stock database to analyze user needs and match them with available resources. The first database contains user information such as personal details, medication information, body measurements, and locations, while the second database holds stock information and product categories.

The process is designed to match relevant stock items based on users' measurements or need profiles. This matching is achieved by using common columns (e.g., body measurements, the number of users requiring specific measurements, or stock information in the second database). As a result, a need list is created for each user and is then linked to the current stock status to optimize the material supply process.

This approach leverages database relationships and SQL queries to integrate data from different databases while also incorporating data processing at the software layer when necessary.

5.2.1.1 post_disaster_app Database

This database represents a relational database where user information, the data provided during the registration process, and permissions are systematically organized. The **post_disaster_app** database is linked to the mobile application during the user registration process and stores the data entered by the user.

It must be noticed that if the user does not accept certain terms, such as the one stating that their personal data will be stored in the database (as specified in Figure 5.2.1.7), their registration to the application cannot be completed. Due to the fact that the provided information will be used to improve post-disaster aid coordination, enhance warehouse management, and facilitate efficient resource distribution.

The **post_disaster_app** database consists of seven data tables which are users_table, contact_info_table, location_table, health_info_table, medication_info_table, clothing_info_table, and approval_info_table. Detailed information about these tables will be given below.

←T→				user_id	user_name	user_surname	user_birthdate	user_gender	password_hash
☐	Düzenle	Kopyala	Sil	6	Ahmet	Yılmaz	1995-05-12	Male	abcd1234hash
☐	Düzenle	Kopyala	Sil	7	Ayşe	Demir	1988-03-22	Female	hash789xyz
☐	Düzenle	Kopyala	Sil	8	Mehmet	Çelik	2000-11-15	Male	1234secure5678
☐	Düzenle	Kopyala	Sil	9	Elif	Aksoy	1997-02-08	Female	elifpwd123hash
☐	Düzenle	Kopyala	Sil	10	Fatih	Korkmaz	1985-07-19	Male	fatihsecurehash
☐	Düzenle	Kopyala	Sil	11	Cem	Güler	1992-08-05	Male	cemhash123
☐	Düzenle	Kopyala	Sil	12	Merve	Şahin	1999-01-15	Male	merve987secure
☐	Düzenle	Kopyala	Sil	13	Hasan	Kaya	1987-03-19	Male	hasanxyz567
☐	Düzenle	Kopyala	Sil	14	Ceren	Akbaş	1994-06-28	Female	cerensuperhash
☐	Düzenle	Kopyala	Sil	15	Ömer	Yıldız	1990-09-10	Male	omersecret123

Figure 5.2.1.1.1 (users_table)

The **users_table** is a table that stores the basic information of users. This table includes the **user_id** field, which is a unique and auto-incremented identifier for each user. Users' first and last names are stored in the **user_name** and **user_surname** fields, respectively, while their date of birth is kept in the **user_birthdate** field. The user's gender is specified in the **user_gender** field. For security purposes, user passwords are not stored directly; instead, their hashed versions are saved in the **password_hash** field. This table uses the user_id field as the primary key to establish relationships with user-related information in other tables.

←T→				id	user_phone	user_mail	user_id	user_emergency_contact_phone
☐	Düzenle	Kopyala	Sil	12	5051234567	ahmet@example.com	6	5351234567
☐	Düzenle	Kopyala	Sil	13	5369876543	ayse@example.com	6	NULL
☐	Düzenle	Kopyala	Sil	14	5079876523	mehmet@example.com	8	5559876523
☐	Düzenle	Kopyala	Sil	15	5559012345	elif@example.com	9	5239012345
☐	Düzenle	Kopyala	Sil	16	5340098765	fatih@example.com	10	5120098765
☐	Düzenle	Kopyala	Sil	17	5431239876	cem@example.com	11	NULL
☐	Düzenle	Kopyala	Sil	18	5329876589	merve@example.com	12	5339876589
☐	Düzenle	Kopyala	Sil	19	5078912345	hasan@example.com	13	5068912345
☐	Düzenle	Kopyala	Sil	20	5567893456	ceren@example.com	14	5467893456
☐	Düzenle	Kopyala	Sil	21	5345678901	omer@example.com	15	5145678901

