UIT2404 – Advanced Data Structures and Algorithm Design

Project Report

Academic year
2022-2023

Oceanly Pipeline Networks

Client – Dr. S. Karthika
(Associate Professor, IT department)

- Thirumalaivasan K – 3122 21 5002 115
- Thiruveditha S- 3122 21 5002 116
- Ujjwal Singh Chib– 3122 21 5002 117
- Sri Sai Ankit V – 3122 21 5002 118
- Vasundhara B– 3122 21 5002 119
- Vemula Muni Karthik - 3122 21 5002 120

INDEX

1) Abstract.. 3

2) Introduction.................................... 4

3) Motivation...................................... 4

4) Objective.. 5

5) Problem Statement 6

6) Literature Review............................. 7

7) Requirements engineering.................... 9

8) Risk Management…………………… 10

9) Design and development of solution........ 27

10) Process Management......................... 29

11) Deployment 34

12) Gap Analysis............................... 38

13) Conclusion and Further Scope............ 40

14) References................................... 43

1)Abstract

This project aims to optimize water distribution systems by computing an optimum pipeline route for delivering water to residential locations. The project utilizes maps APIs to gather information on water bodies, industries, and houses in proximity to the desired location. A connected graph is constructed to represent the spatial relationships, and the Minimum Spanning Tree (MST) algorithm, specifically Prim's algorithm, is employed to establish a cycle-free network. A Depth-First Search (DFS) algorithm is then implemented to explore all possible paths from the water body source to the destination. By utilizing a priority queue data structure, the most optimal path from source to destination within the MST is identified and highlighted in blue, while the remaining paths are marked in red. The project report provides an overview of the methodology, implementation details, and emphasizes the significance of an optimized pipeline route for efficient water distribution.

2)Introduction

Efficient water distribution is crucial for ensuring a reliable and sustainable water supply to residential areas. However, traditional pipeline routing approaches often overlook important factors such as proximity to water bodies, industries, and other houses. This project aims to address these limitations by optimizing the pipeline route. By leveraging advanced algorithms and maps APIs, the project constructs a connected graph representing the spatial relationships. The Minimum Spanning Tree (MST) algorithm, specifically Prim's algorithm, is then employed to establish a cycle-free network. Subsequently, the Depth-First Search (DFS) algorithm is utilized to explore all potential paths from the water body source to the destination. By employing a priority queue data structure, the most optimal path within the MST is identified and highlighted, enabling efficient water distribution.

3)Motivation

The motivation behind this project stems from the need to enhance the efficiency and sustainability of water distribution systems. Arbitrary pipeline routing often leads to resource wastage and higher operational costs. By optimizing the pipeline route, the project aims to minimize these inefficiencies. Additionally, considering factors such as the proximity of industries and potential pollution sources allows for better environmental impact mitigation, ensuring the delivery of clean and safe water to residents. The project's focus on operational efficiency aims to reduce maintenance requirements, thereby improving the overall effectiveness of water distribution systems.

4)Objective

The primary objective of this project is to compute an optimum pipeline route for delivering water to a specific residential location. The project aims to achieve the following objectives:

Utilize maps APIs to identify water bodies, industries, and houses in proximity to the desired location, providing accurate and up-to-date data for pipeline routing.

Construct a connected graph that represents the spatial relationships between water bodies, industries, and houses, forming the basis for optimizing the pipeline route.

Apply Prim's algorithm to generate a Minimum Spanning Tree (MST) devoid of cycles within the graph, minimizing the total pipeline length and optimizing resource utilization.

Implement the Depth-First Search (DFS) algorithm to explore all potential paths from the water body source to the destination, considering distance, terrain, and infrastructure.

Utilize a priority queue data structure to identify the most optimal path from the source to the destination within the MST, based on predefined criteria such as the shortest distance, minimal environmental impact, and avoidance of potential hazards.

By achieving these objectives, the project aims to improve the efficiency, sustainability, and reliability of water distribution systems, ensuring a consistent supply of clean water to residential areas while minimizing resource wastage,and operational costs.

5)Problem Statement

The problem statement of this project entails the requirement to create a software application that addresses the design challenges of a water distribution system at the residential level. The system consists of a network of pipeline valves and outlets, where each valve represents a one-to-many water supply point.
The application should be able to optimize the water delivery process within this network, taking into account the unknown constructional plan. The constructional plan refers to the specific layout and interconnections of valves and pipelines, which may vary from one residential area to another.
The primary goal is to develop an efficient solution that ensures reliable water distribution from any node to every other node in the network, but more particularly to the specified destination nore or residential location. This means that the pipeline should be designed in such a way that all locations get adequate supply of water but more priority should be given to the given residential location
The application should consider various factors in order to achieve an optimized design. These factors may include the distance between nodes, avoidance of cycles and unnecessary loops, and any constraints or limitations imposed by the residential area.
By creating an optimized water delivery system, the application will improve the overall efficiency of water distribution at the residential level. This will result in better access to water resources, reduced wastage, and an enhanced user experience for the residents.
Ultimately, the application should provide a reliable and flexible solution that can be adapted to different residential areas, accommodating their unique constructional plans and ensuring an optimized water distribution system.

6)Literature review

Mathematical Approaches:
a. Combinatorial approach: The optimization of residential water distribution systems can be approached from a combinatorial perspective, considering the connections between pipeline valves and outlets. Combinatorial techniques can be used to analyze the network structure and optimize the delivery system from each node to every other node.
b. Graph theory: The network of pipeline valves and outlets can be represented as a graph, with nodes representing the valves and outlets, and edges representing the pipelines. Graph theory can provide insights into connectivity, flow paths, and optimization algorithms for efficient water distribution in residential systems.
Optimization Techniques: a. Linear programming: Linear programming techniques can be applied to optimize the water distribution system by formulating the problem as a mathematical program with linear constraints. This approach considers objectives such as minimizing cost, maximizing efficiency, and ensuring equal distribution of water to residential nodes.
b. Genetic algorithms: Genetic algorithms, inspired by natural selection and evolution, can be employed to optimize the water distribution system. These algorithms generate and evolve a population of potential solutions, evaluating their fitness based on factors such as pipeline lengths, pressure constraints, and demand requirements.
c. Simulated annealing: Simulated annealing is a probabilistic optimization algorithm that mimics the annealing process in metallurgy. It can be used to find near-optimal solutions by iteratively searching the solution space while allowing for occasional uphill moves. Simulated annealing can effectively optimize the water distribution system by considering factors such as pressure regulation, pipe sizing, and flow balancing.

