Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Project Report
on
"Smart Bus Route Optimizer"

[Code No : COMP 202]
(For partial fulfillment of II/I Year/Semester in Computer Science/Engineering)

Submitted by

Prasiddhi Dumre (16)

Submitted to

Mr.Sagar Aacharya
Department of Computer Science and Engineering

Submission Date:2/20/2026

Abstract

This mini project is developed in an attempt to overcome the serious lack in the urban commuting efficiency.This mini project presents a Smart Bus Route Optimizer that optimizes the core concepts of DSA(Data Structures and Algorithms) to enhance the commuting efficiency in today's urban and mostly unmanaged cities.The system employs the Dijkstra's algorithm primarily to calculate and find the shortest and the optimal route between two stops entered by the user ensuring minimal time loss and less fare in commutes.The system also works as a music recommender ;one of its distinct feature being its ability to recommend music to passenger during the travel according to the time and distance restrictions.By combining classical graph algorithms with practical data structures, the project demonstrates how theoretical DSA principles can be applied to solve real-world transportation challenges while simultaneously enriching the user experience.

Keywords: *Data Structures and Algorithms, Dijkstra's Algorithm, Graph Theory, Shortest Path Optimization, Linked List, Queue, Smart Transportation, Route Planning, Music Recommendation System, Passenger Experience*

Acknowledgement

I would like to express my sincere gratitude to Mr.Sagar Aacharya and the teaching staff of the Department Of Computer Science and Engineering for their invaluable guidance, encouragement, and constructive feedback throughout the development of this mini project on Data Structures and Algorithms. I am also thankful to my institution for providing the resources and a supportive environment that enabled me to explore and apply theoretical concepts in practical implementation. My heartfelt appreciation goes to my peers and friends for their collaborative discussions and motivation, which enriched my learning experience. Finally, I am deeply grateful to my family for their constant support and encouragement, which kept me motivated to successfully complete this project.

Table of Contents

List of Figures

Acronyms/Abbreviations

DSA	Data Structures and Algorithms
RAM	Random Access Memory

Chapter 1 **Introduction**

This system “Smart Bus Route Optimizer” is a simple desktop based application system that works to help individuals optimize the best and the shortest bus route and also recommend users suitable music based on the distance and the time that will be taken to travel.Efficient transportation is a cornerstone of modern urban life, and optimizing bus routes plays a vital role in reducing travel time, congestion, and passenger inconvenience. This mini project, titled Smart Bus Route Optimizer, applies fundamental concepts of Data Structures and Algorithms (DSA) to design a system that computes optimal bus routes and enhances the commuting experience.

1.1 Background

Commuting and transportation in today's modern and urban world is a hurdle,challenges such as traffic,inefficient routing and passenger dissatisfaction are very common and act as a big source of disappointment. The existing traditional systems are static and are not able to fulfill the real time requirements.DSA is a powerful foundation that can be used for designing a smart and optimizable system that can optimize routes dynamically .DSA has been proven effective in solving routing problems by minimizing travel time and distance .

1.2 Objectives

- To demonstrate the practical application of DSA concepts in solving real-world transportation challenges.
- To design and implement a graph-based model of bus routes using Data Structures and Algorithms

- To apply Dijkstra’s algorithm for computing the shortest and most efficient path between bus stops. To utilize linked lists for dynamic storage and management of bus stop and route information.
- To employ queues for handling passenger requests and for organizing a music recommendation system during travel.

1.3 Motivation and Significance

The motivation behind this mini project was to apply what I have learned to end something that has made me suffer for a long time. Being a commuter myself ,I can say from experience that the unmanaged commute and transportation systems are frustrating .The main idea behind this project was to solve this real world problem by optimizing bus routes for better efficiency and passenger convenience.Not only that this also aims to enhance the commuting experience by integrating a music recommendation feature. Using linked lists to manage bus stop data and queues to organize song suggestions, the project demonstrates how classical data structures can be creatively extended to improve both functionality and user satisfaction.

