DOMAIN PROJECT-1

End Term Presentation
on
Pathfinder AI

S.No.	Student's Name	Roll Number	SAP Id
01	Keertivallabh Sharma	R2142220539	500107149
02	Jyotiraditya Singh	R2142220530	500107065
03	Shrey Sharma	R2142220169	500101806
04	Rishav Singh	R2142220149	500105685

Under the guidance of

Dr. Panduranga Ravi Teja

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
Dehradun-248007 Aug-Nov, 2024

Index

Contents

Chapter 1: Abstract

The rapid growth of urban areas and the increasing need for efficient transportation solutions demand innovative approaches for route planning and construction when connecting places like big cities. The government cannot solely rely on the shortest route to connect two cities; they must also consider optimal pathways that integrate major landmarks, key locations, and important destinations. Our project aims to develop a prototype application that employs a combination of pathfinding algorithms to find the optimal path for constructing a road connecting cities. The proposed solution will aim at allowing users to interactively upload city-wide maps, select specific destinations, and mark key points such as hospitals, offices, government buildings, airports, and stations. The application will compute the most efficient route based on factors like the priority of important locations. Through this project, we aim to demonstrate the feasibility and practicality of algorithm-driven decision-making in navigation applications, providing a valuable contribution to enhancing user navigation experiences.

Chapter 2: Introduction

Efficient route planning is an important aspect of improving urban mobility and reducing travel time, especially in developing nations like India, where expanding infrastructure and urbanization pose notable challenges.

This project focuses on creating a prototype web application that applies basic pathfinding algorithms to address some of these limitations in a simplified manner. The application allows users to interact with city-wide maps, select multiple destinations, and prioritize key locations such as hospitals, government offices, and airports. Based on these inputs, the system calculates routes that deviate from straight-line paths to account for the prioritization of these locations.

While this solution is limited in scope, it aims to demonstrate how even basic algorithmic approaches can be tailored to address specific use cases like urban navigation and planning. The project involves building a simple user interface, implementing algorithms to handle user inputs and map data, and testing the application on sample datasets to evaluate its functionality.

Through this effort, the project aims to highlight the potential for developing accessible tools that consider user priorities and practical constraints, providing a small but meaningful step toward better route planning in urban settings.

Chapter 3: Problem Statement

The goal of the project is to develop an application that will visualize a path between two given locations based on prioritized locations and landmarks; as a means to aid the construction of highways.

Chapter 4: Literature

Efficient design and planning for navigation have long been studied in the context of route optimization [1]. Various algorithms and models, such as Dijkstra’s algorithm, A* search, and genetic algorithms, have been employed to enhance route efficiency, reduce travel time, and minimize costs [2]. Dijkstra's algorithm, introduced in 1959, is a widely used method for finding the shortest path between nodes in a graph and is commonly applied in navigation problems. However, its traditional form does not account for multiple dynamic factors, such as traffic density, route feasibility, or the priority of essential locations, which are crucial in real-world scenarios [3].

Research on route optimization in diverse contexts highlights the importance of considering multiple criteria beyond mere distance [4]. Studies suggest that incorporating factors such as population density, location accessibility, economic benefits, and environmental impact can lead to more effective and user-friendly navigation solutions. Multi-criteria decision-making (MCDM) models and Geographic Information Systems have increasingly been utilized to improve the accuracy and relevance of navigation planning in various geographic and socio-economic contexts.

In recent years, there has been growing interest in integrating GIS with optimization algorithms to enhance navigation applications. It provides a powerful platform for handling large spatial datasets and visualizing complex geographic information, while algorithms like Dijkstra’s can perform efficient computations on these datasets. Previous works have demonstrated that combining with optimization techniques allows for more interactive, data-driven navigation processes that consider multiple constraints and objectives [5].

This project builds on the existing literature by developing a prototype navigation application that integrates modified pathfinding algorithms with GIS-based tools. Unlike traditional approaches, our application allows users to interactively upload maps, select destinations, and mark critical locations, considering diverse criteria such as traffic density, route feasibility, and strategic importance. This approach aligns with the needs of diverse regions, where varied geographic, demographic, and economic conditions require adaptable and context-sensitive navigation tools [6]. By leveraging the strengths of both classic algorithms and modern GIS technology, this project aims to contribute to more efficient, user-friendly, and data-driven navigation solutions.

Chapter 5: Objectives

The primary objective of this minor project is to design and implement a system that addresses a real-world problem effectively by applying theoretical knowledge and leveraging modern technologies. The specific objectives of the project are as follows:

Problem Identification and Analysis

To identify and analyze a relevant real-world problem and propose an innovative and practical solution that aligns with current technological advancements.