Case Studies and Applications:
a. Real-world applications: Several studies have applied optimization techniques to real-world residential water distribution systems, considering factors such as pipe sizing, network topology, demand patterns, and hydraulic constraints. These case studies provide insights into the effectiveness and practical implementation of optimization approaches.
b. Sustainability and resilience: Optimization of water distribution systems in residential areas can contribute to sustainable and resilient water management. By minimizing energy consumption, reducing water losses, and ensuring equitable distribution, optimized systems can improve resource efficiency and adaptability to changing conditions.
c. Integrated management systems: Optimization techniques can be integrated into broader water management systems, incorporating factors such as water supply, demand forecasting, and asset management. This integrated approach enhances the overall efficiency and effectiveness of residential water distribution systems.
The literature review highlights the importance of mathematical approaches and optimization techniques in designing and optimizing residential water distribution systems. Combinatorial approaches and graph theory provide insights into network structure and connectivity, while linear programming, genetic algorithms, and simulated annealing offer effective optimization strategies. Real-world case studies and the application of optimization techniques contribute to sustainable and resilient water management in residential areas.

7)Requirements engineering

Functional Requirements:

1)Network Creation:
The system should create a connected graph representing the water distribution network.
It should apply Prim's algorithm to derive the minimum spanning tree (MST) of the network.
2)Optimization:
The system should compute the optimum path from a water body to a residential location.
It should prioritize minimizing the cost of the pipeline and reducing the number of industries covered.
3)User Interface:
The system should provide a user-friendly front end that allows users to input the graph and view the optimized path.
It should display the water distribution network on a map interface, highlighting the optimum path.
4)Integration with Maps API:
The system should integrate with the Google Maps API to derive coordinates of water bodies, industries, and distances between locations.
It should use the Maps API to visualize the water distribution network and display nearby places.
Backend and API Integration:
The system should use a Python backend program (3.10.2) to handle the computation and optimization tasks.
It should integrate with Flask and REST APIs to transfer data between the frontend and backend in JSON format.

Non-Functional Requirements:

Performance:
The system should compute the optimized path and display the results within an acceptable time frame, considering real-time API calls.
It should handle a reasonable number of nodes and edges efficiently, ensuring smooth performance.
Usability:
The user interface should be intuitive, visually appealing, and easy to navigate.
The system should provide clear instructions and feedback to the users during the interaction.
Reliability:
The system should handle unexpected errors gracefully and provide appropriate error messages to the users.
It should validate user inputs and handle edge cases effectively.
Scalability:
The system should be designed to handle larger networks and scale up to accommodate multiple residential areas.
It should maintain performance and optimization capabilities even with increased data and network complexity.
Security:
The system should ensure the privacy and security of user data, especially when integrating with external APIs.
It should handle authentication and authorization securely if user accounts and access control are implemented.
These requirements form the basis for developing and implementing a water distribution solution that optimizes the pipeline network based on cost, coverage, and real-time data.

8)Risk Management

Anticipated risks:

●Understanding the team members and their issues. (Moderate)

● Completion of basic and important requirements in the first place within the time constraint. (Moderate)
● Selection of appropriate data structures that goes along with the user requirements. (Moderate)
● Finding an effective API and having the work done within the authorized limit to requests. (Moderate)

Mitigation plan:

● Keeping team members updated and manage to have proper communication.
● Using sprints (JIRA) effectively to improve productivity.
● Referred various books and online materials to get familiar with Advanced Data Structures.
● Choosing open-source API over paid source

	Sprint	**User Story #**	**User Story**	**Essential/Desirable**	**Need of requirement**	**Description of requirement**
1	1	1	Risk management	Essential	Minimize potential disruptions and ensure project success	Identify and manage potential project risks
		2	Logic to imply	Essential	Ensure accurate and efficient implementation	Define the underlying logic and algorithms
		3	Working with JIRA		Sequential Project planning	Utilize JIRA for project management and tracking

		4	Features of UI	Essential	Create an intuitive and engaging user experience	Define desired user interface features
		5	Data Storage Options	Essential	Determine the most suitable data storage solution that provides scalability, performance, data integrity,	Identify and evaluate various data storage options, such as relational databases, NoSQL databases, or cloud-based storage solutions, based on project requirements.
	2	6	Prototype Development - Main page UI		Visualize and validate the user interface design	Develop a prototype for the main page UI
		7	Backend Development		al Enable data ng and system functionality	Backend functionalities, such as handling user inputs, storing data, and implementing the Prims and dfs algorithm, were developed.
		8	API Resea		external services and data sources	Research was conducted to identify suitable APIs for integrating map functionalities and obtaining accommodation information.

2	3	9	Main Page Integration w/Flask	Essential	seamless communicati on between front end and back end	Integrate the main page user interface (UI) with the Flask framework, enabling communication between front-end and back-end components.
		10	Product Page UI		Present product formation effectively and attractively	The UI for the product page, where the best accommodation options are displayed, was designed and implemented.
		11	Maps API and Places API	Essential	location-based information and services	Integrate Maps API and Places API to access location-based data, including maps, geolocation, nearby places, and other related information.
	4	12	Knapsack Integration w/ Flask	Essential	efficient accommoda tio n recommendat io n	Integrate the prims algorithm, implemented using greedy programming, with the Flask framework to recommend the best accommodation options based on user preferences.

		13	Main and product page flow		l Ensure smooth experience and interaction	The flow between the main page and the product page was implemented and tested to ensure a seamless user experience.
3	5	14	Testing Main Page		Identify and resolve any issues or bugs	Conduct comprehensive testing of the main page functionality, including user interactions, form submissions, error handling, and data validation.
		15	Testing Product Page		Ensure proper ality and user experience	Conduct comprehensive testing of the product page functionality, including product information display, user
						interactions, and compatibility across different devices.
		16	Deploying the project		Make the application accessible to users	Deploy the project to a production environment, configuring the necessary infrastructure, server setup, and ensuring accessibility to users.

Implementation and Risk Management

Name: Sri Sai Ankit V

Register Number: 3122 21 5002 118

Roles: Developer

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of requirement
4	1	3	Working with JIRA	Essential	Streamline project planning and collaboration	Utilize JIRA for project management and tracking
	2	6	Backend Development	Essential	Enable data processing and system functionality	Backend functionalities, such as handling user inputs, storing data, and implementing the prims algorithm and dfs algorithm and using appropriate data structures
4	5	15	Deploying the project	Essential	Make the application accessible to users	Deploy the project to a production environment, configuring the necessary infrastructure, server setup, and ensuring accessibility to users.