Figure 5.2.1.1.2 (contact_info_table)

The **contact_info_table** is a table that contains users' contact information. The user's phone number is stored in the **user_phone** field, their email address in the **user_mail** field, and their emergency contact phone number in the **user_emergency_contact_phone** field. Additionally, the user's address is stored in the **user_address** field. This table is linked to the **users_table** through the **user_id** field. This relationship ensures that each user's contact information can be identified and facilitates communication in case of emergencies.

←T→				id	user_id	user_address	user_gps_location
☐	Düzenle	Kopyala	Sil	7	6	Ankara, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	8	7	İstanbul, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	9	8	İzmir, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	10	9	Bursa, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	11	10	Antalya, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	12	11	Konya, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	13	12	Kayseri, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	14	13	Samsun, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	15	14	Trabzon, Türkiye	[GEOMETRY - 25 B]
☐	Düzenle	Kopyala	Sil	16	15	Gaziantep, Türkiye	[GEOMETRY - 25 B]

Figure 5.2.1.1.3 (location_table)

The **location_table** is a table that contains users' address details. The user's address is stored in the **user_address** field, while their real-time geographic location (or the last known location if GPS is turned off) is stored in the **user_gps_location** field. This table is linked to the **users_table** through the **user_id** field. This relationship ensures that each user's address information can be identified.

←T→				id	user_medicine	user_id	user_height	user_weight	user_chronic_disease	user_allergies
☐	Düzenle	Kopyala	Sil	7	{"Metformin": "500mg"}	6	210.00	125.00	Diabetes	Gluten
☐	Düzenle	Kopyala	Sil	8	{"Losartan": "50mg"}	7	167.00	68.00	Hypertension	NULL
☐	Düzenle	Kopyala	Sil	9	{"Salbutamol": "2mg"}	8	185.00	90.00	Asthma	NULL
☐	Düzenle	Kopyala	Sil	10	NULL	9	162.00	103.00	NULL	Nuts
☐	Düzenle	Kopyala	Sil	11	{"Levothyroxine": "75mcg"}	10	178.00	105.00	Hypothyroidism	NULL
☐	Düzenle	Kopyala	Sil	12	NULL	11	177.00	71.00	NULL	NULL
☐	Düzenle	Kopyala	Sil	13	{"Amitriptyline": "10mg"}	12	160.00	57.00	Migraine	Perfumes
☐	Düzenle	Kopyala	Sil	14	{"Cantab Plus" : "16mg"}	13	0.00	0.00	Hypertension	NULL
☐	Düzenle	Kopyala	Sil	15	{"Multivitamin": "1 tablet daily"}	14	167.00	65.00		NULL
☐	Düzenle	Kopyala	Sil	16	{"Forpack": "12/200 mcg"}	15	183.00	92.00	Asthma	Animal dander

Figure 5.2.1.1.4 (health_info_table)

The **health_info_table** is a table that stores users' health information. Each user's height and weight are stored in the **user_height** and **user_weight** fields, respectively. These details are later used to provide the **user_body_size** parameter, as shown in Figure 5.2.1.6, on the Body Point Metrics, which is described in section 4.2.1.1. Chronic illnesses are specified in the **user_chronic_disease** field, while allergy information is recorded in the **user_allergies** field. Medications regularly used by the user are stored in the **user_medicine** field. For the **medicine_equivalent** medications specified in Figure 5.2.1.5, AI identifies equivalents based on the **user_medicine** data entered here and writes them into the corresponding **medicine_equivalent** field. This table is linked to the **users_table** through the **user_id** field, allowing detailed health information for each user to be recorded.

