REAL TIME TRAFFIC ANALYSIS

## A PROJECT REPORT

***Submitted by***

**PRABUKUMAR .V(2303811720521039)**

***in partial fulfillment of requirements for the award of the course***

**CGA1121 – DATA STRUCTURES**

***in***

**INFORMATION TECHNOLOGY**

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

**SAMAYAPURAM – 621 112**

## May,2024

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**

**SAMAYAPURAM – 621 112**

# BONAFIDE CERTIFICATE

Certified that this project report titled **“REAL TIME TRAFFIC ANALYSIS”** is the bonafide work of **PRABUKUMAR V(2303811720521039),** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

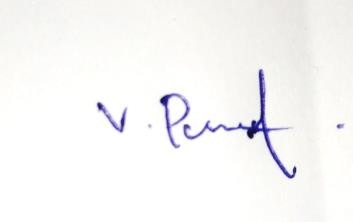
|  |  |
| --- | --- |
| **SIGNATURE**  Mrs.A.Delphin Carolina Rani, M.E.,Ph.D., **HEAD OF THE DEPARTMENT** PROFESSOR  Department of CSE  K. Ramakrishnan College of Technology (Autonomous)  Samayapuram – 621 112. | **SIGNATURE** Mr.D.P.Devan M.E., **SUPERVISOR** ASSISTANT PROFESSOR  Department of CSE  K.Ramakrishnan College of Technology (Autonomous)  Samayapuram – 621 112. |

Submitted for the viva-voce examination held on 16.06.2024

**DECLARATION**

I declare that the project report on **“REAL TIME TRAFFIC ANALYSIS”** is the result of original work done by my and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfillment of the requirement of the award of the course **CGA1121- DATA STRUCTURES.**

### Signature



PRABUKUMAR V

Place: Samayapuram Date: 16.06.2024

iii

**ACKNOWLEDGEMENT**

It is with great pride that I express our gratitude and indebtedness to our institution, “**K. Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,**

Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr. A. DELPHIN CAROLINA RANI M.E., Ph.D.,** Head of

the Department of COMPUTER SCIENCE AND ENGINEERING, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mr.D.P.Devan M.E.,,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

#### VISION OF THE INSTITUTION

To emerge as a leader among the top institutions in the field of technical education.

#### MISSION OF THE INSTITUTION

Produce smart technocrats with empirical knowledge who can surmount the global challenges.

Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students.

Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations.

#### VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

#### MISSION OF DEPARTMENT

**M1: Industry Specific:** To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

**M2: Research:** To prepare students for research-oriented activities.

**M3: Society:**To empower students with the required skills to solve complex technological problems ofsociety.

#### PROGRAM EDUCATIONAL OBJECTIVES

1. **PEO1: Domain Knowledge**

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

#### PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur. **3. PEO3: Ethics and Values**

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

#### PROGRAM SPECIFIC OUTCOMES (PSOs)

**PSO 1: Domain Knowledge**

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

#### PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

#### PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

#### PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## ABSTRACT

Real time traffic analysis involves the continuous monitoring, collection, and analysis of traffic data to extract meaningful insights and facilitate informed decision-making. Various data source contribute to real -time traffic analysis, including traffic cameras, GPS devices, road sensors, and mobile apps. Integration of diverse data streams enhance that accuracy and reliability of traffic information. The abstract begins by elucidating the significance of real-time traffic analysis in enhancing transportation efficiency, safety, and sustainability. It highlights the challenges posed by urbanization, population growth, and the increasing number of vehicles on roads, necessitating advanced traffic management strategies. Next, the abstract discusses the methodologies employed in real-time traffic analysis, including sensor technologies, data collection techniques, and data processing algorithms. It explores the utilization of various sensors such as loop detectors, cameras, GPS devices, and connected vehicles to gather traffic data in real-time. Furthermore, it delves into the application of advanced data processing techniques such as machine learning, data fusion, and predictive analytics to extract actionable insights from the collected data.