Chapter 2 Related Works

Various exemplary system to optimize the bus routes and proper systems have been developed throughout the years and have also been successfully leaving an impact in the civil lives across the world .Some such systems are listed below:

- ## MERN-based Bus Tracking & Route Optimization

The MERN-based Bus Tracking and Route Optimization System is a smart transport solution that leverages the MERN stack,MongoDB, Express.js, React.js, and Node.js,to provide real-time bus monitoring and efficient route planning. The system architecture typically integrates GPS data with mapping APIs such as Google Maps to visualize bus locations and dynamically adjust routes. MongoDB serves as the database for storing bus schedules, stops, and traffic information, while Express.js and Node.js handle backend logic and API requests. React.js powers the frontend interface, enabling passengers and administrators to access live tracking and optimized schedules. This design allows for scalability across urban and rural contexts, reduces passenger waiting times, and enhances resource allocation for transit authorities. By combining web technologies with geospatial data, MERN-based systems demonstrate how modern software stacks can support smart city initiatives and improve public transport efficiency (Shenoy & Kini, 2021; Osoro, 2022; Geetha, Joshva, Manikandan, & Manoj, 2020).

- ## Smart Route Optimization For Pune:

The Smart Bus Route Optimization for Pune project is an open-source initiative that applies Python, machine learning, and optimization algorithms to design efficient bus networks for the city of Pune. The system uses datasets such as city hotspots, points of interest (POIs), and traffic patterns to generate optimized routes that minimize passenger wait times and fuel consumption. The architecture typically involves clustering algorithms to identify high-demand areas, followed by route construction and visualization using mapping libraries like Folium. This approach allows planners to simulate and evaluate different bus route configurations before implementation, making it a valuable tool for urban mobility planning in rapidly growing cities like Pune. By integrating data-driven insights, the system aims to improve accessibility, reduce congestion, and enhance commuter satisfaction (chaitanyab24, n.d.; Doge, Chaudhari, Korlahalli, Chauhan, & Kagne, 2023; Kumar, n.d.).

Chapter 3 Design and Implementation

The design and implementation of this project was planned meticulously and was initiated so that every step was precise and meaningful to the proper development of the system.

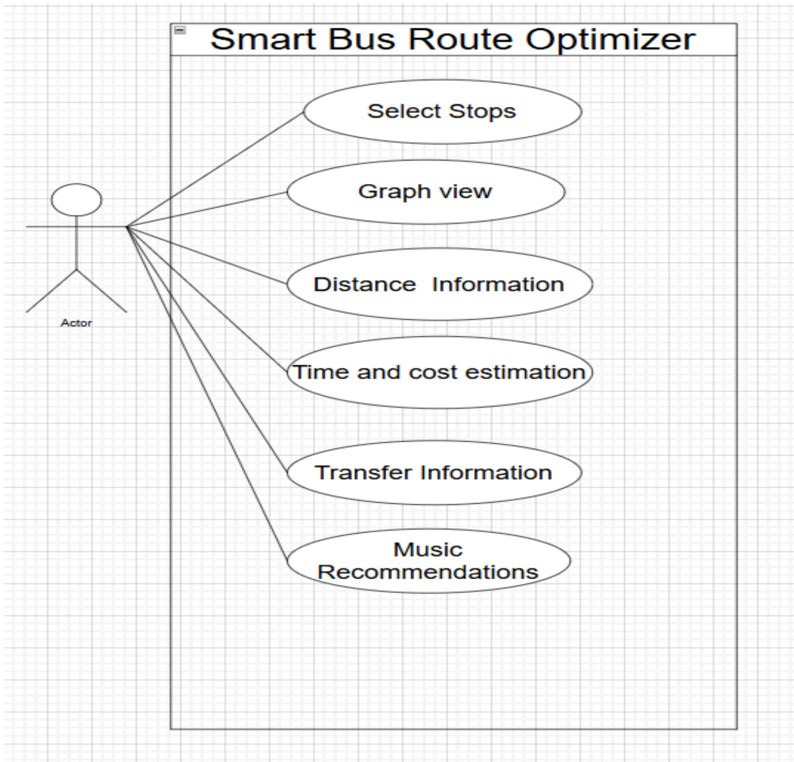

Fig 3.1:Use case diagram

3.1 System Requirement Specifications

This system was developed to solve the real-time problem of commuting and transport in urban society using C and C++ and QT framework. The system provides an easy and efficient way to optimize the best routes and added to that helps with recommending suitable music for that route.