←T→				id	health_info_id	medicine_name	medicine_dosage	medicine_frequency	medicine_equivalent
☐	Düzenle	Kopyala	Sil	9	7	Metformin	500mg	Twice daily	{"Metforal": "850mg once daily"}
☐	Düzenle	Kopyala	Sil	10	8	Losartan	50mg	Once daily	{"Valsartan": "80mg"}
☐	Düzenle	Kopyala	Sil	11	9	Salbutamol	2mg	As needed	{"Ventolin": "2 puffs"}
☐	Düzenle	Kopyala	Sil	12	11	Levothyroxine	75mcg	Once daily	{"Euthyrox": "100mcg alternate days"}
☐	Düzenle	Kopyala	Sil	13	13	Amitriptyline	10mg	At bedtime	{"Elavil": "10mg"}
☐	Düzenle	Kopyala	Sil	14	15	Multivitamin	1 tablet daily	Daily	{"Centrum": "1 tablet"}
☐	Düzenle	Kopyala	Sil	15	16	Forpack	1-2 tablets once a day	500 mg	{ "Memantine (Namenda)": "5-20 mg", "Axura":...
☐	Düzenle	Kopyala	Sil	16	14	Cantab Plus	20 mg	1 tablet once a day	{ "Paroxetine (Paxil)": "10-40 mg", "Seroxat...

Figure 5.2.1.1.5 (medication_info_table)

The medication_info_table provides detailed information about medications. It includes the **health_info_id** field, which references the **health_info_table** to link the medication data to specific user health information. The **medicine_name** field contains the name of the prescribed or used medication, while the **medicine_dosage** field specifies the recommended dosage (e.g., 500mg, 2mg). The **medicine_frequency** field indicates how often the medication is taken, such as twice daily or as needed. The **medicine_equivalent** field lists equivalent medications and their respective doses, formatted in a JSON-like structure. This table is connected to the **health_info_table** through the **health_info_id** field, ensuring that medication details are accurately linked to user health records and supporting the tracking of medication usage, dosage instructions, and alternative options.

				id	user_id	user_shoe_size	user_body_size	user_special_clothing
☐	Düzenle	Kopyala	Sil	5	6	42.5	Extra Large	None
☐	Düzenle	Kopyala	Sil	6	7	37.0	60,00	Small
☐	Düzenle	Kopyala	Sil	7	8	43.0	Medium	NULL
☐	Düzenle	Kopyala	Sil	8	9	36.5	Small	Wheelchair
☐	Düzenle	Kopyala	Sil	9	10	41.0	Medium	NULL
☐	Düzenle	Kopyala	Sil	10	11	40.5	Small	NULL
☐	Düzenle	Kopyala	Sil	11	12	38.0	Small	Prosthetic Leg
☐	Düzenle	Kopyala	Sil	12	13	44.0	Large	NULL
☐	Düzenle	Kopyala	Sil	13	14	39.5	Small	NULL
☐	Düzenle	Kopyala	Sil	14	15	42.0	Medium	NULL

Figure 5.2.1.1.6 (clothing_info_table)

The **clothing_info_table** contains information about users' clothing preferences and requirements. The **user_id** field links the data to the **users_table**, ensuring each record corresponds to a specific user. The **user_shoe_size** field indicates the shoe size of the user, while the **user_body_size** field provides the user's general clothing size, such as Small, Medium, Large, or Extra Large. The **user_special_clothing** field specifies any special clothing needs or accessories, such as a wheelchair or a prosthetic leg. This table facilitates the management of tailored clothing information and special requirements for individual users.

				id	user_id	isDataProcessed	isContractApproved
☐	Düzenle	Kopyala	Sil	6	6	1	1
☐	Düzenle	Kopyala	Sil	7	7	1	1
☐	Düzenle	Kopyala	Sil	8	8	1	1
☐	Düzenle	Kopyala	Sil	9	9	1	1
☐	Düzenle	Kopyala	Sil	10	10	1	1
☐	Düzenle	Kopyala	Sil	11	11	1	1
☐	Düzenle	Kopyala	Sil	12	12	1	1
☐	Düzenle	Kopyala	Sil	13	13	1	1
☐	Düzenle	Kopyala	Sil	14	14	1	1
☐	Düzenle	Kopyala	Sil	15	15	1	1