Moreover, the abstract examines the diverse applications of real-time traffic analysis across different domains. It discusses its role in traffic management, congestion mitigation, incident detection, route optimization, and intelligent transportation systems (ITS). Additionally, it explores its applications in urban planning, emergency response, environmental monitoring, and smart city initiatives.

In conclusion, the abstract emphasizes the importance of real-time traffic analysis in addressing the complexities of modern transportation systems. It underscores the need for continued research, innovation, and collaboration to develop robust and scalable solutions for managing urban mobility effectively.

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **CHAPTER** | **TITLE** | **PAGENO** |
| [**ABSTRACT**](file://localhost/C:/Users/Jayanan/Downloads/h)  [**LIST OF FIGURES**](file://localhost/C:/Users/Jayanan/Downloads/h)  [**LIST OF ABBREVIATIONS**](file://localhost/C:/Users/Jayanan/Downloads/h) | | [vii](file://localhost/C:/Users/Jayanan/Downloads/h) x  xi |

1. **INTRODUCTION**

1.1.1 Introduction 1

* 1. Purpose And Importance 1
  2. [Objectives 2](#_bookmark0)
  3. [Project Summarization 2](#_bookmark1)

1. PROJECT METHODOLOGY
   1. [Introduction to System Architecture 3](#_bookmark2)
   2. [Detailed System Architecture Diagram 4](#_bookmark3)
2. DATA STRUCTURE PREFERANCE
   1. [Explanation of why a Singly Linked List was chosen 5](#_bookmark4)
   2. [Comparison with Other Data Structures 6](#_bookmark5)
   3. [Advantages and disadvantages of using a SLL 6](#_bookmark6)
3. DATA STRUCTURE METHODOLOGY
   1. [Singly linked list 7](#_bookmark7)
      1. Key features 7
   2. [Node Structure 7](#_bookmark8)

4.3 [7](#_bookmark9)

1. MODULES
   1. Adding of vehicles 8
   2. Commands 8
   3. Storing information 8

|  |  |
| --- | --- |
| 5.4 Detects the potential security threads | 8 |
| **6 CONCLUSION & FUTURE SCOPE** |  |
| 6.1 Conclusion | 9 |
| 6.2 Future Scope | 9 |
| **APPENDICES** |  |
| Appendix A-Source code | 10 |
| Appendix B -Screen shots | 19 |
| **LIST OF FIGURES** | |
| **FIGURE NO TITLE**  2.1 Architecture Diagram | **PAGENO.**  4 |

# LIST OF ABBREVIATIONS

## ABBREVIATIONS

GPS-Global Positioning System

ITS-Intelligent Transportation System

xi

**CHAPTER 1 INTRODUCTION**

## INTRODUCTION TO PROJECT

The real-time traffic analysis project aims to gather data from GPS, cameras, and sensors, using queues for data streams, hash tables for quick lookups, and graphs to represent road networks. It cleans and normalizes data for consistency and employs shortest path and flow algorithms for route optimization and congestion prediction. The system integrates predictive models for traffic forecasting and provides real-time updates. Visualizations include interactive dashboards and maps for clear representation of traffic conditions.

## PROPOSED AND IMPORTANCE OF THE PROJECT

The project aims to develop a real-time traffic analysis system that collects data from various sources like GPS, traffic cameras, and sensors. Utilizing data structures such as queues, hash tables, and graphs, the system will efficiently process and store traffic data. It will implement algorithms for route optimization, congestion prediction, and provide real-time updates and visualizations through interactive dashboards and maps. The system will also use predictive models to forecast traffic conditions, enhancing traffic management and urban mobility.

Importance of the Project:

Improved Traffic Flow:

By optimizing routes and predicting congestion, the project can significantly reduce traffic jams and travel times.

Enhanced Safety:

Real-time updates and alerts can help drivers avoid accidents and hazardous conditions, improving overall road safety.Environmental Benefits: Smoother traffic flow reduces vehicle emissions and fuel consumption, contributing to a cleaner environment.