Application of Theoretical Knowledge

To utilize concepts and principles learned during coursework, such as algorithms, data structures, and software development methodologies, in the development of the project.

System Design and Development

To design a robust and scalable system or application that efficiently fulfills the identified requirements and offers a user-friendly interface.

Integration of Modern Technologies

To incorporate advanced tools, frameworks, and technologies (such as AI, IoT, or full-stack development) into the project to enhance functionality, performance, and user experience.

Efficient Data Handling

To ensure secure and efficient processing of data using optimized algorithms and data structures for storage, retrieval, and manipulation.

User-Centric Approach

To design the system with a focus on user needs, ensuring intuitive interaction and seamless functionality.

Testing and Validation

To perform comprehensive testing and validation to ensure the system is reliable, secure, and performs as intended under various conditions.

Project Management and Team Collaboration

To develop project management skills by organizing tasks, setting milestones, and ensuring timely completion. If the project is team-based, the objective also includes enhancing collaboration and communication skills within the team.

Documentation and Reporting

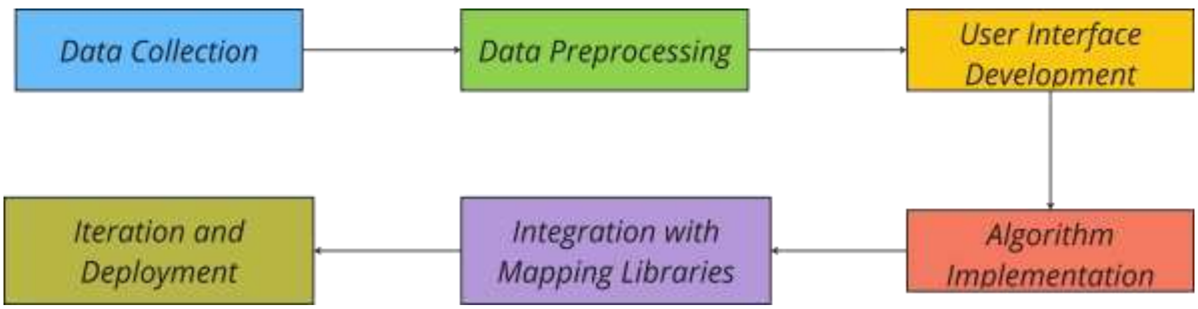
To document all stages of the project development lifecycle, including problem analysis, design, implementation, testing, and future scope, in a well-structured and detailed report.

Presentation and Demonstration of Outcomes

To effectively present the project outcomes, showcasing its innovation, technical competency, and real-world applicability, through a detailed report and practical demonstration.

Chapter 6: Methodology

The project starts with *Data Collection*, where city-wide maps are obtained from various sources like Open Street Maps (OSM), Google Maps Platform, ArcGIS etc. The next phase is *Data Pre Processing*, in this phase the obtained maps are cleaned and converted into a compatible format for the application.
Many browsers do not support .tiff, .shp, .geojson and other maps formats natively, so we used Google Maps API . Next, comes *UI development* focusing on creating an interactive user interface for convenience and easy navigation. Following this, the *Algorithm implementation* phase involves applying modified pathfinding algorithms like Dijkstra's, Bezier Curves, De Casteljau's Algorithm to determine optimal pathways for road construction. The application is then enhanced through integration to visualize the maps and roads effectively. Finally, the project undergoes iteration and deployment, where we will test it and iterate over times, to refine it to satisfaction.