Figure 5.2.1.1.7 (approval_info_table)

The **approval_info_table** records information related to user data processing and contract approvals. The **user_id** field links each record to a corresponding user in the **users_table**. The **isDataProcessed** field indicates whether the user's data has been processed, typically represented as a binary value (1 for yes, 0 for no). Similarly, the **isContractApproved** field specifies whether the user has approved the required contracts, also using a binary value. This table ensures clear tracking of users' data processing and contract approval statuses, providing a straightforward mechanism for compliance and user management.

5.2.1.2 post_disaster_stock Database

This database represents a relief or logistics management system and primarily consists of four tables: **Category**, **Storage**, **Product**, and **Stock** which are named as categories, storages, products and stocks orderly. These tables are interrelated and organized to manage product categories, storage locations, product details, and stock statuses. The system's purpose is to easily track which categories the products belong to, the warehouses where they are stored, stock levels, and any deficiencies in the warehouses.

←T→				categoryID	categoryName
☐	Düzenle	Kopyala	Sil	1	Kıyafet
☐	Düzenle	Kopyala	Sil	2	Ayakkabı
☐	Düzenle	Kopyala	Sil	3	Özel İhtiyaç
☐	Düzenle	Kopyala	Sil	4	İlaç

Figure 5.2.1.2.1 (categories table)

The **Category** table contains the categories in which products are classified. Categories such as "Clothing," "Footwear," "Special Needs," and "Medication" are defined in the system. Each category is represented by a unique **CategoryID**, which is used in other tables to specify the category to which a product belongs. For example, the "Sweater" product belongs to the "Clothing" category defined in this table.

←T→				StorageID	StorageName	StorageLatitude	StorageLongitude	StorageLocation
☐	Düzenle	Kopyala	Sil	1	Siteler Mahallesi Depo	38.367067	26.861887	Siteler Mah., Güzelbahçe, İzmir 35310
☐	Düzenle	Kopyala	Sil	2	Yalı Mahallesi Depo	38.37655	26.890593	İstikbal Cad., Yalı Mah., Güzelbahçe, İzmir 35310
☐	Düzenle	Kopyala	Sil	3	Kahramandere Mahallesi Depo	38.359755	26.882029	Kahramandere Mah., Güzelbahçe, İzmir 35310
☐	Düzenle	Kopyala	Sil	4	Çamlı Mahallesi Depo	38.3268	26.820108	Çamlı Mah., Güzelbahçe, İzmir 35310

Figure 5.2.1.2.2 (storages table)

The **Storage** table defines the physical locations where products are stored. Each storage location is identified by a unique **StorageID** and contains information such as the name, address, and geographical coordinates of the warehouse. For example, "Siteler Neighborhood Warehouse" is located in Siteler Neighborhood in İzmir, and its coordinate information is also recorded. This data facilitates warehouse management and logistics operations.

				ProductID	ProductName	CategoryID	ProductSize
☐	Düzenle	Kopyala	Sil	1	Kazak	1	Large
☐	Düzenle	Kopyala	Sil	2	Tekerlekli Sandalye	3	Standard
☐	Düzenle	Kopyala	Sil	3	Kış Botu	2	42
☐	Düzenle	Kopyala	Sil	6	Pantolon	1	Extra Large
☐	Düzenle	Kopyala	Sil	7	T-shirt	3	Small
☐	Düzenle	Kopyala	Sil	8	Bebek Bezi	3	Standard
☐	Düzenle	Kopyala	Sil	9	Cantab Plus	4	Standard
☐	Düzenle	Kopyala	Sil	10	Kazak	1	Small

Figure 5.2.1.2.3 (products table)

The Product table contains details about all the products defined in the system. Each product is identified by a ProductID, and the table stores properties such as the product's name, the category it belongs to (CategoryID), and size information. For instance, the "Sweater" product belongs to the Clothing category and is listed as "Large" in size. Similarly, the "Wheelchair" is categorized as a Special Needs product and is designated as "Standard" in size.