Economic Efficiency:

Reduced congestion leads to lower transportation costs and increased productivity as goods and services move more efficiently.Urban Planning: Data-driven insights can aid city planners in designing better road networks and traffic management strategies.

Emergency Response:

Faster route optimization for emergency vehicles can save lives by reducing

response times.

## OBJECTIVES

1. Data Collection
2. Data Processing and Cleaning
3. Traffic Analysis
4. Real-Time Updates

## PROJECT SUMMARIZATION

The real-time traffic analysis project is a comprehensive initiative aimed at revolutionizing how we understand and manage urban transportation. By harnessing cutting-edge sensor technologies and advanced data processing techniques, the project endeavors to provide timely and accurate insights into traffic conditions, enabling transportation stakeholders to make informed decisions and optimize traffic flow. Through the deployment of sensors such as loop detectors, cameras, GPS devices, and connected vehicles, the project aims to continuously monitor traffic in real-time, facilitating rapid incident detection, congestion management, and route optimization. Moreover, by integrating real-time traffic analysis with intelligent transportation systems (ITS) components, the project seeks to create a seamless and responsive transportation ecosystem that enhances safety, efficiency, and sustainability on urban roads.

The outcomes of the real-time traffic analysis project are expected to be transformative, with significant benefits for individuals, businesses, and communities. Improved traffic flow and reduced congestion will lead to shorter travel times, increased productivity, and minimized environmental impact. Furthermore, by providing commuters and logistics operators with up-to- date information on traffic conditions and alternative routes, the project will optimize route

planning and navigation, further enhancing efficiency and convenience. Overall, the project aims to usher in a new era of transportation management, leveraging real-time data and analytics to create smarter, more resilient, and sustainable urban transportation networks.

### CHAPTER 2

**PROJECT METHODOLOGY**

## INTRODUCTION TO SYSTEM ARCHITECTURE

The architecture for the real-time traffic analysis project is designed to provide a holistic and agile framework for monitoring, analyzing, and managing traffic conditions in urban areas. By leveraging advanced sensor technologies, data processing techniques, and user-centric applications, the architecture empowers transportation stakeholders to make data-driven decisions, optimize traffic flow, and enhance safety and efficiency on urban roads.

## High-Level System Architecture

The high-level system architecture for the real time traffic analysis typically consists of several key components:

1. Homepage
2. Commands (iii)Information storage

(iv) Detects security threats

## Components of the System Architecture A.Homepage:

This homepage involves the user to enter the options.

## Commands:

The commands refers the options used in the project to enter the vehicles details like vehicle entered time, number of vehicle, average speed of the vehicle. These all comes under the category of gathering information about the vehicle and commands for display, detecting potential security threads are included.

## Storing information:

Real-time traffic analysis information storage involves the continual collection, processing, and storage of data from diverse sources such as traffic cameras, sensors, and GPS devices. This data undergoes real-time processing to extract insights on traffic patterns, congestion, accidents, and weather conditions, which are then stored in databases or data warehouses.

## Detect potential threads:

Potential threads for real-time traffic analysis encompass predictive analytics for anticipating congestion, anomaly detection systems for swift identification of irregularities, dynamic route optimization algorithms for efficient travel, integration of multi-modal transportation data for comprehensive solutions, improved coordination between traffic management.

## DETAILED SYSTEM ARCHITECTURE DIAGRAM

Include a diagram that visually represents the system architecture. The diagram should depict how each component interacts with the others.

