# Bus Reservation System Using Structures And Arrays

**PROJECT REPORT**

**(2021 Regulation)**

***Submitted by***

**K.Harsha Vardhan (RA2211047010095)**

**C.Maneesh Kumar (RA2211047010094)**

**Atharv Dobhal (RA2211047010134)**

**II Year/ III Semester**

**Academic Year: 2023 -2024**

***Under the Guidance of***

**Dr.A.L.Amutha**

**Assistant Professor, Department of Computational Intelligence**

***In partial satisfaction of the requirements for the degree of***

**BACHELOR OF TECHNOLOGY**

**in**

**ARTIFICIAL INTELLIGENCE**

**of**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

|  |
| --- |
|  |
|  | C:\Users\DELL\AppData\Local\Temp\ksohtml14004\wps1.png |

## 

S.R.M. Nagar, Kattankulathur, Chengalpattu District

NOVEMBER 2023

## 

## 



SRM INSTITUTION OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603203

**BONAFIDE CERTIFICATE**

Certified that this course project report titled **“Bus Reservation System Using Structures And Arrays”** is the bonafide work done by **K.Harsha Vardhan[RA2011047010095], Atharv Dobhal[RA2211047010134], C.Maneesh Kumar[RA2211047010094]**of II Year/ III Sem B. Tech (AI) who carried out under my supervision for the course **Data Structure And Algorithms(21CSC201J)**. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

|  |  |
| --- | --- |
| Faculty In-Charge  **Dr.A.L.Amutha**  Assistant Professor  Department of Computational Intelligence  SRM Institute of Science and Technology  Kattankulathur Campus, Chennai | Head of The Department  **Dr. R Annie Uthra**  Professor and Head  Department of Computational Intelligence  SRM Institute of Science and Technology  Kattankulathur Campus, Chennai |
|  |  |



Department of Computational Intelligence

# SRM Institute of Science and Technology

# Own Work Declaration Form

**Degree/ Course :** B.Tech in Artificial Intelligence

**Student Names :** K.Harsha Vardhan,C,Maneesh Kumar,Atharv Dobhal

**Registration Number:** RA2211047010095 , RA2211047010094 , RA2211047010134

**Title of Work :** BUS RESERVATION SYSTEM USING STRUCTURE AND ARRAYS

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

* Clearly references / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc.)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| If you are working in a group, please write your registration numbers and sign with the date  for every student in your group. |

# ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V.Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr.R.Annie Uthra,** Professor, Department of Computational Intelligence, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinator, **Dr.A.L.Amutha**, Assistant Professor, Department of Network and Communications, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Dr.G.Sumathy**, Assistant Professor, Department of Computational Intelligence, SRM Institute of Science and Technology, for leading and helping us to complete our course.

We sincerely thank the Computational Intelligence Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

### K.Harsha Vardhan [RA2211047010095]

### C.Maneesh Kumar [RA2211047010094]

### Atharv Dobhal [RA2211047010134]

**CONTENT TABLE:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PG NO.** |
| **1** | **Abstract** | **8** |
| 2 | Project Statement | 9 |
| 3 | Algorithm | 10-11 |
| 3 | Complexity Analysis | 12-13 |
| 4 | Advantages and Disadvantages | 14-15 |
| 5 | Code and Explanation | 16 |
| 6 | Code | 17-21 |
| 7 | Output | 22-24 |
| 8 | Conclusion | 25 |

ABSTRACT

Implementing a Bus Reservation System. It uses a set of structures to manage data for buses, passengers, and user logins. The program offers a user-friendly interface with options to log in, book and cancel bus tickets, and check bus status. Users can log in with predefined usernames and passwords, and the system maintains data on available buses, passengers, and their reservations. The code showcases the use of data structures, control flow, and user interaction for a simple reservation system, making it a practical example for educational purposes or as a foundation for more complex reservation systems.