Risk Management :

Risk #	Risk Description	Probability	Impact	Mitigation Plan

1	Developers inability to come up with an efficient logic to solve the problem	High	High	Understand the problem statement thoroughly, list down the given data , assumptions to make and an overall blue print of how the project is to be designed. Talk with the client regularly to get a clear view of his/her requirements
2	Insufficient knowledge or expertise in data structures and algorithms to implement for the problem	Moderate	Moderate	.Understand the concept of all data structures and algorithms being taught, verify internet and other books for the same

Test Log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	Error Handling	Triggering errors or exceptions	Appropriate error messages	PASS
2	2	Performance Testing	Simulating high user tools	Stable and responsive backend	PASS

Implementation and Risk Management

Name: Vemula Muni Karthik

Register Number: 3122 21 5002 120

Roles: Developer

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of Requirement
3	1	5	Integration of python backend with flask	Essential	Determine the most suitable way to integrate the python backend with maps api for the overall program to be executable	Identify the various tools or platforms we can us to provide a smooth integration between a google third party api and our backend
	2	7	Backend Development	Essential	Working on the google maps pltatform	Backend functionalities, such as handling user inputs, storing data ,transferring data between different files in json format and working different test cases

3	5	15	Deploying the project	Essential	Make the application accessible to users	Deploy the project to a production environment, configuring the necessary infrastructure, server setup, and ensuring accessibility to users.

Risk Management :

Risk #	Risk Description	Probability	Impact	Mitigation Plan
1	Unable to figure out a free api over an api that demands billing.	High	High	Having a clear idea on our project requirements Develop thorough knowledge on API'S their functionalities and refer various resources for the same

Test Log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	API Endpoint Testing	Sending requests to API endpoints	Proper response and expected data	PASS

2.	2	Data Validation	Submitting forms with valid/invalid data	Proper validation and error messages displayed	PASS
3	3	User Story Acceptance	User story and acceptance criteria	User story meets acceptance criteria	PASS
4	4	Feature Prioritization	Product backlog	Proper prioritization of features	PASS
5	5	Acceptance Criteria Clarity	Reviewing acceptance criteria	Clear and unambiguous acceptance criteria	PASS

Implementation and Risk Management

Name: B Vasundhara

Register Number: 3122 21 5002 119

Roles: Scum Master,Developer

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of Requirement
1	1	1	Risk management	Essential	Minimize potential disruptions and ensure project success	Identify and manage potential project Risks

2	3	9	Main Page Integration w/Flask	Essential	Enable seamless communication between front end and back end	Integrate the main page user interface (UI) with the Flask framework, enabling communication between front-end and back-end components.

Risk Management :

Risk #	Risk Description	Probability	Impact	Mitigation Plan
1	Scrum Masters lack of availability due to conflicting responsibilities	High	High	Plan and prioritize the Scrum Masters responsibilities to minimize conflicts with back end programming tasks. Allocate additional resources or delegate specific Scrum Master duties to other team members when necessary to ensure smooth functioning of scrum process
2	Insufficient knowledge or expertise in Scrum processes impacting project efficiency	Moderate	Moderate	Provide training and resources to enhance the Scrum masters knowledge of scrum methodologies and best practices

Test Log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	Database integration	Using SqlAlchemy provided by Flask	Data consistency and accuracy	PASS
2.	2	Sprint planning meeting	User stories and backlog	Prioritized backlog and sprint plan created	PASS
3	3	Sprint review meeting	Completed user stories and demos	Client satisfaction and feedback	PASS

Implementation and Risk Management

Name: S Thiruveditha

Register Number: 3122 21 5002 116

Roles: Developer

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of requirement
1	1	4	Features of UI	Essential	Create an intuitive and engaging user experience	Define desired user interface features

	2	6	Prototype Developm en t - Main page UI	Essential	Visualize and validate the user interface design	Develop a prototype for the main page UI
	3	10	Product Page UI	Essential	Present product information effectively and attractively	The UI for the product page, where the the search option for computing pipeline is displayed, was designed and implemented.
	4	13	Main and product page flow	Essential	Ensure smooth user experience and interaction	The flow between the main page and the product page was implemented and tested to ensure a seamless user experience.
1	5	14	Testing Main Page	Essential	Identify and resolve any issues or bugs	Conduct comprehensive testing of the main page functionality, including user interactions, form submissions, error handling, and data validation.

Risk Management :

Risk #	**Risk Description**	**Probability**	**Impact**	**Mitigation Plan**

1	Incomplete or unsatisfactory website design	Moderate	High	Clearly define requirements: Ensure that the requirements for the website design are well-documented and agreed upon by all stakeholders. This will help mitigate the risk of misunderstandings or gaps in the design process.

2	Member's inability to meet project deadlines	Moderate	High	Regularly monitor the member's progress and provide support as needed. Break down tasks into smaller milestones with clear deadlines and conduct periodic reviews to track progress. Offer assistance or reallocate resources to ensure timely completion of tasks.

Test Log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	2	Homepage Layout Test	N/A	Correct layout with all required elements	PASS
2.	2	Navigation Menu Test	Clicking on menu items	Proper navigation to respective pages	PASS

Implementation and Risk Management

Name: Ujjwal Singh Chib

Register number: 3122 21 5002 117

Role in the project: Developer

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of requirement
1	2	6	Prototype Develop me nt - Main page UI	Essential	Visualize and validate the user interface design	Develop a prototype for the main page UI
1	3	14	Testing Main Page	Essential	Identify and resolve any issues or bugs	Conduct comprehensive testing of the main page functionality, including user interactions, handling, and data validation.

Risk Management :

Risk #	Risk Description	Probability	Impact	Mitigation Plan
1	Lack of coordination between testing activities and backend-frontend integration	High	Moderate	Establish clear communication channels between the member responsible for testing and the backend and frontend development teams. Encourage regular meetings and feedback loops to align testing activities with ongoing integration efforts. Collaborate closely to

				address any issues or dependencies promptly.
2	Inadequate test coverage due to limited testing resources	Moderate	High	Allocate sufficient resources for testing activities, including time, tools, and personnel. Prioritize test cases based on critical functionalities and potential risks. Implement test automation where feasible to optimize testing efforts and improve coverage.

Test Log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	User interface Testing	Interfacing with UI elements	Intuitive and user friendly interface	PASS

Implementation and Risk Management

Name: Thirumalaivasan K

Register number: 3122 21 5002 115

Role in the project: Developer

Implementation

Epic	Sprint	User Story#	User Story	Essential/ Desirable	Need of requirement	Description of requirement
2	3	11	Maps API and Places API	Essential	Display location based information and services	Integrate Maps API and Places API to access location-based data, including maps, geolocation, nearby places, and other related information.