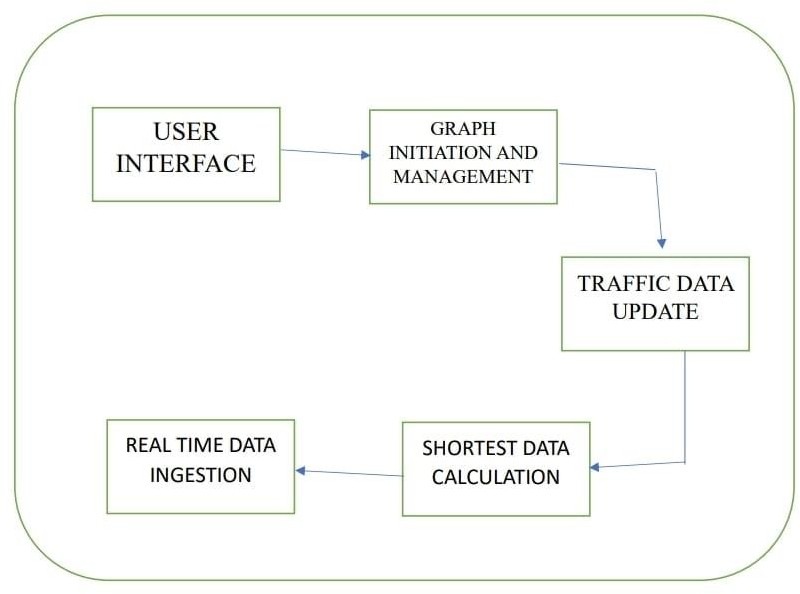


Fig.2.2 (Architecture Diagram)

Arrows or lines indicate the direction of data flow between blocks, illustrating how information moves through the system. Annotations within blocks may detail inputs, outputs, or processing steps, providing clarity on the system's operation.

### CHAPTER 3

**DATA STRUCTURE PREFERANCE**

### 3. 1 EXPLANATION OF WHY A SINGLY LINKED LIST WAS CHOSEN

Singly linked lists use memory efficiently since they only require additional memory for storing pointers to the next element, rather than maintaining additional pointers for previous elements as in doubly linked lists. This can be advantageous when dealing with large volumes of real-time traffic data, where memory optimization is important.

## Dynamic Data Handling:

Dynamic Data Handling: Real-time traffic data can be highly dynamic, with frequent updates and changes in traffic conditions. Singly linked lists provide efficient support for dynamic data handling, allowing for easy insertion and deletion of data elements as traffic conditions change.

* + 1. **Ease of Implementation:** Singly linked lists are relatively simple to implement compared to other data structures like trees or graphs. This simplicity makes them a suitable choice for scenarios where the focus is on quickly developing and deploying real-time traffic analysis systems**.**
    2. **Simplicity of Implementation:** This is a basic implementation of a singly linked list. While more advanced features and optimizations can be added, this simple implementation demonstrates the fundamental concepts of how a singly linked list works..

## Flexibility :

Singly linked lists can dynamically adjust their size as elements are inserted or deleted. Unlike arrays, which have a fixed size, singly linked lists can grow or shrink based on the number of elements they contain. This flexibility makes them suitable for scenarios where the number of elements is unknown or changes frequently, such as real-time traffic analysis where new events occur continuously.

* + 1. **Space Efficiency :** Singly linked lists offer efficient insertion and deletion operations at the beginning of the list or after a specific node. This can be advantageous in real-time traffic analysis scenarios where new events need to be quickly added to or removed from the data structure.

## COMPARISON WITH OTHER DATA STRUCTURES

Arrays provide direct access to elements based on their indices, resulting in faster access times compared to linked lists. However, arrays have a fixed size and require pre-allocation of memory, which can be inefficient if the size needs to change dynamically.

Doubly linked lists offer backward traversal in addition to forward traversal, making them suitable for scenarios where bidirectional access is required. However, they have higher memory overhead due to the additional pointers linking nodes.

## ADVANTAGES AND DISADVANTAGES OF USING A SLL

* + 1. **Advantages of Using a Singly Linked List:**

Singly linked lists allocate memory dynamically, meaning they can grow or shrink in size as needed without the need for a predefined memory block. This flexibility is particularly useful for applications where the amount of data can vary significantly over time.Insertions and deletions can be performed efficiently, especially when adding or removing elements at the beginning of the list. This is because these operations only require updating a few pointers and do not involve shifting elements, as is necessary in arrays.