Chapter 7: System Requirements

1. Software Requirements

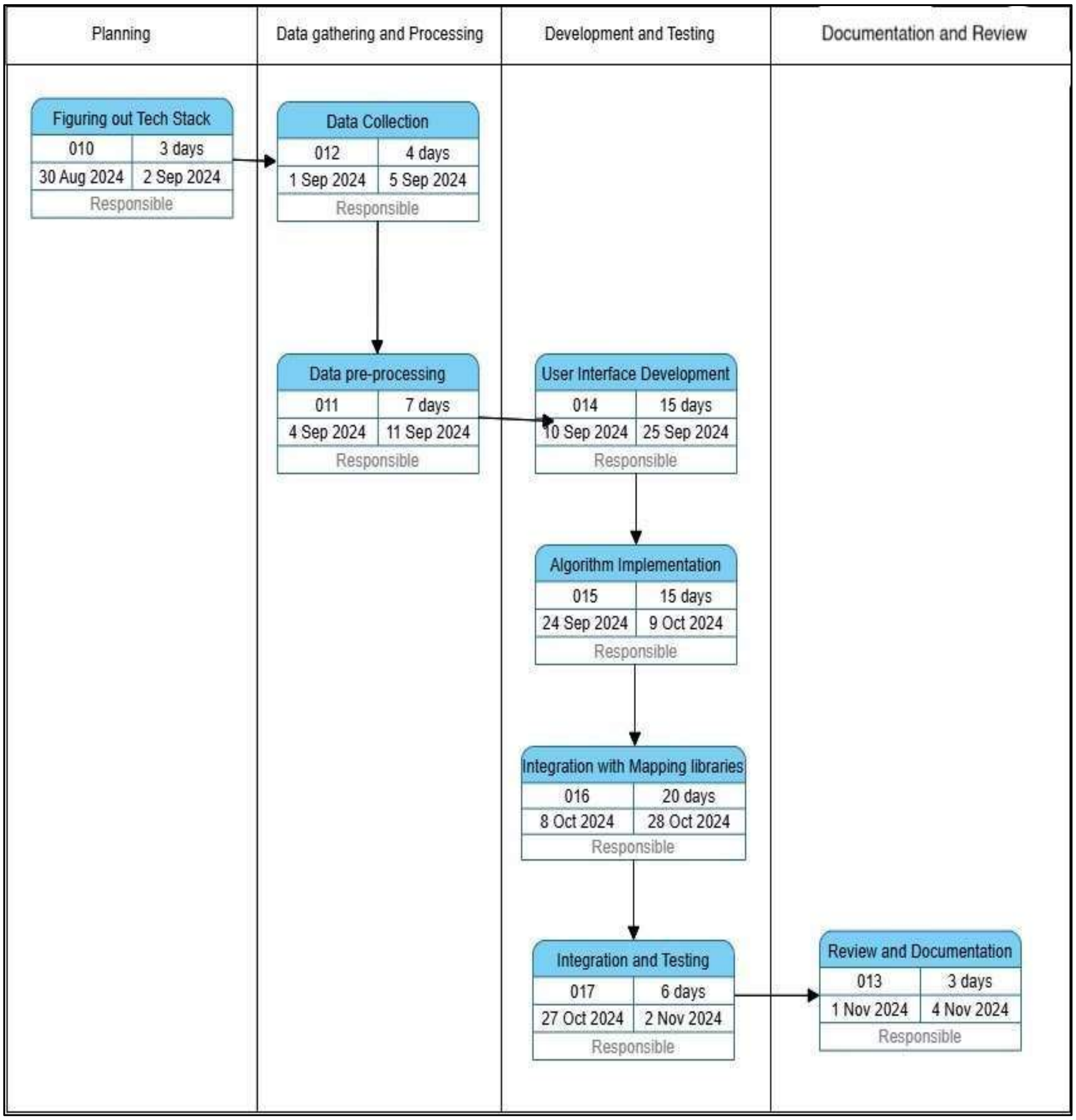
Operating System : Windows 10/8/7 (32-bit or

64-bit)/ Linux Software : Text Editor, Browser

2. Hardware Requirements

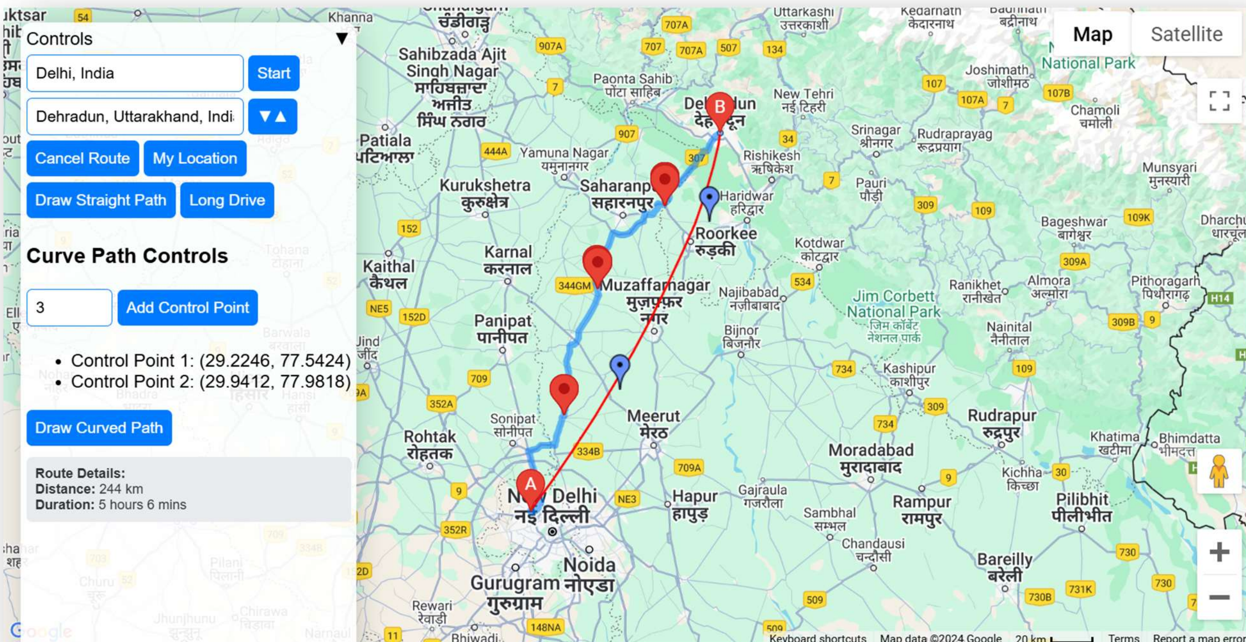
Processor	:	Dual Core 2.7 GHz or better
RAM	:	512 MB or higher
Disk Space	:	512 MB