Risk Management :

Risk #	Risk Description	Probability	Impact	Mitigation Plan
1	Insufficient expertise or knowledge in Maps Api and which one to prefer based on the project scenario	Low	Moderate	Do thorough research on maps API and how to create an account in the API, details regarding how to manage them etc
2	Inadequate test coverage due to limited testing resources	Moderate	High	Allocate sufficient resources for testing activities, including time, tools, and personnel. Prioritize test cases based on critical functionalities and potential risks. Implement test automation where feasible to optimize testing efforts and improve coverage.

Test Log report

TC id	RS #	Test case description/ condition	Test case input	Expected Output	Result (PASS/ FAIL)
1	1	Error Handling test	Triggering errors or exceptions	Proper error messages and handling	PASS
2	2	Performance testing	Simulating high user loads	Stable and responsive system performance	PASS

.

9)Design and Development of Solution:

1.Connectivity Analysis:
Connect all nodes in the water distribution network assuming a connected graph.
Use Prim's algorithm to derive the minimum spanning tree (MST) of the graph.
Ensure there is a path from one node to every other node, avoiding cycles and minimizing the overall cost.

Traversal and Analysis:
Start with a water body as the source node and perform Depth-First Search (DFS) traversal on the MST.
Traverse the MST to reach the residential destination node.
Compute the distance travelled and the number of industries covered during the traversal.

Priority Queue:
Create a priority queue, implemented as a min heap, to store the traversal paths.
Assign the priority to each path based on the number of industries covered (first priority) and the distance travelled (second priority).
Push the traversal paths to the priority queue.

Optimum Path Determination:
The root of the priority queue (top element of the min heap) represents the optimum path in the network to the residential location.
Extract the root element from the priority queue to obtain the optimal path information.

Frontend Development:
Use HTML and CSS to create a user-friendly frontend interface.
Incorporate Maps API (such as Google Maps API) to derive the coordinates of places nearby the water body and industries.
Display the derived coordinates, distances, and industries on the frontend interface to provide visual representation and information to the user.

Implementation Details:
Implement the connectivity analysis, MST derivation, and DFS traversal using suitable algorithms and data structures.
Utilize a priority queue (min heap) to store and prioritize the traversal paths based on the defined criteria.
Integrate Maps API to retrieve coordinates and calculate distances between locations.
Design appropriate functions and modules to perform the required computations and optimize the water distribution path.
Tools and Technologies:
Programming Language: Choose a suitable programming language for implementation (e.g., Python).

HTML and CSS: Create a user-friendly frontend interface using HTML and CSS.
Maps API: Utilize a Maps API (e.g., Google Maps API) to retrieve coordinates and calculate distances.
Data Structures: Use data structures such as graphs, priority queues (min heap), and suitable data representations for nodes, edges, distances, and industries.
Algorithm:
Utilize Prim's algorithm for MST derivation, DFS for traversal, and priority queue operations for path prioritization.
By incorporating HTML, CSS, and Maps API into the frontend development, you can enhance the user experience by providing visual representations of nearby places, distances, and industries.
The implemented solution should optimize the water distribution path in residential areas while considering the coverage of industries and the distance travelled

10)Process Management

Agile methodology is an iterative and collaborative approach to project management and software development. It emphasizes flexibility, adaptability, and customer collaboration throughout the development process.

The core principles of Agile revolve around delivering value in short iterations called sprints, embracing change, and fostering continuous improvement. Agile teams work in a highly collaborative and cross-functional manner, with regular feedback loops and frequent communication. This methodology promotes transparency, empowers team members to make decisions, and enables rapid response to changing requirements. By embracing Agile, organizations can enhance productivity, quality, and customer satisfaction by delivering incremental and iterative solutions.

Scrum is an Agile framework that focuses on iterative and incremental product development. It involves self-organizing, cross-functional teams that work in short iterations called sprints. Daily stand-up meetings, product backlogs, and sprint reviews are key components of Scrum, promoting transparency, collaboration, and rapid delivery of value.

Through the course of this project, we have adopted Scrum Methodology of working by incorporating our project into various sprints and organizing our requirement lists into user stories and assigning them to each team member. We have implemented Scrum methodology using JIRA software.

The project consisted of four main epics.

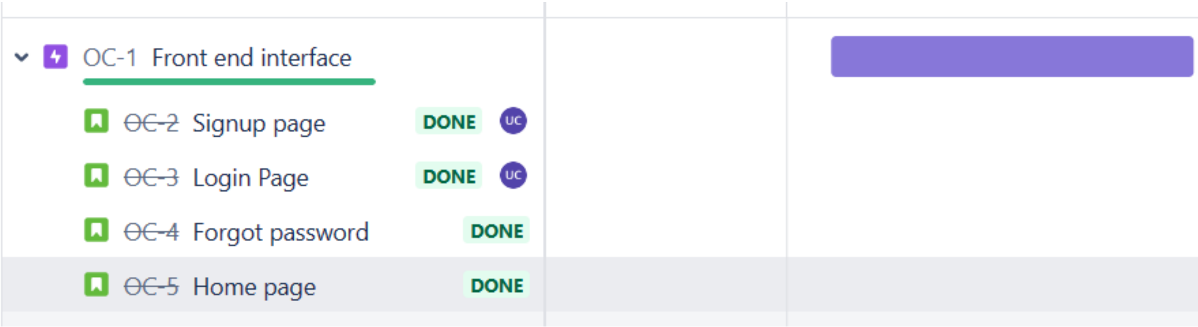
Epic 1-This Epic majorly deals with research and phase 1 - Development, involved conducting research, designing and coding the front end interface

Epic 2-Integration and Phase 2 Development, focused on integrating the DBMS with HTML pages, implementing map functionalities, and refining the user interface.

Epic 3- This Epic majorly involves working with Google Maps Platform and their provided API's, creating an account on the platform and using them

Epic 4-This Epic majorly involves the complete implementation of the backend, designing and implementing the appropriate algorithms and using the appropriate data structures.

1)Epic 1: Front End Interface



a) Sprint 1 and 2- Sign in and Login page

During the first sprint, the team focused on developing the front end interface first, so we have a clear idea on how our project will look like

● Risk Management: The team identified potential risks and devised strategies to mitigate them, ensuring the project's smooth execution.