				StockID	StorageID	ProductID	ProductAmount	BaseLimit
☐	Düzenle	Kopyala	Sil	3	1	2	100	50
☐	Düzenle	Kopyala	Sil	4	1	3	1500	1000
☐	Düzenle	Kopyala	Sil	5	2	8	800	700
☐	Düzenle	Kopyala	Sil	6	4	1	750	800
☐	Düzenle	Kopyala	Sil	7	1	6	1200	1000
☐	Düzenle	Kopyala	Sil	8	2	2	200	150
☐	Düzenle	Kopyala	Sil	9	3	7	2500	2000
☐	Düzenle	Kopyala	Sil	10	3	9	670	750
☐	Düzenle	Kopyala	Sil	11	2	10	3200	3000

Figure 5.2.1.2.4 (stocks table)

The **Stock** table is used to manage the stock status of products in warehouses. Each stock entry is identified by a **StockID** and specifies which warehouse (**StorageID**), which product (**ProductID**), the quantity of the product available (**ProductAmount**), and the minimum stock level set for the product (**BaseLimit**). The BaseLimit value is populated based on the data from the **post_disaster_app** database. For example, there are 250 "Wheelchairs" in the "Siteler Neighborhood Warehouse," which meets the minimum stock level of 150 units. However, the "Kahramandere Neighborhood Warehouse" has 670 "Cantab Plus" medicines which is below the minimum stock level of 750 units, indicating a stock shortage which is enclosed by the red rectangle. In such cases, real-time notifications about stock shortages are sent to users in nearby locations via the application.

The relationships among these four tables enable the system to easily track where products are stored, their stock statuses, and any shortages. The relationship between the **Category** and **Product** tables ensures that each product belongs to a specific category. The relationship between the **Storage** and **Stock** tables shows where products are physically stored. The relationship between the **Product** and **Stock** tables specifies the type and quantity of products in each warehouse.

This structure provides a robust framework for managing warehouses, categorizing products, monitoring stock levels, and identifying shortages. For instance, products that fall below critical levels in warehouses can be quickly identified, and necessary actions can be taken promptly.

5.3 AI-Driven Route Optimization and Logistics Management for Disaster Relief

5.3.1 Route Optimization Potential

The proposed route optimization system driven by AI will have very high adaptability. Simulation results of related research suggest that reinforcement learning techniques can substantially enhance adaptability to dynamic changes-such as unexpected road closures or sudden traffic surges-compared to conventional methods. These results would also mean a feasibility test for the implementation of advanced AI models in routing applications.

The main advantage of the system is dynamic route optimization. For example, in areas with high urban traffic, the reinforcement learning model selects alternative routes in a balance between minimum travel time and safety on the road. It reduces delays and increases the satisfaction of users by giving real-time solutions adapted to the new situation.

What's more, the system will combine GPS data, traffic APIs, and updates of inventory. It means that the AI model can synthesize the information so routes remain efficient in real time. For example, when a depot goes through stock depletion unexpectedly, it would route the users to the next nearest depot with inventory upon recalculations. This level of responsiveness is highly critical in disaster situations since quick access to vital needs can save lives.

A practical example could consider a post-disaster scenario in Güzelbahçe for various depots that have to supply water and blankets at different neighborhoods. The system will calculate the shortest or less congested routes for the users in transit between each depot, while it does the dynamic adjustment for road closure due to debris. For instance, it will route a user from Yelki to Depot 2 for blankets via Depot 4 for food supplies in anticipation of sudden congestion, which will keep delays at a minimum while enabling users to collect the most essential resources.

Other key points on route optimization include the minimal consumption of available resources. Since the system minimizes travel distances, thereby helping to reduce fuel consumptions and decrease carbon dioxide emissions, all this falls into the broader goals of sustainability; hence, the system is really effective along with being eco-friendly.

