Vivekanand Education Society's Institute Of Technology

Department of Information Technology

DSA Mini Project

AY 2025-26

TITLE : Pipeline Management System using Dijkstra's Algorithm

SDG : Industry, Innovation and Infrastructure

Domain: Data Structures and Algorithms

Member: Ayush Lokhande

Mentor Name: Kajal Jewani

Content


1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
THE GLOBAL GOALS
For Sustainable Development
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE AND JUSTICE STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS

Introduction to Project

1. The **Pipeline Management System** is developed to efficiently design, monitor, and control the flow of resources like water or gas through interconnected pipelines. It aims to streamline network operations and minimize human intervention for better management.

2. The project applies **Dijkstra's Algorithm** to calculate the **shortest and most efficient path** between different nodes in the pipeline network, ensuring smooth and optimized routing of resources.

3. It focuses on achieving **cost-effective**, **safe**, and **reliable** pipeline operations by reducing unnecessary distance, energy usage, and maintenance challenges within the system.

4. The system contributes to improved **resource distribution**, faster **fault detection**, and effective **maintenance planning**, leading to enhanced performance and reduced downtime.

Problem Statement

1. Monitoring and managing extensive pipeline networks manually leads to inefficiencies, delayed fault detection, and difficulty in assessing the overall health of the system.

2. Traditional routing and management systems lack **data analytics** to evaluate pipeline conditions, material performance, and long-term durability.

3. Without a visual and analytical interface, it becomes hard to **interpret network status** and identify issues like leakage, pressure drops, or material failures.

4. There is a need for an **intelligent, data-driven system** that uses algorithms and analytics to optimize routes, track material health, and present insights through interactive graphs.

Objectives of the project

1. To design an efficient **Pipeline Management System** that automates route optimization, monitoring, and maintenance using **Dijkstra's Algorithm**.

2. To implement **health analytics** for assessing pipeline conditions, material strength, and performance across different sections of the network.

3. To provide **graphical representations and visual insights** for better understanding of network flow, cost analysis, and fault detection trends.

4. To ensure a **cost-effective, safe, and reliable** distribution system that supports timely maintenance and improved decision-making through data analytics.

Requirements of the system (Hardware, software)

🖥️ **Hardware Requirements**

1. A computer system with at least **Intel i5 or equivalent processor** to efficiently execute algorithmic computations and graphical rendering.

2. Minimum **8 GB RAM** and **500 GB storage** to handle large pipeline datasets and store analytical reports and visual outputs.

3. A **high-resolution display** and **stable internet connection** for smooth visualization of pipeline analytics and real-time monitoring.

⚙️ **Software Requirements**

1. **Operating System:** Windows / Linux for running development and testing environments seamlessly.

2. **Programming Tools:** Python / Java with libraries for graph algorithms, data visualization (like Matplotlib or JavaFX Charts).

3. **Database:** MySQL / SQLite for storing node data, pipeline material details, and system logs efficiently.