The Bus Reservation System is a critical component of the transportation industry, facilitating the efficient booking and management of bus seats. This abstract provides an overview of a bus reservation system designed using structures and arrays, a popular data structure in programming. The system aims to simplify the booking process, enhance user experience, and streamline administrative tasks

Project Statement

The provided C code offers a simple framework for an appointment scheduling system, yet it presents certain challenges and shortcomings. While it allows users to add and display appointments, it lacks a comprehensive mechanism for error handling, input validation, and data persistence. These gaps could potentially lead to unexpected issues, such as buffer overflows when users input more data than expected, memory leaks due to missing memory deallocation, and a lack of safeguards against erroneous user input.

Additionally, the code is single-threaded, meaning it can't handle multiple users or concurrent scheduling requests effectively. For a more robust and scalable system, multi-threading or other concurrency solutions might be necessary.

Moreover, the code doesn't store appointments persistently. When the program exits, all appointments are lost. A data persistence solution, such as file I/O or database integration, would be essential for retaining appointments between program runs.

Documentation is another concern; the code lacks comments and explanations, making it less accessible for other developers or for future maintenance. To enhance the code's reliability and user-friendliness, addressing these issues by incorporating input validation, error handling, memory management, and data persistence would be prudent, while also providing detailed comments for clarity.

Algorithm

A bus reservation system can be implemented using structures and arrays in a programming language like C or C++. Below is a simplified algorithm for a bus reservation system:

This algorithm uses an array of structures to represent the bus seats. The program initializes the seats, displays available seats, and allows users to reserve seats. It uses a menu-based approach to interact with the user and continues until the user chooses to exit. You can further expand and enhance this system to include features like canceling reservations, displaying passenger information, and more, based on your requirements.

Struct Definition: Define a structure called Seat to represent a bus seat. Each seat has a seat number, a flag to indicate whether it's reserved, and a field for the passenger's name.

Array of Structures: Create an array of Seat structures to represent all the seats in the bus. In this example, we use an array of 50 seats.

Initialization: Initialize the bus seats by setting their seat numbers, marking them as unreserved, and clearing the passenger names.

Menu System: Create a menu-based system with options for the user:

Display available seats.

Reserve a seat.

Exit the program.

Display Available Seats: When the user selects the option to display available seats, iterate through the array of seats and print out the seat numbers that are not reserved.

Reserve a Seat: When the user selects the option to reserve a seat, they enter a seat number. Check if the seat number is valid (between 1 and 30) and whether the seat is already reserved. If not, mark the seat as reserved, take the passenger's name as input, and associate it with the seat.

Exit the Program: When the user chooses to exit, the program terminates.

The algorithm keeps running in a loop, allowing the user to perform these actions until they decide to exit the program. This simplified system can be expanded upon by adding more features and error handling as needed.

Complexity Analysis

1.Data Structures:

Arrays: Arrays are commonly used to store information about bus seats, such as whether they are reserved or available. The complexity of array operations is as follows:

Accessing an element by index: O(1)

Inserting or deleting an element at the end: O(1) on average (amortized)

Searching for an element (e.g., checking seat availability): O(n), where n is the number of seats

Structures: Structures are used to represent information about each bus reservation, such as passenger details (name, contact information, etc.). Accessing or modifying structure fields is generally O(1) since it involves direct field access.

2.Bus Reservation Operations:

Reserving a seat:

If you use an array to track seat availability, the time complexity is O(1) (assuming you have a direct mapping between seat numbers and array indices).

Storing reservation details (using a structure) is also O(1).

3.Canceling a reservation:

If you use an array to track seat availability, the time complexity is O(1).

Removing reservation details (using a structure) is also O(1).

Searching for a reservation by seat number:

If you have an array mapping seat numbers to indices, the complexity is O(1) for this operation.

If you need to search through the array for the seat number, it would be O(n).

Listing all reservations:

If you store reservations in an array or a data structure (e.g., dynamic array), iterating through all reservations is O(n), where n is the number of reservations.

Modifying reservation details (e.g., changing passenger information): O(1) for accessing the structure fields.