Theoretical analysis also underlines the scalability of the model. If the number of depots and users is increased, then the algorithms of the system can handle the added complexity without significant degradation in performance. This is achieved through parallel processing techniques and efficient data structuring that ensure the system remains robust even under high demand.

5.3.2 Multi-Depot Routing Insights

The multi-depot routing strategy has great potential to enhance the efficiency of logistics in cases where users need various items from different depots. This system will, therefore, identify depot visits in order of importance by leveraging clustering techniques in addition to Genetic Algorithms to avoid as much superfluous travel as possible while ensuring timely delivery.

As mentioned in multi-depot title of methodology, these are indeed clear visions on how the proposed approach directly benefits the users by reducing time and effort in travel for the collection of necessities, besides dynamic adaptability to real-time changes that guarantee effectiveness in operational conditions varying from time to time, providing an efficient practical solution to deal with multiple depot scenarios.

Initial multi-depot routing designs look quite promising. The system should reduce unnecessary travel by using clustering techniques and Genetic Algorithms, which are meant for optimal resource allocation and thus help reduce operations costs. Real-time adaptability enhances this system to respond well against any sudden disruption, assuring continuity in critical supply chains.

5.3.3 System Scalability Concept

Scalability is considered the central concept of the design; a designed system should be such that it can scale with more users and more depots without significant degradation in performance. The architecture will also make use of frameworks like Apache Kafka or AWS Lambda for real-time processing of data. In fact, this modular approach to architecture will enable different scaling of various components by leveraging their resource utilization optimally.

Any potential bottlenecks, such as delays in data synchronization between depots or increased latency during high user loads that may be revealed through stress tests, would be proactively solved. Dynamic load balancing algorithms will distribute the user requests evenly across multiple servers, preventing overloads of the system. Moreover, the use of database partitioning strategies, such as horizontal sharding, will help to efficiently manage large volumes of data by their division into smaller, more manageable chunks.

Simulations also reveal that enhanced caching is required to avoid redundant calls to APIs. A multilevel caching system, integrating local with distributed caches, can reduce response time by a huge amount at peak usage. Further testing for scalability will be geared toward incorporating enhancements into simulated exponential growth scenarios of both user requests and numbers of depots. Iterative enhancement will continue to maintain the system robust yet efficient under extreme conditions of operations.

5.3.4 Adaptability to Dynamic Updates

Furthermore, the road conditions and inventory level setups within the system design will be dynamically updated through the streaming of real-time data from multiple APIs, which also include Google Maps and OpenStreetMap. To accomplish this, the architecture of the system will offer complex event-driven processing updates in a timely manner.

Adaptive algorithms, like RL, would form a key basis in the recalculation of routes with new data coming in. For example, in case of an impromptu closure of a road, reinforcement models would ensure that immediately routing shifts directions through alternative routes creating minimum inconvenience. Similarly, any update of the inventory at a depot will lead to a recalculation so that users are routed to the closest resource pool.

Predictive analytics for strengthening the system's reliability means that machine learning models will use experience to predict traffic hours and supply shortages so it may route around them. By knowing what is likely to occur next, the system would automatically adapt to changing environments in the most robust way, reduce the number of delays, and increase user satisfaction.

The system design will be updated dynamically regarding road conditions and inventory. Real-time adjustments in traffic data, adaptive route recalculations enable the user to get to depots with the mission of collecting goods efficiently under various scenarios. This ability is crucial for maintaining dependability in dynamic environments.