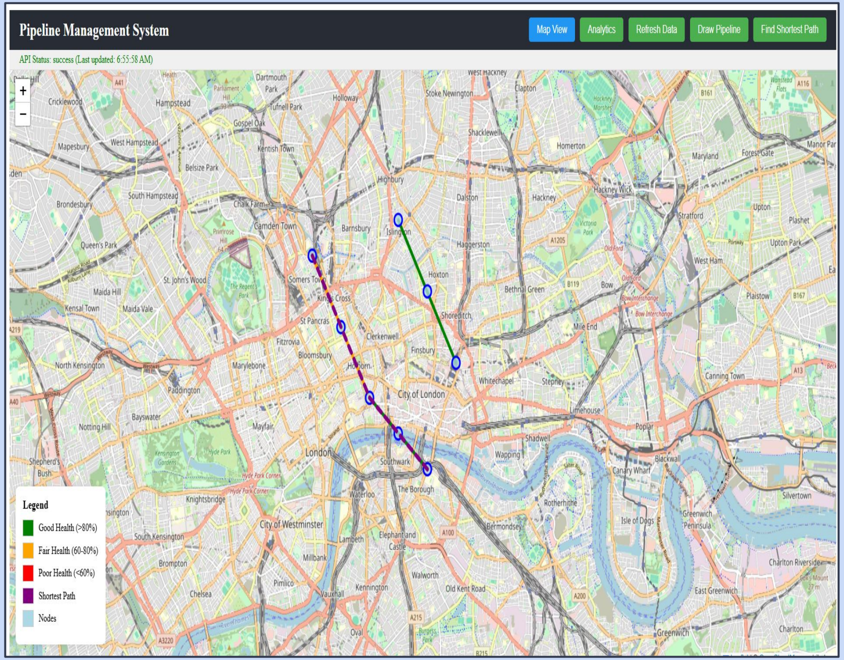
Data Structure and Concepts Used

- The pipeline network is represented as a **Graph**, where nodes denote junctions or stations, and edges represent pipelines connecting them.
- An **Adjacency List** structure is used to store the graph efficiently and to keep track of all connected pipelines for each node.
- **Dijkstra's Algorithm** is applied to compute the **shortest path** between two nodes based on distance, cost, or efficiency.
- A **Priority Queue (Min-Heap)** is utilized to select the next node with the minimum distance, ensuring optimal performance of the algorithm.
- **Arrays and Hash Maps** are used to store node distances, visited nodes, and parent relationships for path reconstruction.
- **Linked Lists** are used to dynamically maintain pipeline paths and sequences for route visualization and analytics.
- **Analytical Data Structures** such as tables and graphs are used to represent **pipeline health, material efficiency, and performance metrics** visually.

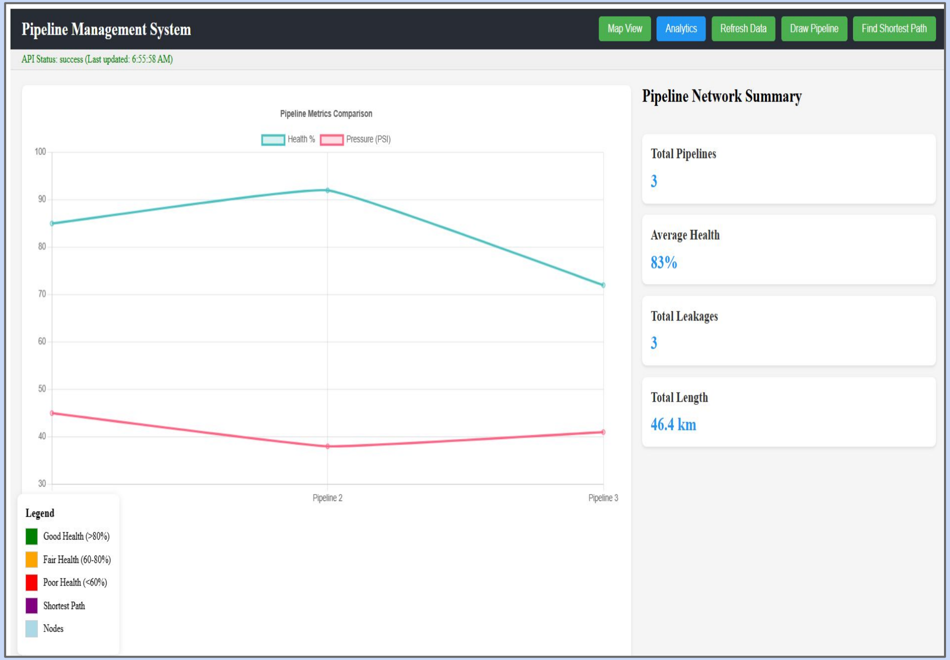
Front End

- The **front end** of the Pipeline Management System is developed using **JavaFX**, providing an interactive and user-friendly graphical interface.
- It allows users to **visualize the pipeline network**, including nodes, paths, and shortest routes calculated using Dijkstra's Algorithm.
- The interface includes **graphical charts and analytics dashboards** to display pipeline health, material performance, and efficiency metrics.
- Users can **input network data**, view **real-time results**, and generate **graph-based insights** for decision-making.

Front End



Pipeline Management System
Map View
Analytics
Refresh Data
Draw Pipeline
Find Shortest Path
API Status: success (Last updated: 6:55:58 AM)
Legend
Good Health (>80%)
Fair Health (60-80%)
Poor Health (<60%)
Shortest Path
Nodes