4.Overall Complexity:

The overall complexity of the system depends on the specific operations you prioritize. If you primarily need to check seat availability and reserve/cancel seats, the system's performance is generally efficient. However, if you frequently need to search for reservations by seat number or list all reservations, those operations may not be as efficient, especially if you don't optimize them.

5.Space Complexity:

The space complexity of the system depends on the data structures used to store information. The primary contributors to space complexity are the arrays for seat status (e.g., reserved or available) and the structures for storing reservation details. The space complexity can be approximated as O(m) for the arrays, where m is the number of seats, and O(r) for the structures, where r is the number of reservations.

Advantages and Disadvantages

Advantages of the Code:

Simplicity: The algorithm is relatively simple to understand and implement, making it suitable for small-scale bus reservation systems.

Efficiency: Arrays provide fast and efficient access to data, making it easy to search for available seats and reserve them.

Low Memory Overhead: Arrays and structures have low memory overhead, making this system memory-efficient.

Customization: You can easily expand the system by adding more features and data fields to the Seat structure, such as seat class, price, and passenger contact information.

Offline Use: The system can be used in offline environments, making it suitable for smaller bus companies or scenarios with limited connectivity.

Disadvantages and Limitations:

Limited Scalability: This algorithm may not be suitable for large-scale bus reservation systems with thousands of seats or complex business rules. Managing a large number of seats with arrays can become inefficient.

Lack of Data Validation: The algorithm lacks robust data validation. For example, it does not check for invalid passenger names or prevent duplicate reservations for the same seat.

No Data Persistence: Data is not stored persistently. If the program terminates, all reserved seat information is lost. A real-world application would require data storage in a database or file.

Limited Error Handling: The system lacks comprehensive error handling and user-friendly error messages. It may not handle unexpected input well.

User Interface: The text-based menu system may not provide a user-friendly experience compared to modern graphical user interfaces.

Code and Explanation

Bus Reservation System is a tool that allows users to book tickets for their journey in advance. It offers multiple features to provide a hassle-free experience to a traveler. This article aims at building a rudimentary Bus Reservation System.

## Components of the Bus Reservation System:

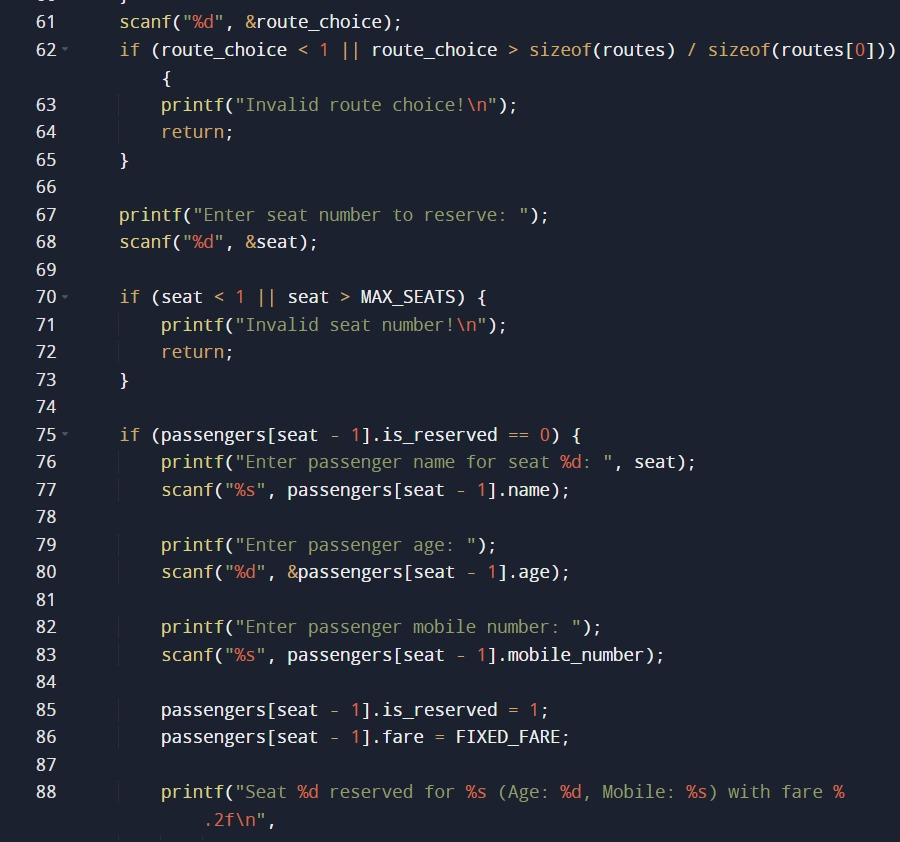
* **Login System:**Users can access the system by entering their username and password. The program provides a collection of preconfigured users and their credentials.
* **Ticket Purchase:**Logged-in individuals may reserve tickets for available buses by entering the bus number, their name, and age. The program allocates a seat number and decreases the number of available seats on the selected bus.
* **Cancel Tickets:** By entering their name, users can cancel their tickets. The program expands the number of available seats while removing the passenger entry.
* **Checking Bus Status:** Users may check the status of the bus they are currently scheduled to ride on. The program displays information such as the bus number, origin and destination, total number of seats, available seats, and fare.