5.3.5 Identified Limitations

A number of limitations are expected in the proposed system, which shall be developed and deployed:

1. **Dependency on Third-Party APIs**: The system depends on third-party APIs for data about roads and traffic flow, which may introduce latency or inaccuracies during periods of high demand. To mitigate this, caching strategies could be implemented to store frequently accessed data locally, and backup data sources can be used to ensure continuity during API downtimes.
2. **GPS Coverage Challenges**: Poor GPS coverage will weaken the location tracking accuracy of the system. Alternative localization methods such as Wi-Fi triangulation, cellular tower data, or integration with Bluetooth-based beacons can be integrated for better position accuracy in both urban and remote areas.
3. **Environmental Data Gaps**: Inadequate availability of real-time environmental data regarding weather or disaster-specific information may affect responsiveness to changes in the case of the system. These gaps can be mitigated through the integration of additional data streams from partnerships with meteorological agencies or even satellite-based imaging.
4. **Scalability Issues for Large-Scale Deployments**: While the simulations demonstrate good scalability, real-life tests will be needed for large-scale deployment involving highly populated areas and several depots to reveal hidden bottlenecks. The issues of large-scale operations may be mitigated by introducing load-balancing algorithms supported by distributed computing frameworks like Apache Kafka or Amazon Kinesis.

5. **Complexity of User Interface**: A very technical backend will be hard for users to comprehend in using the application. The design should be intuitive and user-friendly, with clear visual cues, such as step-by-step routing instructions and interactive maps.

5.3.6 Future Directions

Future development will increase the effectiveness of the system in the following ways:

- More data sources, such as drone imagery, will provide real-time environmental insights.
- Developing predictive models using advanced machine learning techniques on variations in traffic and inventory.
- Evaluate other alternative modes of transportation, such as drones or autonomous vehicles, to make last-mile delivery fast.
- Design an offline mode to extend functionality to areas where connectivity is poor.

6. Conclusions

In conclusion, this project highlights the critical role that artificial intelligence (AI) and data-driven optimization techniques play in improving disaster response and resource management. By combining the P-median approach for optimal placement of aid depots with AI-supported drug alternative identification, the system enables more efficient resource allocation and quick access to essential needs.

The results demonstrate that strategically placing at least eight depots will significantly enhance the speed and effectiveness of emergency relief distribution and address the current resource allocation challenges faced by local authorities. Additionally, AI-powered route optimization, using real-time traffic and inventory data, offers a dynamic and adaptive approach to logistics, ensuring the system remains efficient even in the face of unexpected disruptions.

User data collected via the mobile application, along with any relevant medication data, is processed by AI algorithms and stored in databases based on personal information, body mass index calculations, and drug alternative analysis. These databases are integrated with depots, allowing stock levels for each depot to be constantly updated and inventory control to be maintained. As a result, users are quickly directed to the right products, minimizing stock issues and ensuring efficient resource distribution.

Future developments should focus on scaling the system, integrating real-time environmental data, and improving AI models to respond to potential shortages or traffic issues. Additionally, continuous assessment and flexible planning are crucial to ensure that the depot network and delivery routes can adapt to the changing needs of communities.

This comprehensive strategy lays the foundation for a more resilient disaster management infrastructure, where community preparedness, rapid response, and resource optimization become priorities, ultimately saving lives and enhancing the overall effectiveness of post-disaster recovery efforts.

7. References

Bastı, M. (2012). P-medyan tesis yeri seçim problemi ve çözüm yaklaşımları. AJIT-e: Online Academic Journal of Information Technology, 3(7), 29-39. DergiPark

Siegel, Z. E. (2021). P-Median Problems and Solution Strategies. Models for Operations Planning, Scheduling, and Control. Prof. Kumar Rajaram, UCLA.

Ekin, E. (t.y.). Solution Approach to P-Median Facility Location Problem with Integer Programming and Genetic Algorithm.

Sümer, Ü. T. (t.y.). Meta sezgisel yaklaşımlar ile p-medyan tesis yeri seçimi. (Yüksek Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü).

Ünal A., Kazan H., Çuhadar L. International Conference on Smart Logistics - ICSL2022, İstanbul, Türkiye, 24 - 25 Kasım 2022, ss.30-32

Lian, K.Q. and Tribello, G.A., 2024. An elementary approach to the vehicle routing problem via Python and google API. American Journal of Operations Research, 14(6), pp.169-190.

krishna Vaddy, R., 2023. AI and ML for transportation route optimization. International Transactions in Machine Learning, 5(5), pp.1-19.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 328–339.