Chapter 8: PERT Chart



Project Highlights

- **Screenshots of Project working**
  Developed a user interface for the project using OpenStreetMap (OSM) as the primary map platform. This interface allows users to interact with a city-wide map, where they can select cities to connect and mark essential landmarks like hospitals, offices, government buildings, airports, and stations.



<u>**Achieved Objectives:**</u>

1. Route Visualization

Successfully implemented route plotting on the map using Google Maps API.
Integrated functionality for displaying straight paths and curved paths (Bézier curves) between selected points.

2. Pathfinding with Dijkstra's Algorithm

Implemented Dijkstra's algorithm for calculating the shortest path between two points.
Enabled dynamic graph generation from user-added control points and markers.
Displayed the shortest path on the map as a highlighted polyline.

3. User Interaction Features

Enabled users to add and remove control points dynamically on the map.
Provided interactive options to reverse the route or adjust markers for flexible route planning.

4. Cost and Impact Estimation

Developed a report generation system that:
Estimates construction costs based on road length.
Evaluates environmental impacts like forest cover cleared.
Calculates estimated construction time and employment opportunities.

5. Map Integration

Integrated Google Maps API with support for:
Real-time geocoding of start and end points.
Autocomplete for user inputs (e.g., start and end locations).
Calculation of distances and directions between points.

6. Elevation Data Integration

Incorporated **elevation data** for trekking routes using Google Maps Elevation API.
Displayed elevation statistics, such as **total gain** and **estimated trekking time**, for user-selected paths.

7. Customizable Trekking and Long Drive Modes

Implemented a **trekking mode** with support for adding multiple trekking markers and calculating cumulative distances.
Developed a **long drive mode** for generating scenic routes and evaluating round-trip paths.

8. Map Interaction Enhancements

Allowed users to interact with the map for:
Adding **custom control points** by clicking.
Viewing **landmarks and nearby places** dynamically based on the chosen path.

9. Report Generation

Automated generation of a **detailed PDF report** with:
Route statistics, cost, and time estimations.
Impact analysis and recommendations for sustainable road construction.
Screenshot of the proposed route map.

10. Modular and Extendable Codebase

Designed a **modular graph class** to handle dynamic graph creation and shortest path calculations.
Created reusable functions for adding, removing, and updating markers, making the project extensible for future features.

Scope for further refinements:

- Improvising the algorithm associated with deviation for the roadway.
- Adding features like heatmaps, 3d integration etc.
- Options to upload custom city-wide maps, to be able to work with various formats like .tiff, .bmp, .shp, geojson and other raster and vector formats.
- Report generation mentioning details like cities selected, selected route, cost estimation, impact etc.
- Adding educational features, that gude the user through the interface of application.
- Integrate Elevation and other factorsi like community dynamics, Historical Data Analysis, job opportunities etc. for a more comprehensive view of the application.

Github Repository

https://github.com/Slyphx/Minor--1

Chapter 9: References

- https://docs.mapbox.com/help/glossary/mapbox-js/
- Google Maps Platform. (n.d.). Google Maps API Documentation.
- Project Management Institute. (2017). A Guide to the Project Management Body of Knowledge (PMBOK Guide) (6th ed.).
- https://www.youtube.com/watch?v=J7eyI1ClDOA
- https://www.iirs.gov.in/iirs/sites/default/files/pdf/2023/Overview_of_Remote_Sensing_and_GIS _Applications_E-Book.pdf
- https://www.sciencedirect.com/journal/isprs-journal-of-photogrammetry-and-remote-sensing
- [1] https://www.researchgate.net/publication/route_optimization_design
- [2] https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- [3] https://www.journalofnavigation/algorithm_limitations
- [4] https://www.sciencedirect.com/multi_criteria_navigation
- [5] https://www.researchgate.net/gis_optimization_techniques
- [6] https://www.mdpi5.com/data_driven_navigation_solutions