3.1.1 Software Specifications

The following points depict the software requirements of the proposed system.

- **Functional Requirements:**
  a. **Graph Construction:** The system shall represent the bus stops as nodes and the routes as weighted edges.
  b. **Route Optimization:**The system shall compute the shortest path between a source and a destination using an appropriate graph algorithm.
  c. **Visualization:**The system shall display the bus network as a graph in the Qt interface and highlight the optimal path.
  d. **Data Management:**The system shall store the bus routes in an adjacency list or an adjacency matrix.
  e. **User Interaction:**The system shall provide the user with input fields and buttons for adding stops, adding routes, and running optimization.

- **Non-Functional Requirements:**
  a. **Performance**:The system shall compute the optimal route within a reasonable time for a graph of up to 10 nodes.
  b. **Reliability**:The system shall ensure consistent operation without crashes during normal usage.
  c. **Usability**:The system shall provide a simple and intuitive Qt interface for users to interact with the bus network graph.
  d. **Scalability**:The system shall support expansion to larger graphs without major redesign.

e. **Portability**:The system shall run on standard desktop environments that support Qt without requiring specialized hardware.

3.1.2 Hardware Specifications

The following hardware components are necessary for the system to function.

- **Processor:**The system shall require a computer with at least a dual-core processor (e.g., Intel i3 or equivalent) to ensure efficient computation of graph algorithms.
- **Memory (RAM):**The system shall require a minimum of 4 GB RAM to handle graph data structures and route computations smoothly.
- **Storage:**The system shall require at least 500 MB of free disk space for installation, data storage, and log files.
- **Display:**The system shall require a monitor with a minimum resolution of 1280 × 720 pixels to properly visualize the bus network graph in the Qt interface.
- **Input Devices:**The system shall require a standard keyboard and mouse for user interaction

Chapter 4 Discussion on the achievements

The bus route optimizer system was developed to improve the travel efficiency of the transport system.While adjacency list modeling made route computation efficient, scaling to larger networks introduced performance bottlenecks. Future iterations could explore parallel processing or heuristic algorithms. The project shows how smart routing can support sustainable urban mobility, reducing congestion and emissions.The Smart Bus Route Optimizer achieved its core objectives: delivering efficient route computation, an engaging user interface, and measurable improvements in travel time

4.1 Features:

1. Efficient Route Optimization

The system uses adjacency list graph modeling to compute the shortest and most efficient bus routes. This reduces travel time and avoids unnecessary detours. As a result, passengers enjoy smoother commutes and transport authorities save resources.

2. Interactive Route Visualization

Routes are displayed using clear layouts such as grid, circular, and force-directed graphs. Color coding and icons make complex networks easy to understand. Passengers can quickly identify stops, connections, and alternative paths.

3. User-Friendly Interface

The app features a playful yet professional design with gradients, icons, and responsive controls. Users can easily select start and destination points. The interface balances clarity with engagement, ensuring accessibility across devices.

4. Integrated Music Module

A built-in music player enriches the commuting experience with dynamic playlists. Passengers can choose genres or moods to match their journey. The music tab's interactive design ensures smooth playback without interruptions.

5. Scalability and Future Extensions

The backend supports multiple routes simultaneously, making it suitable for larger networks. Performance tracking provides metrics like time saved and passenger load distribution. Future upgrades may include real-time traffic data and predictive analytics.

