Data Structures and Algorithms

Smart Traffic Management System

Course Project Report

School of Computer Science and Engineering

2023-24

Contents

1. Course and Team Details

1.1 Course details

Course Name	Data Structures and Algorithms
Course Code	23ECSC205
Semester	III
Division	B
Year	2023-24
Instructor	DR. PRIYANKA GAWADE

1.2 Team Details

Si. No.	Roll No.	Name
1.	62	SONALI JADHAV
2.	67	STUTI HUNACHAGI
3.	42	SARVESH NIRMALKAR
4.	03	AKHILESH JOSHI

1.3 Report Owner

Roll No.	Name
42	Sarvesh Nirmalkar

2. Introduction

Smart Traffic Management, this project exemplifies a sophisticated integration of cuttingedge algorithms. The utilization of Dijkstra's algorithm for meticulous route planning and Breadth-First Search (BFS) to establish city connectivity underscores our commitment to optimizing urban mobility. Beyond efficient navigation, the inclusion of parking slot reservation functionality further enhances the comprehensive nature of our solution. This project not only addresses the complexities of traffic management but sets a benchmark for the seamless integration of advanced algorithms in urban planning and mobility enhancement

3. Problem Statement

3.1 Domain

The urban landscape grapples with persistent inefficiencies in traffic management, marked by congestion and suboptimal routes. The pressing need for an adaptive solution is evident as conventional systems struggle to meet the demands of modern cities. Our project addresses this challenge, utilizing advanced algorithms to create a responsive traffic management system, aiming to alleviate congestion and enhance urban mobility

3.2 Module Description

Collecting and organizing city list :
As part of my role in the project, I have to compile a comprehensive list of cities close to Belagavi . I have to do thorough research to make sure that the cities I have selected are in line with the project goals. After the list is finalised, my next step is to organize it in a file format. This file structure is essential for easy access and management of the city data in the project.

Login Page:
The primary purpose of incorporating login page in the project is to verify the identity of a user. Users need to provide valid credentials (such as a username and password) to access the system. A login page enhances security by ensuring that only authorized users can access sensitive information, perform specific actions, or use certain features. Unauthorized access is restricted by requiring valid login credentials

4. Functionality Selection

Si. No.	Functionality Name	Known	Unknown	Principles applicable	Algorithms	Data Structures
	Name the functionality within the module	What information do you already know about the module? What kind of data you already have? How much of process	What are the pain points? What information needs to be explored and understood? What are challenges?	What are the supporting principles and design techniques?	List all the algorithms you will use	What are the supporting data structures?

		information is known?				
1	List all the major cities close to belagavi	The list of cities is readily available on google maps .google maps can be used to select particular cities as per our requirement	As there are too many cities near belagavi, there is a need to select only some parts and use them in project	Bruteforce List of cities is directly stored in the text file and retrived in the code when needed.	File functions	Text files
2	Login Page	The credentials of users are already created and stored in an structure	Complexity vs. Security: Balancing security requirements with a seamless user experience can be challenging.	Bruteforce The credentials created for user can be used to login in the code when needed	Linear search, Rabin karp, Brute force string matching	Array , Structure

5. Functionality Analysis

1.List all the major cities close to Belagavi:

Workflow :

The project provides shortest path from Belgavi to Cities close to it. it is important to prioritize the cities and select optimum number of cities and list them . 10 cities have been listed, and city code is give to each city . City code can be used as a node to represent the city in the graph. All the data is stored In the file named 'cities_list' for the reterival in the program.

Efficiency Analysis:-

Time complexity :- O(N) When retrieving data from a file and organizing it into a structure. Firstly, reading the file involves scanning through every part of it, and the time it takes (time complexity) is directly proportional to the size of the file .This step is often represented as O(N), where N is the number of elements or characters in the file. Secondly, parsing the file content means understanding its structure and converting it into a format suitable for the structure. The time complexity of this step is typically linear, O(N), depending on the complexity of the file format.

2.Login page:

The provided login page works as follows:

User Input:

The user is prompted to enter their username and password.

Authentication:
The entered username and password are sent to the authenticateUser function for verification.
The authenticateUser function compares the entered credentials with a predefined set of user credentials stored in the users array.

Authentication Logic:
The function iterates through the array of User structures to find a match.
If a match is found, the function returns 1, indicating successful authentication.
Otherwise, it returns 0 for authentication failure.

Time Complexity Analysis:

The time complexity of the authenticateUser function is determined by the number of iterations required to find a matching username-password pair or reach the end of the users array.

Best Case:

The best-case scenario occurs when the target user is the first one in the array, resulting in a time complexity of O(1).

Worst Case:

The worst-case scenario is when the target user is the last one in the array or is not present in the array at all.
In this case, the function needs to iterate through the entire array of numUsers elements.
Therefore, the worst-case time complexity is O(n), where n is the number of users.

Average Case:

In the average case, assuming a uniform distribution of target users, the function may need to check about half of the array elements.
The average-case time complexity is O(n).

6. Conclusion

Debugging and Performance Tuning:
Skills in debugging and performance tuning were honed while overcoming challenges in optimizing code for efficiency

Collaboration:
Smart task delegation enabled each team member to contribute their expertise effectively.

Practical Application:
Demonstrated the practical application of knowledge, emphasizing the fundamental role of data structures and algorithms in building reliable and scalable software.

Communication:
Clear and regular communication ensured alignment on project goals and timelines.

7.Refrences

1.Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, Fourth Edition, The MIT Press, 2022.
2. Anany V. Levitin, Introduction to the Design and Analysis of Algorithms. Addison-Wesley Longman Publishing Co, 2012.

~*~*~*~*~*~*~*~