## Disadvantages of Using a Singly Linked List:

Singly linked lists only allow sequential access, meaning elements must be accessed in order, starting from the head of the list. This results in slower search times compared to structures like arrays or hash tables, which support direct access.Each node in a singly linked list requires additional memory for the pointer to the next node. While the overhead is less than in doubly linked lists, it is still significant compared to arrays, which do not require pointers.

### CHAPTER -4

**DATA STRUCTURE METHODOLOGY**

* 1. **SINGLY LINKED LIST**

Linked lists provide dynamic memory allocation, enabling flexible storage and efficient memory usage, especially when the size is unknown. They support dynamic insertion and deletion, making them suitable for scenarios requiring frequent modifications. Linked lists have lower memory overhead per element compared to arrays. However, they require linear traversal for search operations, may exhibit poorer cache performance, and entail higher complexity in management due to pointer manipulation.

## Key features of a Singly linked list :

1. Dynamic memory allocation enables flexible storage and efficient memory usage.
2. Support for dynamic insertion and deletion, suitable for scenarios requiring frequent modifications.

## NODE STRUCTURE

**Data**: This part of the node structure holds the actual information or payload associated with the node. In the context of real-time traffic analysis, the data stored in each node might represent details about a specific traffic event, such as the timestamp, event type, location, and any additional information relevant to the event.

**Next Pointer**: This component of the node structure is a reference or pointer that points to the next node in the linked list. It establishes the connection between nodes, allowing traversal from one node to the next. In a singly linked list, each node has only one reference, pointing to the next node in the sequence.

### INITIALIZATION, INSERTION & DELETION

Insertion and Deletion: Singly linked lists offer efficient insertion and deletionoperations at the beginning of the list or after a specific node. This can be advantageous in real-time traffic analysis scenarios where new events need to be quickly added to or removed from the data structure.

Traversal: Traversing a singly linked list requires sequential access, which can be less efficient than direct access to elements in an array or other data structures. However, with proper indexing or caching mechanisms, the impact of traversal time can be mitigated.

### CHAPTER-5 MODULES

* 1. **Adding vehicles**

This module operates by collecting and processing data from diverse sources, including GPS devices, traffic cameras, and road sensors, to extract key information such as vehicle speed, location, and density. By seamlessly integrating with existing traffic analysis frameworks, it enhances the understanding of traffic patterns and congestion levels.

### Display of Vehicle

Displaying vehicle data within the framework of real-time traffic analysis serves as a vital component in visualizing and understanding the dynamics of urban mobility. Through intuitive graphical interfaces and interactive maps, this display provides a comprehensive overview of various aspects of traffic, including vehicle speed, density, and flow. Real-time updates ensure that users have access to the most current information about road conditions and congestion levels

### Average speed of the Vehicle

In a real-time traffic analysis system, computing the average speed of vehicles is essential for understanding traffic dynamics and optimizing transportation networks. This process begins with continuous data collection from various sources like GPS devices and traffic sensors, capturing vehicle speeds at specific intervals. Through meticulous data processing, anomalies are identified and filtered out to ensure accuracy.

### Detect Security Threats

Analyzing the speed of vehicles in real-time traffic analysis systems presents several potential security threats that demand vigilant mitigation strategies. One critical concern is the risk of data manipulation, where adversaries could tamper with vehicle speed data to fabricate false traffic patterns or disrupt traffic management operations.

### CHAPTER 6 CONCLUSION & FUTURE SCOPE

* 1. **CONCLUSION**

In conclusion, the future of real-time traffic analysis holds significant promise for revolutionizing transportation management and urban planning. Leveraging the efficiency and adaptability of singly linked lists, this technology offers dynamic solutions for optimizing traffic flow, enhancing route guidance systems, and facilitating data-driven decision-making in smart cities. By harnessing the power of predictive analytics, authorities can anticipate and preempt traffic congestion, ultimately improving overall mobility and quality of life for urban residents. As advancements continue, real-time traffic analysis stands poised to play a pivotal role in shaping the cities of tomorrow.