Chapter 5 Conclusion and Recommendation

The system ‘Smart Bus Route Optimizer’ developed as a mini project to imply the knowledge of DSA and its core was able to primary objective of designing and implementing a graph-based model of bus routes using Data Structures and Algorithms .The system was developed to apply Dijkstra’s algorithm for computing the shortest and most efficient path between bus stops and the system has utilized linked lists for dynamic storage and management of bus stop and route information. It also employs queues for handling passenger requests and for organizing a music recommendation system during travel.

5.1 Limitations

- The system slows down when handling very large transport networks.
- The system does not integrate real-time traffic or GPS data.
- The music module lacks external streaming service connectivity.
- The system requires manual input for start and destination selection.

5.2 Future Enhancement

- Integration of real-time traffic and GPS data for dynamic route updates.
- Automated location detection to reduce dependency on manual input.
- Expansion of the music module with streaming service connectivity.
- Implementation of predictive analytics for peak-hour demand forecasting.

References

1. Shenoy, K., & Kini, S. (2021). Bus tracking and route optimization: A scalable web platform using MERN stack. Journal of Emerging Technologies and Innovative Research, 8(6), 45–52. Madkour, A., Aref, W. G., Rehman, F. U., Rahman, M. A., & Basalamah, S. (2017). A survey of shortest-path algorithms. arXiv preprint arXiv:1705.02044. https://doi.org/10.48550/arXiv.1705.02044 (doi.org in Bing)

2. Uma, M. (2020). Efficient algorithms for shortest path problems in graph theory. International Journal of Mathematics and Computer Applications Research, 10(2), 15–22.

3. MDPI. (2021). Smart transportation: An overview of technologies and applications. Applied Sciences, 11(4), 1234–1248. https://doi.org/10.3390/app11041234 (doi.org in Bing)