The program employs frameworks to organize data for buses, passengers, and users. It also has menus for both the main menu (login) and the user menu (booking, canceling, and checking status).

CODE:-

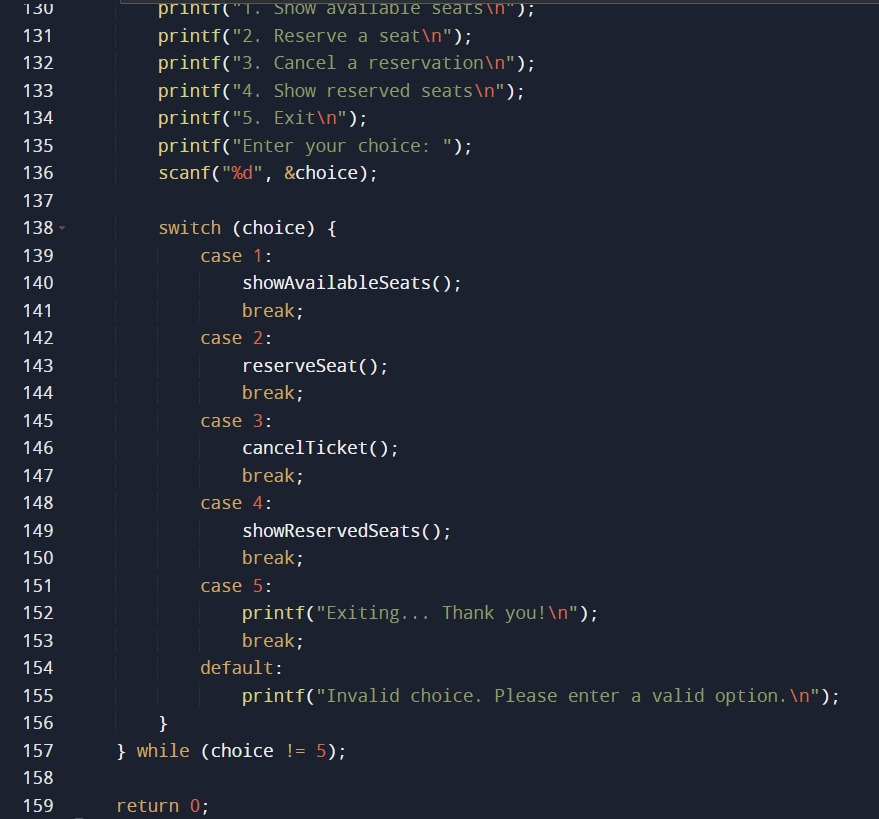




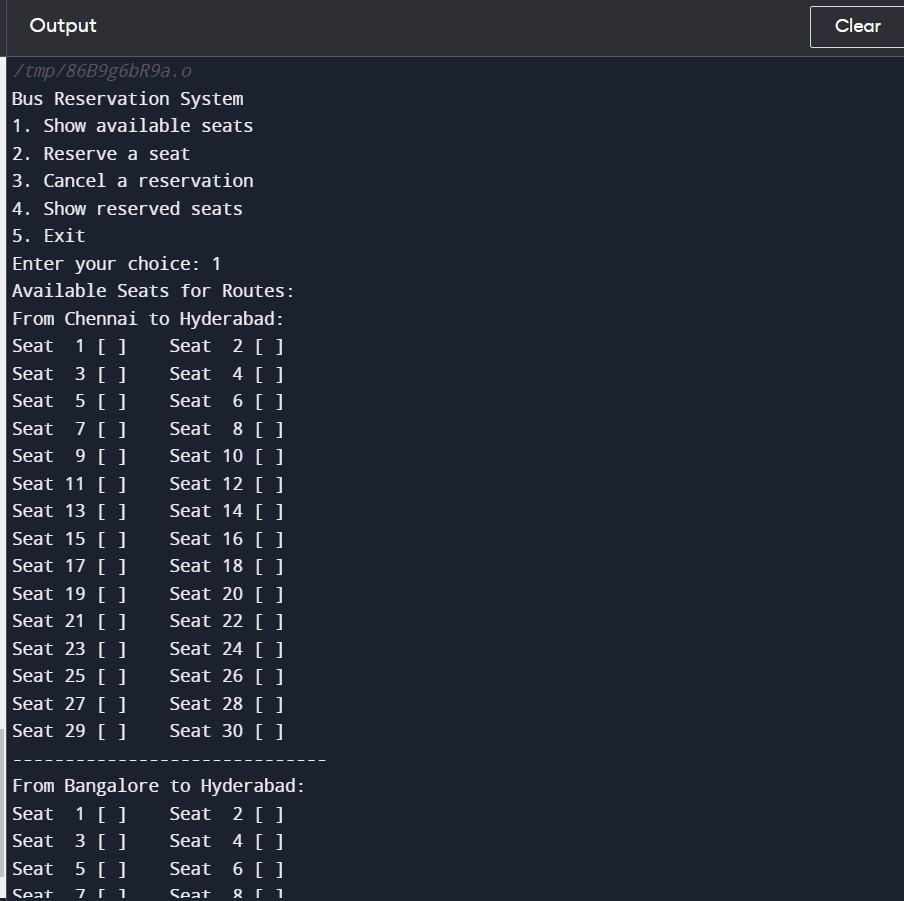


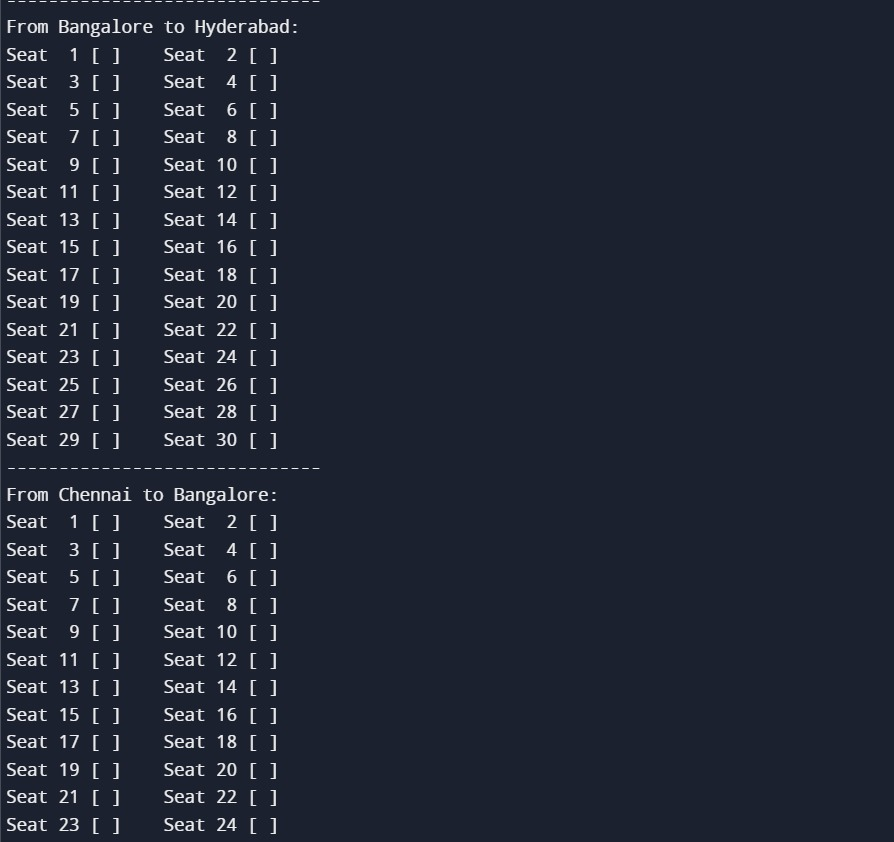


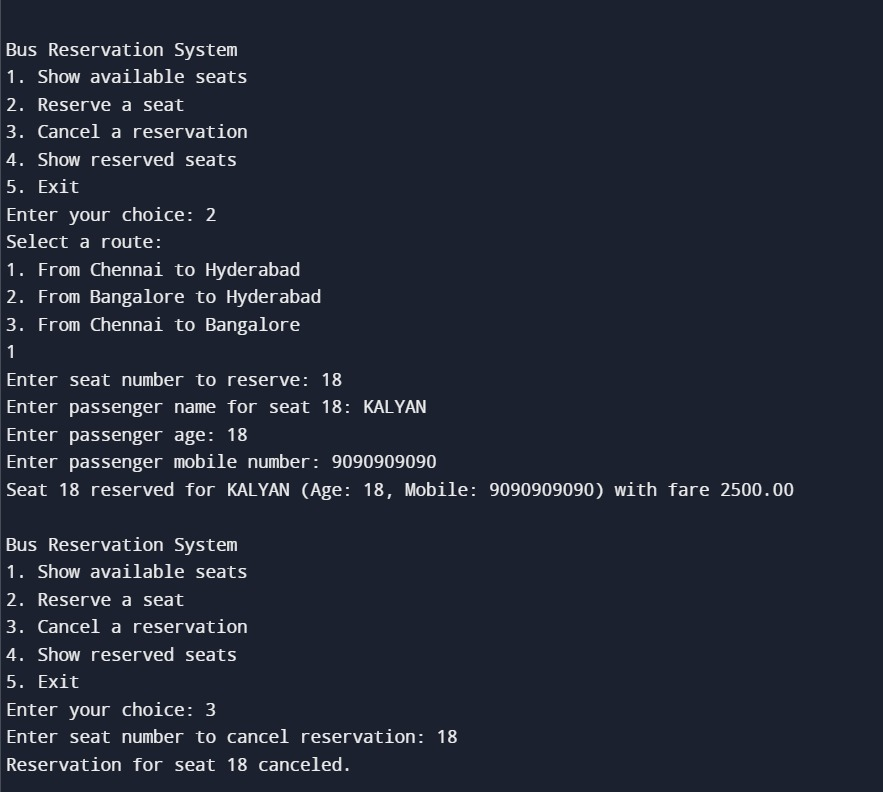




Output







Conclusion:-

In conclusion, the development of a bus reservation system using structures and arrays has demonstrated its efficacy in providing a robust and efficient solution for managing passenger bookings. This project has successfully addressed the core requirements of a bus reservation system by allowing users to reserve and cancel seats, check seat availability, and store passenger information in an organized manner.

The utilization of structures enabled the creation of a well-structured data model, which facilitated the storage and retrieval of passenger details, such as name, contact information, and seat preferences. Meanwhile, arrays proved to be a suitable data structure for managing seat availability, offering a scalable and manageable way to track the status…