## FUTURE SCOPE

Real-time traffic analysis leveraging singly linked lists presents a promising frontier in transportation innovation. By harnessing this technology, traffic management systems can

dynamically optimize signal timings and route guidance, leading to smoother traffic flows and reduced congestion. In the broader context of smart city development, these systems enable data- driven decision-making, enhancing urban mobility and livability. Moreover, the predictive analytics capabilities afforded by real-time traffic analysis empower authorities to anticipate and mitigate future traffic bottlenecks, fostering more efficient transportation networks.

#include <stdio.h>

#include <stdlib.h>

#include <limits.h> #include <stdbool.h> #include <time.h>

#define MAX\_NODES 100 #define INF INT\_MAX typedef struct Node {

int vertex; int weight;

struct Node\* next;

} Node;

typedef struct Graph { int numVertices; Node\*\* adjLists;

} Graph;

### APPENDIX

**A-SOURCE CODE**

// Function to create a new node Node\* createNode(int v, int weight) {

Node\* newNode = (Node\*)malloc(sizeof(Node)); newNode->vertex = v;

newNode->weight = weight; newNode->next = NULL; return newNode;

}

// Function to create a graph Graph\* createGraph(int vertices) {

Graph\* graph = (Graph\*)malloc(sizeof(Graph)); graph->numVertices = vertices;

graph->adjLists = (Node\*)malloc(vertices \* sizeof(Node)); for (int i = 0; i < vertices; i++)

graph->adjLists[i] = NULL;

return graph;

}

// Function to add an edge to the graph

void addEdge(Graph\* graph, int src, int dest, int weight) { Node\* newNode = createNode(dest, weight);

newNode->next = graph->adjLists[src]; graph->adjLists[src] = newNode;

newNode = createNode(src, weight); newNode->next = graph->adjLists[dest]; graph->adjLists[dest] = newNode;

}

// Function to update traffic data

void updateTrafficData(Graph\* graph) { srand(time(0));

for (int i = 0; i < graph->numVertices; i++) { Node\* adjList = graph->adjLists[i];

while (adjList) {

adjList->weight = rand() % 20 + 1; // Random weight between 1 and 20

adjList = adjList->next;

}

}

}

typedef struct MinHeapNode { int v;

int dist;

} MinHeapNode;

typedef struct MinHeap { int size;

int capacity; int\* pos;

MinHeapNode\*\* array;

} MinHeap;

MinHeapNode\* createMinHeapNode(int v, int dist) {

MinHeapNode\* minHeapNode = (MinHeapNode\*)malloc(sizeof(MinHeapNode)); minHeapNode->v = v;

minHeapNode->dist = dist; return minHeapNode;

}

MinHeap\* createMinHeap(int capacity) {

MinHeap\* minHeap = (MinHeap\*)malloc(sizeof(MinHeap)); minHeap->pos = (int\*)malloc(capacity \* sizeof(int)); minHeap->size = 0;

minHeap->capacity = capacity;

minHeap->array = (MinHeapNode\*)malloc(capacity \* sizeof(MinHeapNode));

return minHeap;

}

void swapMinHeapNode(MinHeapNode\*\* a, MinHeapNode\*\* b) { MinHeapNode\* t = \*a;

\*a = \*b;

\*b = t;

}

void minHeapify(MinHeap\* minHeap, int idx) { int smallest, left, right;

smallest = idx; left = 2 \* idx + 1; right = 2 \* idx + 2;

if (left < minHeap->size && minHeap->array[left]->dist < minHeap-

>array[smallest]->dist) smallest = left;

if (right < minHeap->size && minHeap->array[right]->dist < minHeap-

>array[smallest]->dist) smallest = right;

if (smallest != idx) {

MinHeapNode\* smallestNode = minHeap->array[smallest]; MinHeapNode\* idxNode = minHeap->array[idx];

minHeap->pos[smallestNode->v] = idx; minHeap->pos[idxNode->v] = smallest;

swapMinHeapNode(&minHeap->array[smallest], &minHeap->array[idx]);

minHeapify(minHeap, smallest);