● Medium to design UI: The team decided which front end languages to use so it would easy for us to design the front end and we decided to use HTML and CSS

● Features of UI: The team identified and documented the essential features to be incorporated into the user interface (UI) of the system..

b) Sprint 2 and 3: Forgot password and Home page

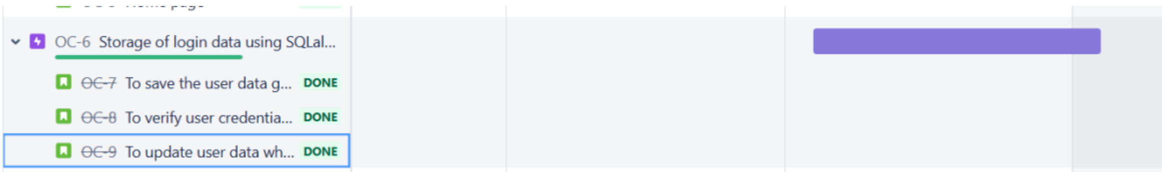
In the second and third sprint, the team focused on laying the foundation for the project's development. The following user stories were assigned:

● Prototype Development – Home page UI- This is how the projects interface will be when the user logs in or registers for the first time. We decided to keep the homepage neat and simple as much as our project demands. We also added a forgot

password(desirable) option so user can retrieve his password through verification otp. This is just to add an additional functionality to our front end
● API Research: Just after we finished discussing and designing our front end, we began analyzing on what API's we should use to ensure our problem statement is satisfied.

During this sprint, some user stories were assigned to pairs of team members who practiced pair coding. Pair coding facilitated collaboration, knowledge sharing, and better code quality.

2. Epic 2: Integration and Phase 2 Development



a) Sprint 3,4,5: Database login and integration with frontend

A database is require to stored the credentials of the users. Instead of using files which would not be an efficient process, we went with the database application provided by flask-SqlAlchemy

- Main Page and Database Integration w/ Flask: The team integrated the database and backend functionalities with the HTML code of the main page using the Flask framework.

.

- Maps API and Places API: With this sprint our Phase1 was completed and all good to go. We decide to start planning to focus on our phase 2 which is the implementation of API's.The team researched and integrated the required APIs for map functionalities and retrieving nearby accommodation information.

  During this sprint, the team collaborated closely to ensure smooth integration between the different components and functionalities.

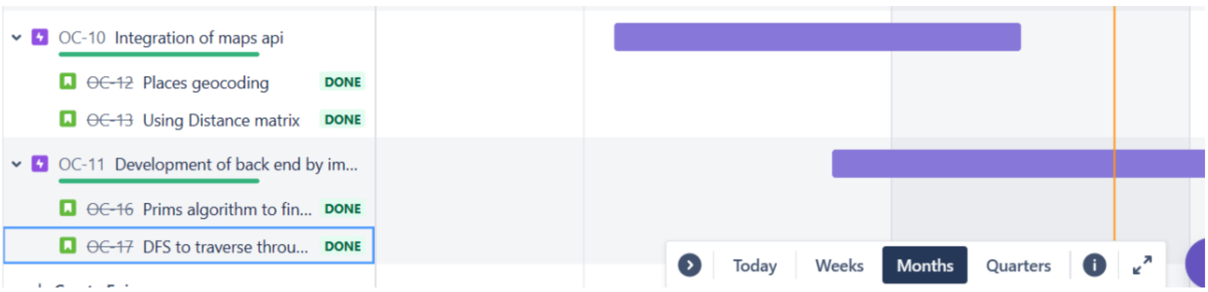
3)**Epic 3: Working with API and integration of API's**

1)Sprint 1: We discussed and went with the decision to use google maps API for our project. After thorough research we realized that no third party API was able to give us the water bodies near to a given location. Thus we use three API from google maps platform, distance matrix API, places API and geocoding API. Geocoding API is to retrieve the coordinates of a location, Places API is to search for a nearby waterbody and industry while Distance Matrix Api is to compute the distances between two places

4)Epic 4: This is the last epic and it is with regards to our backend implementation. We decided how to go about it inour first sprint meeting itself and we started implementing it immediately after Epic 3.

1)Sprint 1: We started the coding part of the backend with this sprint. Prims algorithm was deployed to derive MST out of the initial connected graph.

2) Sprint 2: After the MST was derived, we used DFS algorithm to traverse across the MST starting from source till we reach destination. All the possible paths are noted and the path with less no of industries covered and less overall cost is chosen using priority queue(min heap) implementation.

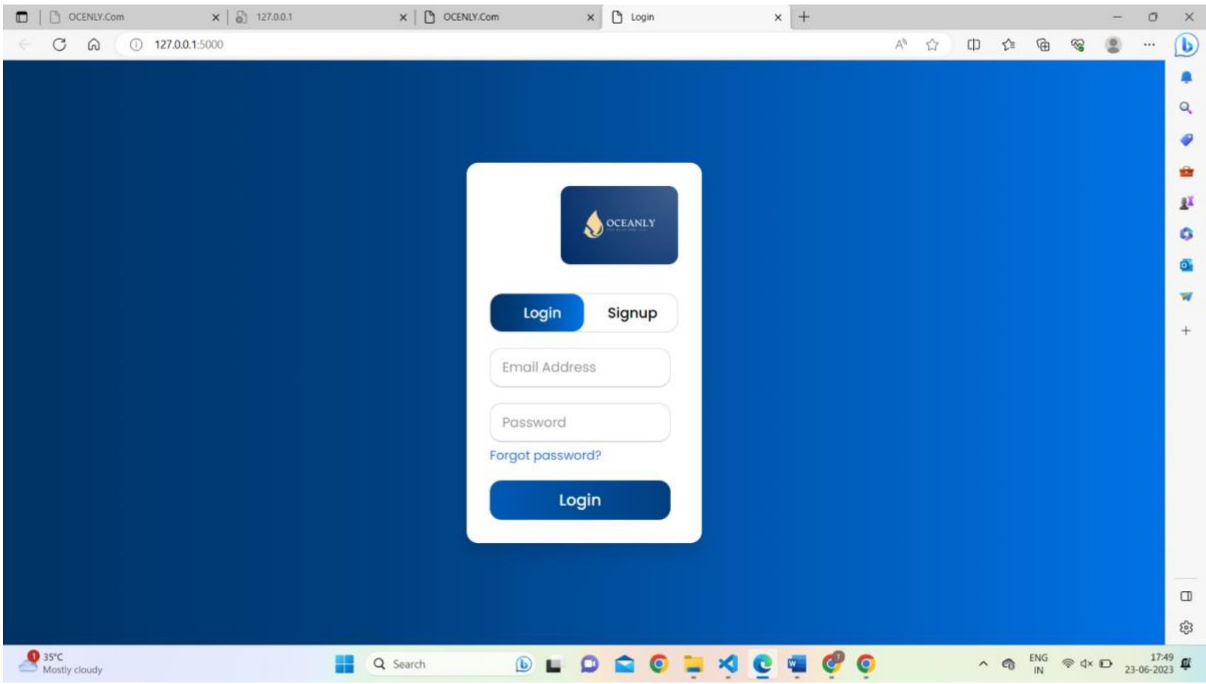


Throughout the project, the team conducted daily stand-up meetings to discuss progress, address challenges, and plan tasks. Regular communication and collaboration were facilitated through the use of JIRA, where team members updated task statuses, added comments, and addressed any queries or issues. The team also practiced continuous improvement, adapting to changing requirements and incorporating feedback during retrospectives conducted at the end of each sprint.