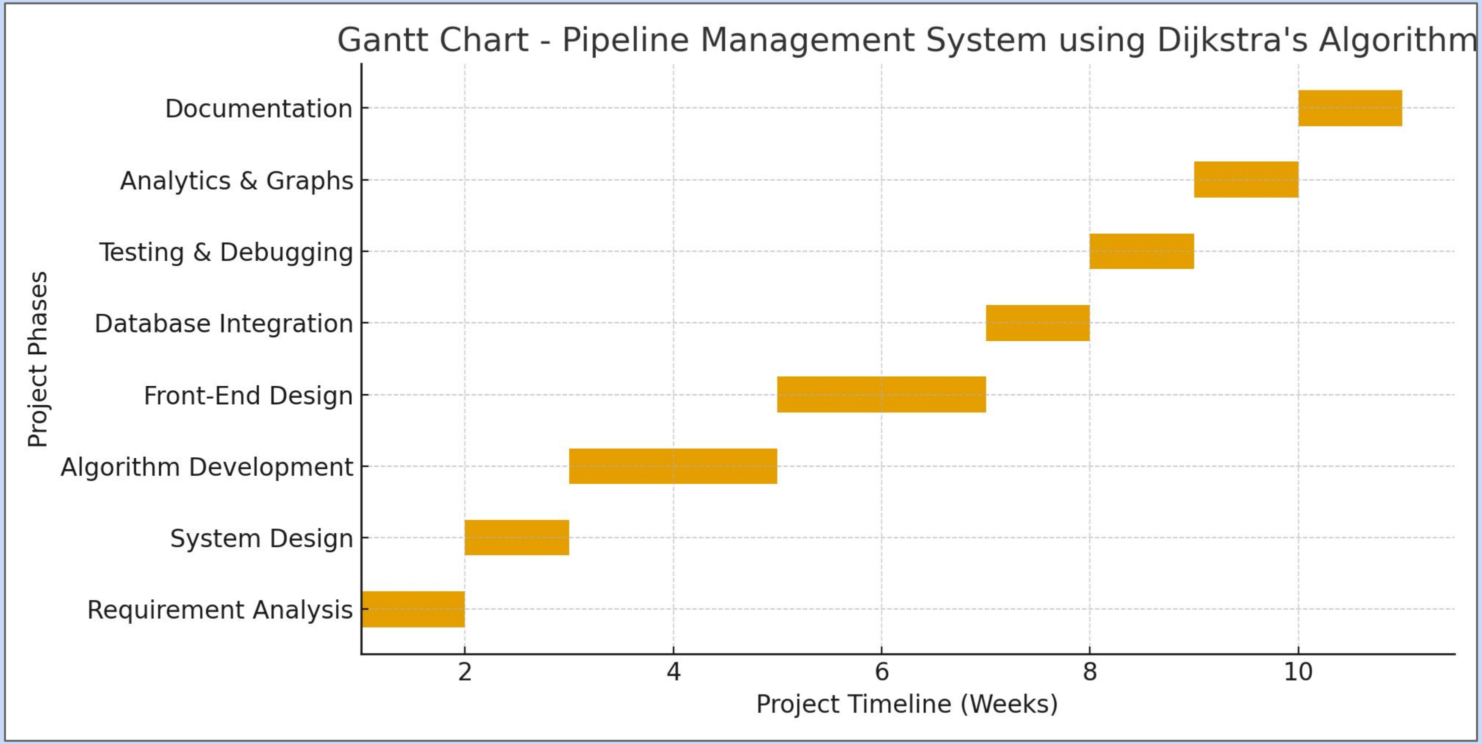


Pipeline Management System
Map View
Analytics
Refresh Data
Draw Pipeline
Find Shortest Path
API Status: success (Last updated: 6:55:58 AM)
Pipeline Metrics Comparison
Health %
Pressure (PSI)
Pipeline 2
Pipeline 3
Pipeline Network Summary
Total Pipelines
3
Average Health
83%
Total Leakages
3
Total Length
46.4 km
Legend
Good Health (>80%)
Fair Health (60-80%)
Poor Health (<60%)
Shortest Path
Nodes