AI. (2024, March 12). Run your own AI (but private). YouTube. https://youtu.be/WxYC9-hBM_g?si=8Eo7I01x8DCV4tjJ

Huang, K., Yin, H., Huang, H., & Gao, W. (2023, September 22). Towards Green AI in Fine-tuning Large Language Models via Adaptive Backpropagation. ArXiv.org. https://doi.org/10.48550/arXiv.2309.13192

Friederich, S. (2017). Fine-tuning. *The Stanford Encyclopedia of Philosophy, Fall 2017*. https://plato.stanford.edu/entries/fine-tuning/

Rangan, K., & Yin, Y. (2024). A fine-tuning enhanced RAG system with quantized influence measure as AI judge. *Scientific Reports, 14*, Article 27446. https://doi.org/10.1038/s41598-024-27446-1

8. Appendix

The drug substitute search application using artificial intelligence has been developed using Python, JavaScript, CSS, and HTML frameworks. The application consists of four files in total.

8.1 equivalent.py

First, the focus will be on the **equivalent.py** Python file. This file manages the **CORS (Cross-Origin Resource Sharing)** policy and ensures the secure processing of requests from different origins. The application, built on the Flask framework, defines API endpoints and starts the application. Additionally, authentication is performed using the **google.oauth2.credentials** library to access Google APIs. The **google.generativeai** library is used to connect to Google's customized generative AI model.,

The application is initiated with **app = Flask(__name__)**. Then, authentication is performed with the **access_token** value, and the model is connected to the Google AI model with the line **model = genai.GenerativeModel(model_name='tunedModels/medicinedetails-kmbpk7zcth5t')**.

The **/equivalent** API endpoint handles the operations related to the drug names received from users. First, the JSON data from the users is retrieved. The application then sends the query from the **prompt_equivalent** variable to the generative AI model, requesting it to generate a list of similar drugs. The model's response is processed, and if successful, a list of similar drugs is returned. If the model does not respond or if the response is blocked by security filtering, an error message is returned.

The Flask application is run with **app.run(debug=True, port=5000)**. This indicates that the application is running locally, and before the application can start, the equivalent.py file with the Python extension needs to be executed to establish a connection to the Flask API.

8.2 script.js

This JavaScript code provides functionality for users to enter a medicine name and search for equivalent medicines. First, after the page loads, the **DOMContentLoaded** event is triggered, and access is provided to the necessary **HTML** elements (medicine name input, button, loading indicator, error message display, and results list). When the user clicks the "**equivalentButton**" the medicine name input is checked. If no medicine name is provided, an alert message is shown.

If a medicine name is entered, the button is disabled, the loading indicator is activated, the error message and results are hidden. Then, a **POST** request is sent to the **/equivalent** endpoint using the **fetch()** API, and the medicine name is transmitted to the Flask API in JSON format. Based on the response from the API, the process continues. If the API response is successful, equivalent medicines are listed and displayed on the screen. If any errors occur, the error message is shown on the screen. Once the process is complete, the loading indicator is hidden, and the button is re-enabled.

8.3 index.html

This HTML file provides an interface for users to enter a medicine name and search for equivalent medications. It includes an input field for the medicine name, a search button, a loading indicator, an error message area, and an empty space where results will be displayed. The functionality is handled by the **JavaScript** file (script.js), and the page works with user interactions once it's loaded. The **CSS** file (style.css) is also included to style the page.

8.4 style.css

This CSS file styles the HTML page layout. The page has a centered structure with content displayed inside a box with a transparent background. The title and input fields are styled. When the user enters a medicine name, the button becomes visible, and when clicked, a loading indicator appears. A spinning icon is shown during the loading process. Error messages are displayed in red, and the results are listed with space between each medicine name.