11)Deployment

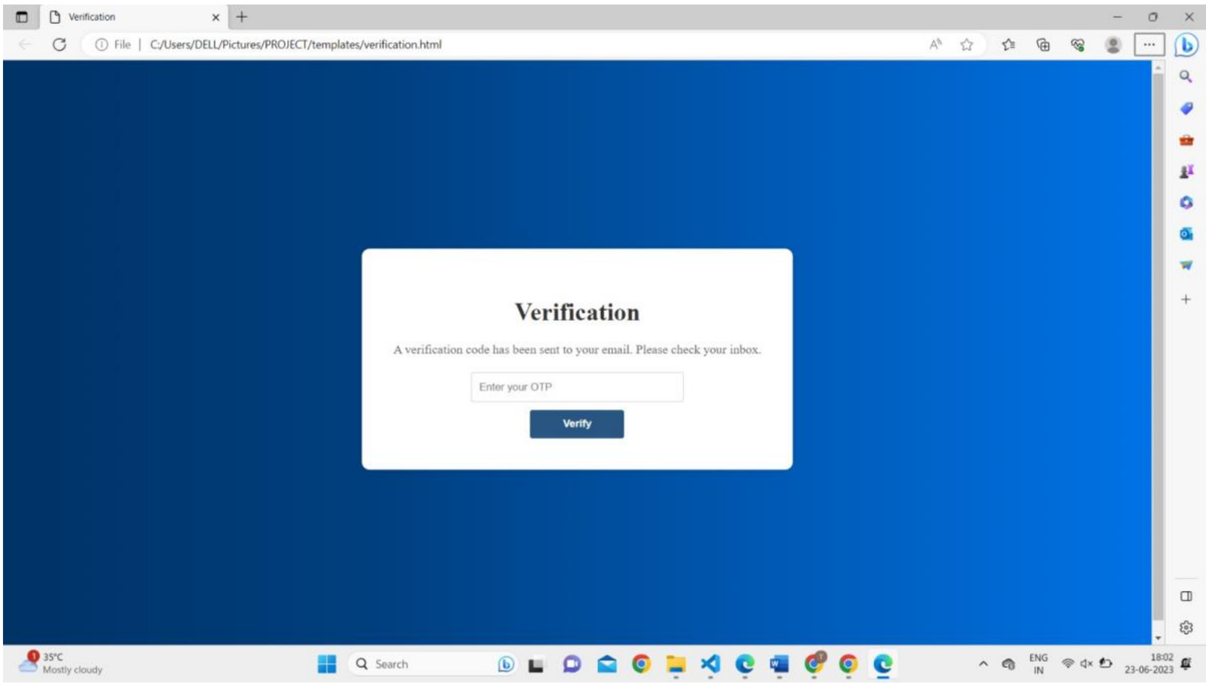
At the completion stage of the project we were able to create an application that computes a pipeline network designed in such a way that water reaches the user specified residential location in the most optimum way possible

This is how our project login page looks like:

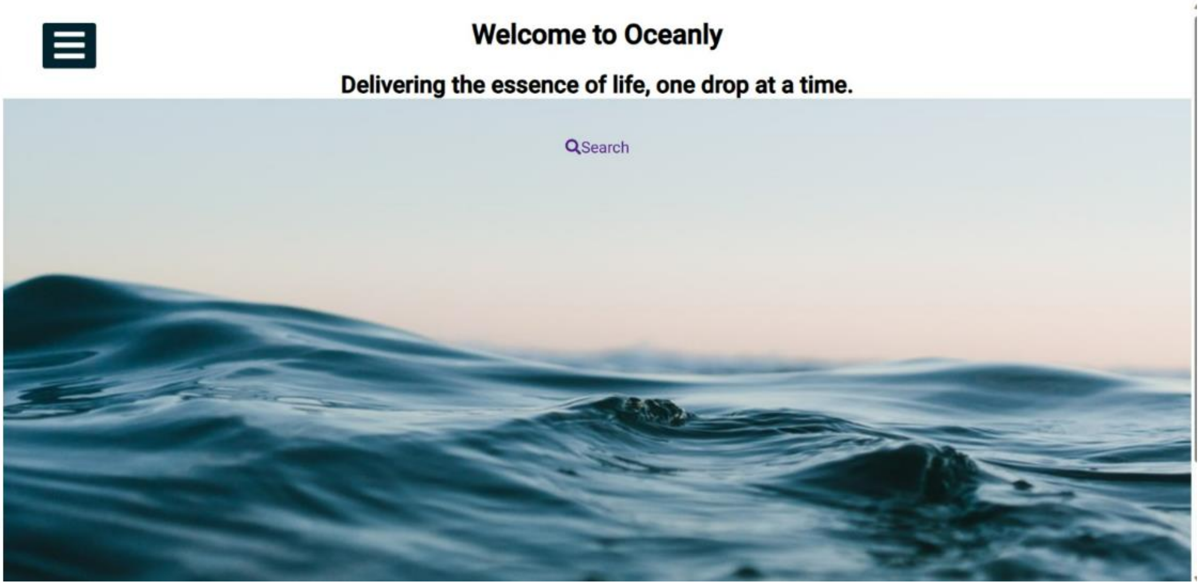


The user is given an option to either sign in or register for the first time. Only if he enters the appropriate credentials he will be able to enter the home page

Incase the user forgets his password an option is enabled where he can retrieve his forgotten password using email to which he will be given a verification OTP