}

}

bool isEmpty(MinHeap\* minHeap) { return minHeap->size == 0;

}

MinHeapNode\* extractMin(MinHeap\* minHeap) { if (isEmpty(minHeap))

return NULL;

MinHeapNode\* root = minHeap->array[0];

MinHeapNode\* lastNode = minHeap->array[minHeap->size - 1]; minHeap->array[0] = lastNode;

minHeap->pos[root->v] = minHeap->size - 1; minHeap->pos[lastNode->v] = 0;

--minHeap->size; minHeapify(minHeap, 0);

return root;

}

void decreaseKey(MinHeap\* minHeap, int v, int dist) { int i = minHeap->pos[v];

minHeap->array[i]->dist = dist;

while (i && minHeap->array[i]->dist < minHeap->array[(i - 1) / 2]->dist) { minHeap->pos[minHeap->array[i]->v] = (i - 1) / 2;

minHeap->pos[minHeap->array[(i - 1) / 2]->v] = i; swapMinHeapNode(&minHeap->array[i], &minHeap->array[(i - 1) / 2]);

i = (i - 1) / 2;

}

}

bool isInMinHeap(MinHeap\* minHeap, int v) { if (minHeap->pos[v] < minHeap->size)

return true; return false;

}

void printArr(int dist[], int n) { printf("Vertex Distance from Source\n"); for (int i = 0; i < n; ++i)

printf("%d \t\t %d\n", i, dist[i]);

}

void dijkstra(Graph\* graph, int src) { int V = graph->numVertices;

int dist[V];

MinHeap\* minHeap = createMinHeap(V);

for (int v = 0; v < V; ++v) {

dist[v] = INF;

minHeap->array[v] = createMinHeapNode(v, dist[v]); minHeap->pos[v] = v;

}

minHeap->array[src] = createMinHeapNode(src, dist[src]); minHeap->pos[src] = src;

dist[src] = 0;

decreaseKey(minHeap, src, dist[src]);

minHeap->size = V;

while (!isEmpty(minHeap)) {

MinHeapNode\* minHeapNode = extractMin(minHeap); int u = minHeapNode->v;

Node\* pCrawl = graph->adjLists[u]; while (pCrawl != NULL) {

int v = pCrawl->vertex;

if (isInMinHeap(minHeap, v) && dist[u] != INF && pCrawl->weight + dist[u]

< dist[v]) {

dist[v] = dist[u] + pCrawl->weight; decreaseKey(minHeap, v, dist[v]);

}

pCrawl = pCrawl->next;

}

}

printArr(dist, V);

}

void analyzeTraffic(Graph\* graph) { printf("Analyzing traffic data...\n");

// Basic analysis: Find the shortest paths from a fixed source (e.g., vertex 0) dijkstra(graph, 0);

}

Module 5

void displayMenu() {

printf("Real-Time Traffic Analysis System\n"); printf("1. Update Traffic Data\n");

printf("2. Analyze Traffic\n"); printf("3. Exit\n");

}

int main() {

int vertices = 5; // Example with 5 vertices Graph\* graph = createGraph(vertices);

addEdge(graph, 0, 1, 4);

addEdge(graph, 0, 2, 1);

addEdge(graph, 1, 2, 2);

addEdge(graph, 1, 3, 5);

addEdge(graph, 2, 3, 8);

addEdge(graph, 3, 4, 3);

addEdge(graph, 4, 0, 7);

int choice; while (1) {

displayMenu();

printf("Enter your choice: "); scanf("%d", &choice);

switch (choice) { case 1:

updateTrafficData(graph); printf("Traffic data updated.\n"); break;

case 2:

analyzeTraffic(graph); break;

case 3:

exit(0); default:

printf("Invalid choice. Please try again.\n");

}

}

return 0;

}

## APPENDIX B - SCREENSHOTS RESULT AND DISCUSSION

ENTERING THE DETAILS OF THE VEHICLE