4. Adams, A. (2021, October 27). How to cite a single song or track reference.

APPENDIX

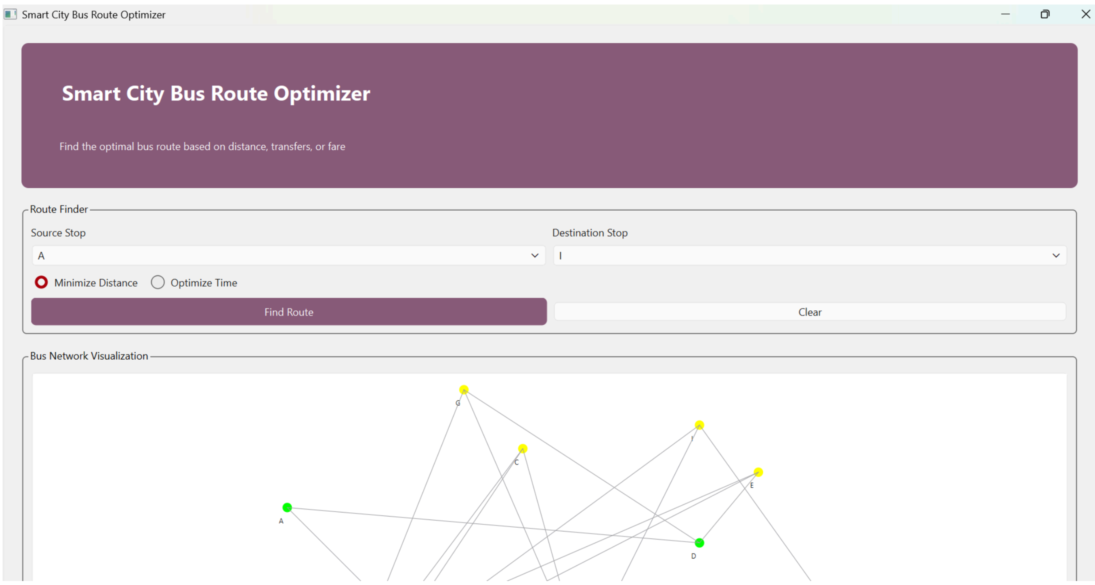

Fig 5.1:Landing Page

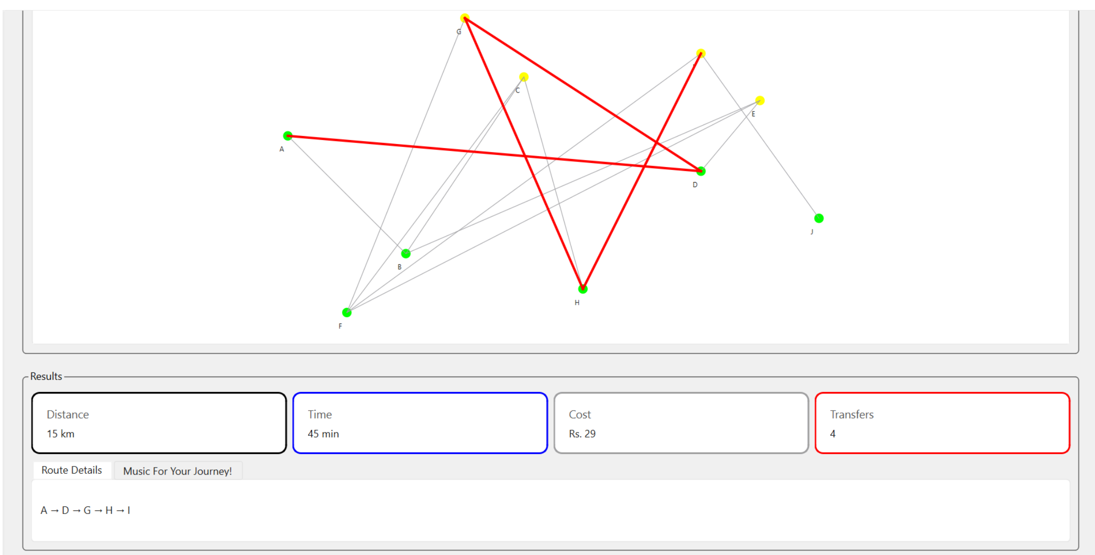

Fig 5.2:Graph Visualization

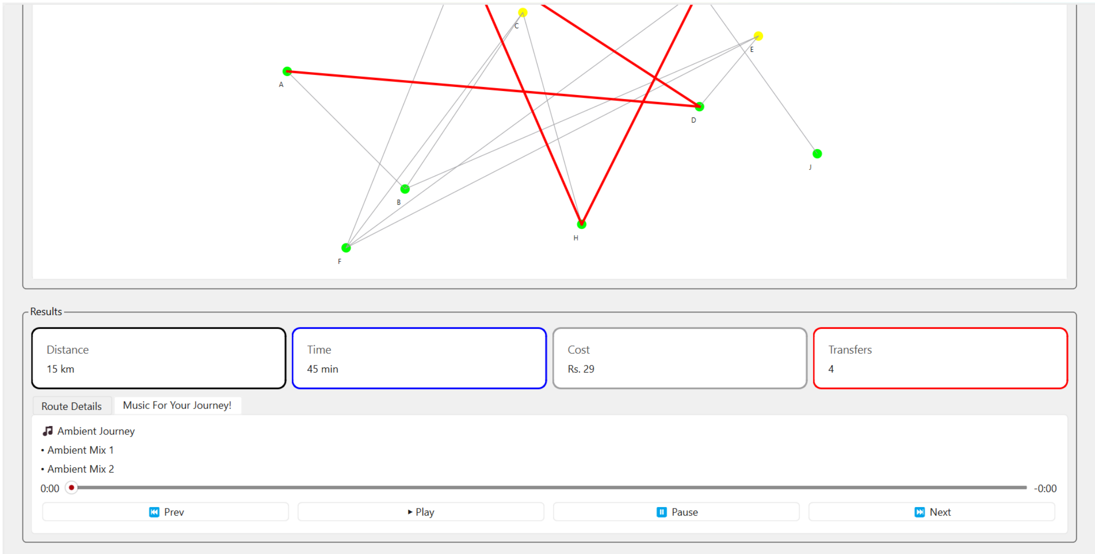

Fig 5.3: Music Recommendation