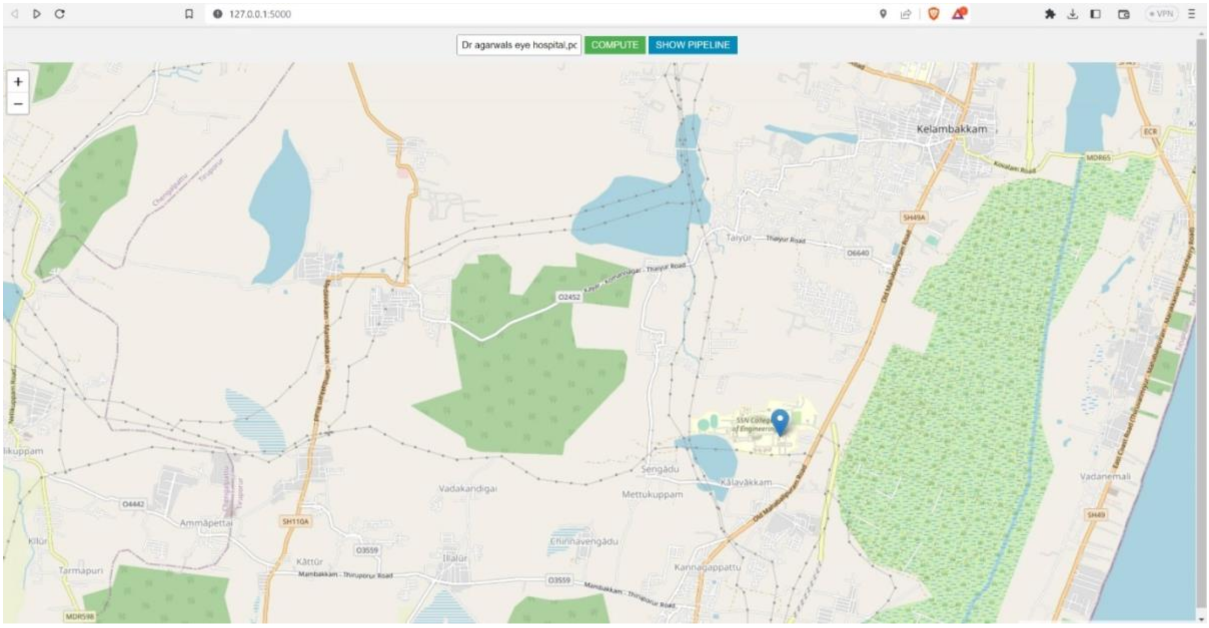


This is how the home page appears after the user registered for the first time or logins with successful credentials

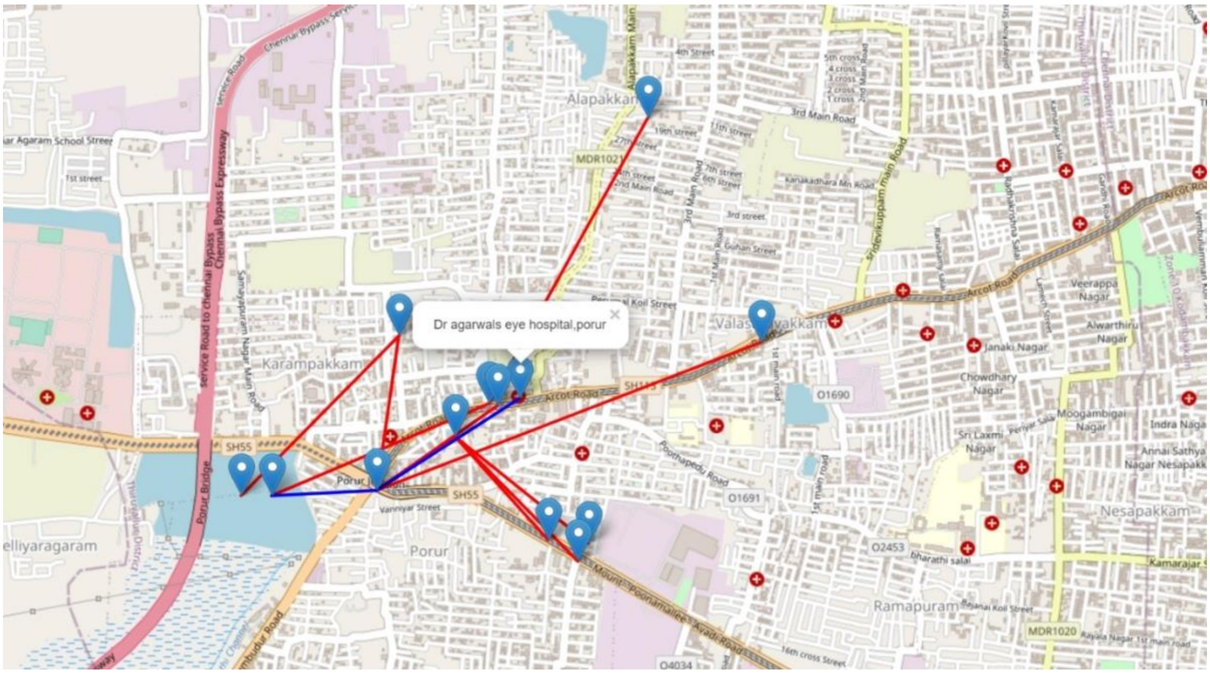


The user after entering the home page can have a variety of options to access, reading about us etc. In the home page he will be given an option to navigate to the page where his residential location will be given and his pipeline will be computed

The below is the page that will appear

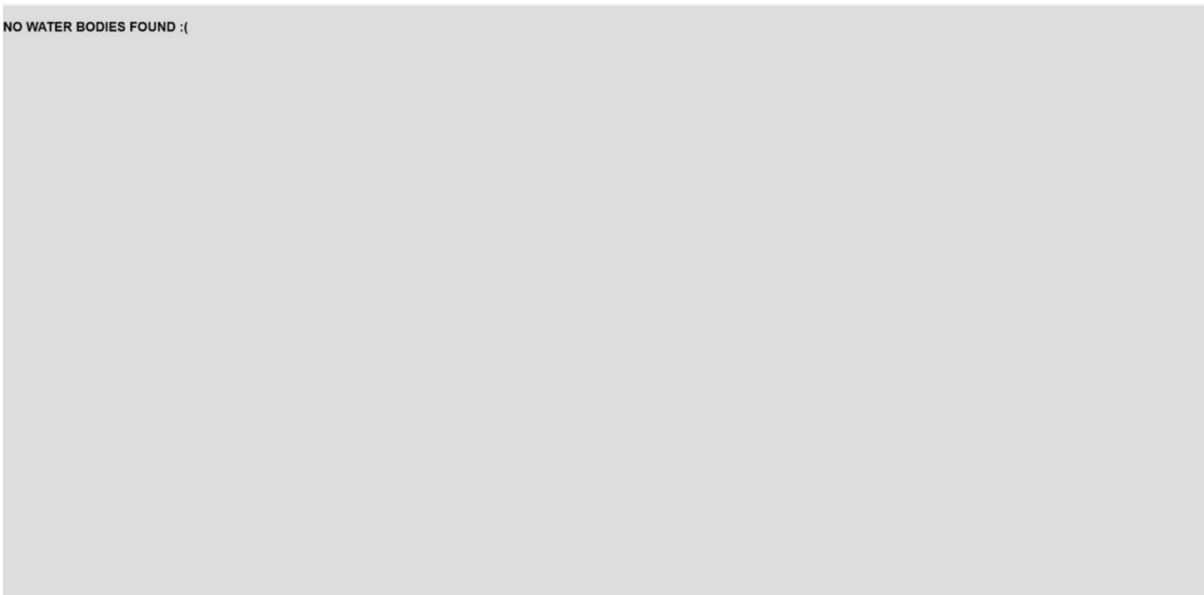


When the user enters his location and clicks on compute pipeline option, his pipeline will be computed and displayed on a map within short time



The entire network is the optimum network and the one highlighted in blue is how the water flows to our residential location with the help of valves

Incase there are no water bodies within a nearby radius we display that a pipeline network cannot be computed because of no nearby water bodies.