Implementation

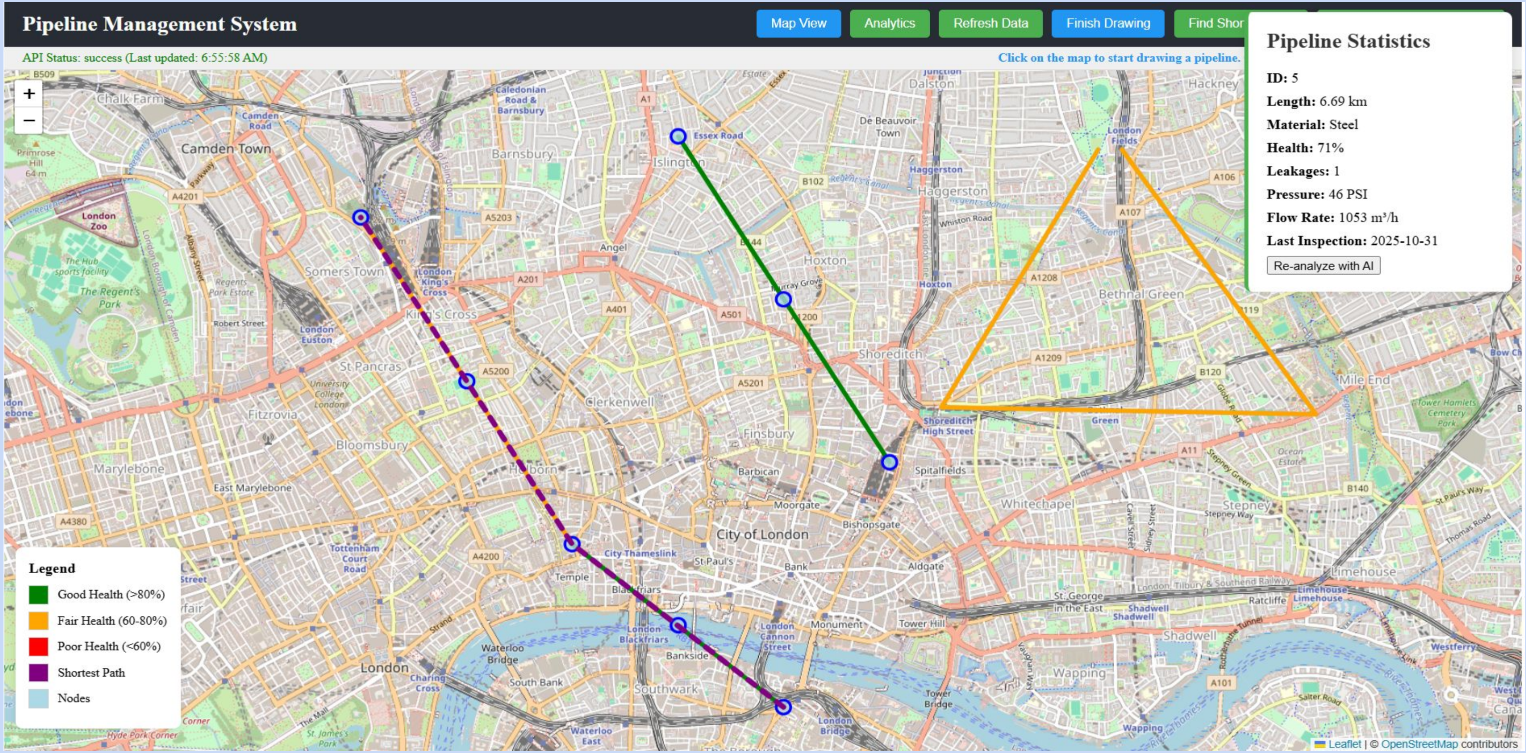
1. The pipeline network is modeled as a **graph** with nodes and weighted edges representing junctions and pipeline lengths.
2. **Dijkstra's Algorithm** is implemented to compute the **shortest and most cost-effective path** between selected nodes.
3. The system uses an **Adjacency List** and **Priority Queue** for efficient graph storage and path selection.
4. A **JavaFX-based front end** provides visualization of nodes, routes, and real-time analytics dashboards.
5. **Health and material data** are analyzed to generate performance metrics and graphical reports.
6. All pipeline, node, and analytics data are **stored and retrieved from a database** for seamless management.

Gantt Chart



Gantt Chart - Pipeline Management System using Dijkstra's Algorithm
Documentation
Analytics & Graphs
Testing & Debugging
Database Integration
Front-End Design
Algorithm Development
System Design
Requirement Analysis
Project Phases
2
4
6
8
10
Project Timeline (Weeks)

Output



Pipeline Management System
Map View
Analytics
Refresh Data
Finish Drawing
Find Shor
API Status: success (Last updated: 6:55:58 AM)
Click on the map to start drawing a pipeline.
Pipeline Statistics
ID: 5
Length: 6.69 km
Material: Steel
Health: 71%
Leakages: 1
Pressure: 46 PSI
Flow Rate: 1053 m³/h
Last Inspection: 2025-10-31
Re-analyze with AI
Legend
Good Health (>80%)
Fair Health (60-80%)
Poor Health (<60%)
Shortest Path
Nodes
Leaflet | © OpenStreetMap contributors

Conclusion

- ★ The **Pipeline Management System** efficiently models the entire pipeline network, enabling accurate and reliable monitoring.
- ★ Using **Dijkstra's Algorithm**, the system determines the **shortest and most cost-effective routes** between junctions.
- ★ The integration of **graph-based data structures** ensures optimal performance in path computation and resource utilization.
- ★ The **JavaFX-based interface** provides an intuitive visualization of pipelines, routes, and analytics for easy understanding.
- ★ **Analytical features and health monitoring** offer valuable insights into pipeline material usage, maintenance, and efficiency.
- ★ Overall, the project demonstrates how **Data Structures and Algorithms can solve real-world infrastructure challenges** effectively.

References

[1] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, "Smart cities of the future," *European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012.

[2] S. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[3] A. Ramesh, R. S. Kumar, and K. Balaji, "Pipeline leakage monitoring and control system for smart city water distribution using IoT," in *Proc. IEEE Int. Conf. on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, Chennai, India, 2017, pp. 356–362.

[4] M. Sharma and R. K. Singh, "Efficient water pipeline monitoring and leak detection system using sensors and data analytics," in *Proc. IEEE Int. Conf. on Smart City and Emerging Technology (ICSCET)*, Mumbai, India, 2018, pp. 1–5.

[5] React Developers, "React – A JavaScript library for building user interfaces," *Meta Open Source*, 2025. [Online]. Available: https://react.dev/