12)Gap Learning Reflection

Throughout this project, we embarked on a journey of gap learning, exploring various concepts and technologies to solve the problem of optimizing water distribution at the residential level. Here are some key insights we gained during this learning experience:
Graph Data Structures and Algorithms:
We delved into the intricacies of graph data structures and their applications in solving complex network problems.
Specifically, we focused on understanding the concept of the minimum spanning tree (MST) and how it can be derived using algorithms like Prim's algorithm.
This knowledge helped us design an efficient water distribution network by ensuring connectivity between nodes while minimizing costs and avoiding cycles.
Implementation of Data Structures:
To effectively represent and manipulate the network, we implemented various data structures such as priority queues and graphs.
By using a priority queue (min heap), we prioritized traversal paths based on the number of industries covered and distance traveled, leading to an optimal water distribution path.
This experience deepened our understanding of data structures and their role in solving real-world problems.
Frontend Development:
To provide a user-friendly experience, we explored frontend technologies such as HTML and CSS.
By leveraging these technologies, we created an intuitive and interactive user interface that allows users to input and visualize the water distribution network.
This frontend development experience broadened our skillset and enhanced our ability to create engaging user interfaces.
Integration of APIs:

As part of our project, we integrated the Google Maps API to derive coordinates of nearby water bodies, industries, and calculate distances.
This process introduced us to the concept of APIs, their functionality, and their role in retrieving and processing external data.
Working with the Google Maps API expanded our knowledge of how to incorporate third-party services into our projects effectively.
In conclusion, this project served as an enriching gap learning experience. We acquired in-depth knowledge of graph data structures, minimum spanning trees, priority queues, and frontend technologies like HTML and CSS. Additionally, our exposure to APIs, specifically the Google Maps API, broadened our understanding of their practical applications.

In addition to the knowledge and skills gained, we encountered some challenges and made certain assumptions during the design of our solution for this project. These factors and assumptions are worth noting:

Assumptions:
Real-time Factors: Implementing real-time aspects of the water pipeline, such as monitoring environmental factors or considering the capacity of pipelines, was beyond the scope of this project.
Therefore, we had to make assumptions to simplify the problem and focus on optimizing cost and time efficiency.
Cost Reduction and Time Efficiency: Our main goal was to reduce the cost of the pipeline while ensuring the fastest possible path to the destination. We prioritized minimizing the number of industries and areas covered along the way.
Handling Absence of Nearby Water Bodies:
In cases where there are no water bodies detected in the vicinity, we handled this scenario by not displaying any network on the interface.
Instead, we provided a clear message indicating that no water bodies were found.

Consideration for Computation Time:

It's important to note that computing the pipeline and generating the optimized path may take some time due to the involvement of real-time APIs that rely on internet connectivity. This potential delay should be considered when using the application.
By acknowledging these assumptions, we were able to streamline our approach and focus on the core objective of optimizing the water distribution network. While these assumptions may limit the scope of the solution, they allowed us to develop an efficient and practical solution for the given problem statement.

13)Conclusion:

In conclusion, our project on optimizing water distribution at the residential level has provided valuable insights into graph-based algorithms, data structures, frontend development, and API integration. By utilizing the backtracking method, we successfully derived an optimized path for water distribution by considering factors such as minimum cost and coverage of industries. The implementation of HTML, CSS and Flask allowed us to create a user-friendly interface that interacts with the backend to solve the water distribution problem.

Through this project, we deepened our understanding of graph data structures, particularly the concept of a minimum spanning tree, and its application in designing an efficient water distribution network. We gained proficiency in using priority queues, nested lists, and other data structures to represent and manipulate the network effectively. Additionally, our experience with frontend development enhanced our skills in creating visually appealing and interactive user interfaces.

The integration of the Google Maps API provided us with real-time data, allowing us to derive coordinates of water bodies and industries, calculate distances, and visualize the network on a map. This exposure to API integration expanded our knowledge of working with external services and leveraging their functionality to enhance our application.

Further Scope:

While our project has addressed the initial problem statement and provided an optimized solution for water distribution at the residential level, there are several areas for further exploration and enhancement:

Real-time Data Integration: Incorporating real-time data on environmental factors, pipeline capacities, and water availability can further improve the accuracy and efficiency of the water distribution network. This would require integrating additional APIs or data sources to retrieve and process real-time information.

Advanced Routing Algorithms: Exploring advanced graph algorithms such as Dijkstra's algorithm or A* search algorithm can provide alternative routing options based on factors like traffic congestion, road conditions, or other constraints specific to the water distribution network.

Scalability and Performance: Evaluating the scalability of the solution for larger networks or expanding it to handle multiple residential areas can be an interesting avenue for future development. Optimizing the performance of the application, particularly the computation time for generating the optimized path, would further enhance user experience.

Error Handling and Validation: Implementing robust error handling mechanisms and input validation techniques can improve the reliability and user-friendliness of the application. This includes

handling scenarios where water bodies or industries are not found in the vicinity or dealing with unexpected API errors.
By addressing these aspects, the water distribution solution can be further refined, ensuring more accurate and efficient distribution networks for residential areas while considering real-time factors and providing a seamless user experience.

14)References and bibliography

Websites referred to:

Learning about HTML and CSS in detail for frontend development-

https://www.w3schools.com/html/html_css.asp

Integrating/deploying flask with Python 3-
https://www.bing.com/search?q=integrating+python+with+flask&cvid=e81c8ffc4f5345f7b8564a3955570fdc&aqs=edge..69i57j0l8.6998j0j9&FORM=ANAB01&PC=LCTS#

Learning about Flask SQLAlchemy-
https://flask-sqlalchemy.palletsprojects.com

Google Maps Platform-
https://mapsplatform.google.com

Data Structure and Algorithms-
https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring...