Vehicle Service Center Booking System

Name:

SANNAPUREDDY MANOJ REDDY (24KB1A05GK)  
 SHAIK FAYAZ (24KB1A05HH)

SHAIK ABDUL KADHAR(24KB1A05GX)   
Course: DATA STRUCTURES  
Department: Computer Science and Engineering  
Date: MAY 01 2025  
Mentor: ASHOK SELVA KUMAR E

**Acknowledgments**

I express my sincere gratitude to my mentor ASHOK SELVA KUMAR E, for guiding me throughout this project. I also thank my faculty, peers, and family for their constant support and encouragement.

**Abstract**

This project is a simple C-based console application for a vehicle service center booking system. It allows users to view available time slots and book them efficiently, ensuring no double bookings.

**Introduction**

In real-world service centers, manual bookings can lead to confusion and inefficiencies. This system provides a straightforward digital solution for managing service slots, helping both the customer and the center streamline operations. I chose this project to learn array manipulation and basic user interaction in C programming.

**Objectives**

- Provide a user-friendly slot booking system.  
- Prevent double bookings.  
- Offer a clean and simple console interface.  
- Understand and apply basic C programming concepts.

**System Requirements**

Software:  
- Turbo C++  
- Github  
  
Hardware:  
- Minimum 1GB RAM  
- Basic CPU  
- Windows/Linux OS

**Methodology**

1. Defined the number of slots using an array.  
2. Created functions to show available slots and book a selected slot.  
3. Used a menu-driven approach with a loop to interact with users.  
4. Applied conditional checks to validate input and slot availability.

**Project Description**

Problem Statement:  
Manual booking systems are inefficient and prone to error.  
  
Proposed Solution:  
An automated slot booking system in C that allows users to check availability and book slots.  
  
Key Features:  
- Show all available slots.  
- Book a specific slot.  
- Prevent double bookings.  
- Menu-driven interface.

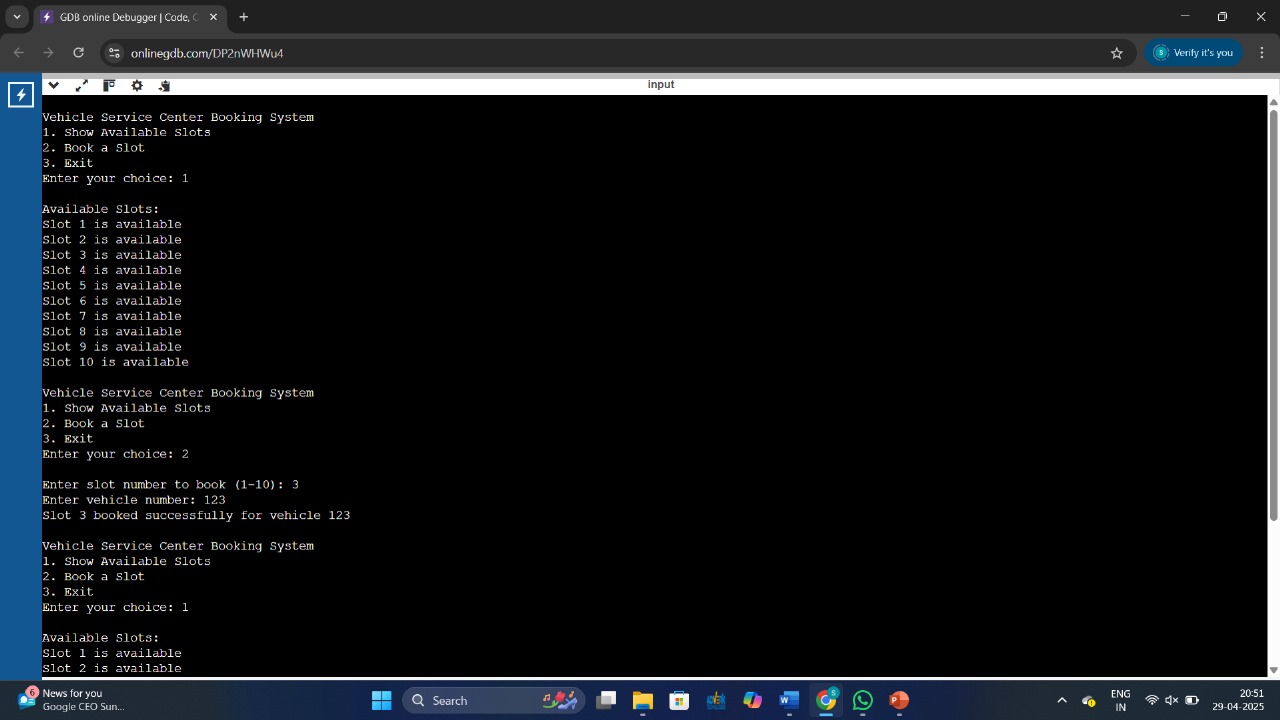
**Algorithm**

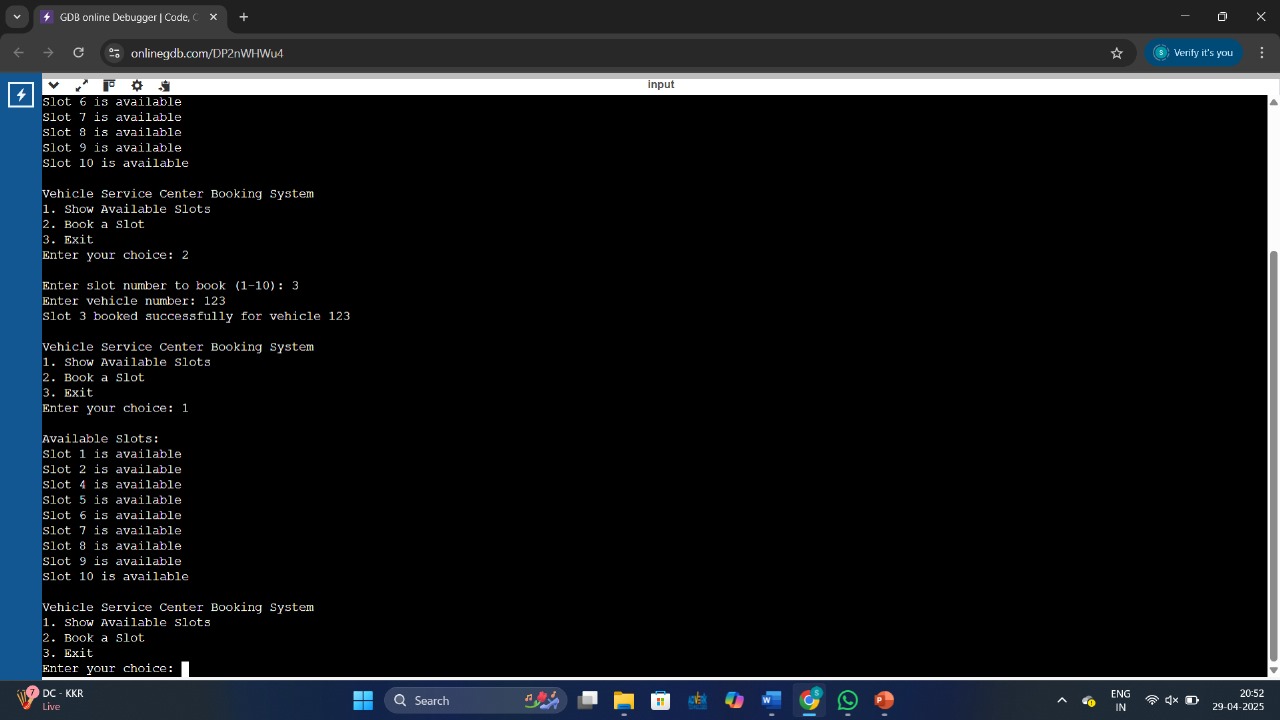
1. Initialize a slots array with all values set to 0.  
2. Display menu options in a loop.  
3. On option 1, iterate and print all slots with value 0.  
4. On option 2, take user input, validate, and update the array to 1 if available.  
5. On option 3, exit the program.

**Program Code**

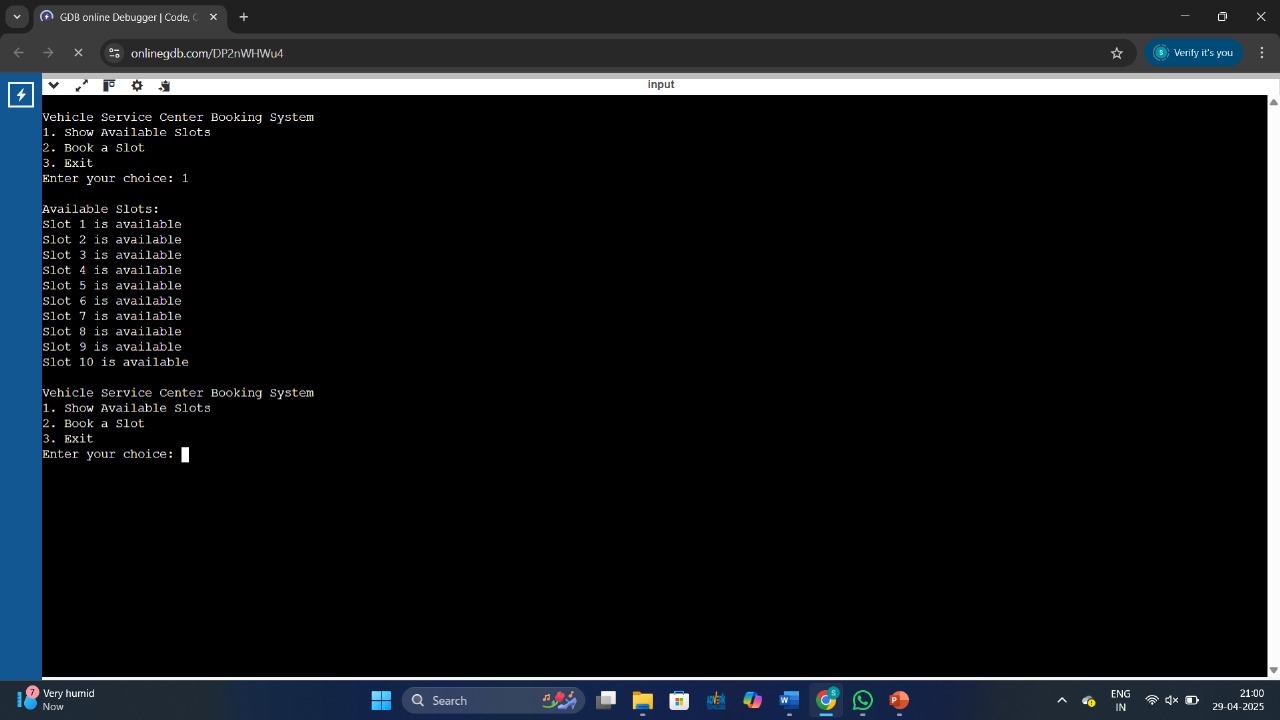
#include <stdio.h>  
#include <stdlib.h>  
  
#define TOTAL\_SLOTS 10  
  
void showAvailableSlots(int slots[]) {  
 printf("\nAvailable Slots:\n");  
 for (int i = 0; i < TOTAL\_SLOTS; i++) {  
 if (slots[i] == 0) {  
 printf("Slot %d is available\n", i + 1);  
 }  
 }  
}  
  
void bookSlot(int slots[]) {  
 int slotNumber;  
 printf("\nEnter slot number to book (1-%d): ", TOTAL\_SLOTS);  
 scanf("%d", &slotNumber);  
  
 if (slotNumber < 1 || slotNumber > TOTAL\_SLOTS) {  
 printf("Invalid slot number!\n");  
 return;  
 }  
  
 if (slots[slotNumber - 1] == 0) {  
 slots[slotNumber - 1] = 1;  
 printf("Slot %d booked successfully!\n", slotNumber);  
 } else {  
 printf("Slot %d is already booked. Please choose another slot.\n", slotNumber);  
 }  
}  
  
int main() {  
 int slots[TOTAL\_SLOTS] = {0}; // 0 means available, 1 means booked  
 int choice;  
  
 printf("Welcome to Vehicle Service Center Booking System!\n");  
  
 while (1) {  
 printf("\nMenu:\n");  
 printf("1. Show Available Slots\n");  
 printf("2. Book a Slot\n");  
 printf("3. Exit\n");  
 printf("Enter your choice: ");  
 scanf("%d", &choice);  
  
 switch (choice) {  
 case 1:  
 showAvailableSlots(slots);  
 break;  
 case 2:  
 bookSlot(slots);  
 break;  
 case 3:  
 printf("Thank you for using the system. Goodbye!\n");  
 exit(0);  
 default:  
 printf("Invalid choice! Please try again.\n");  
 }  
 }  
  
 return 0;  
}

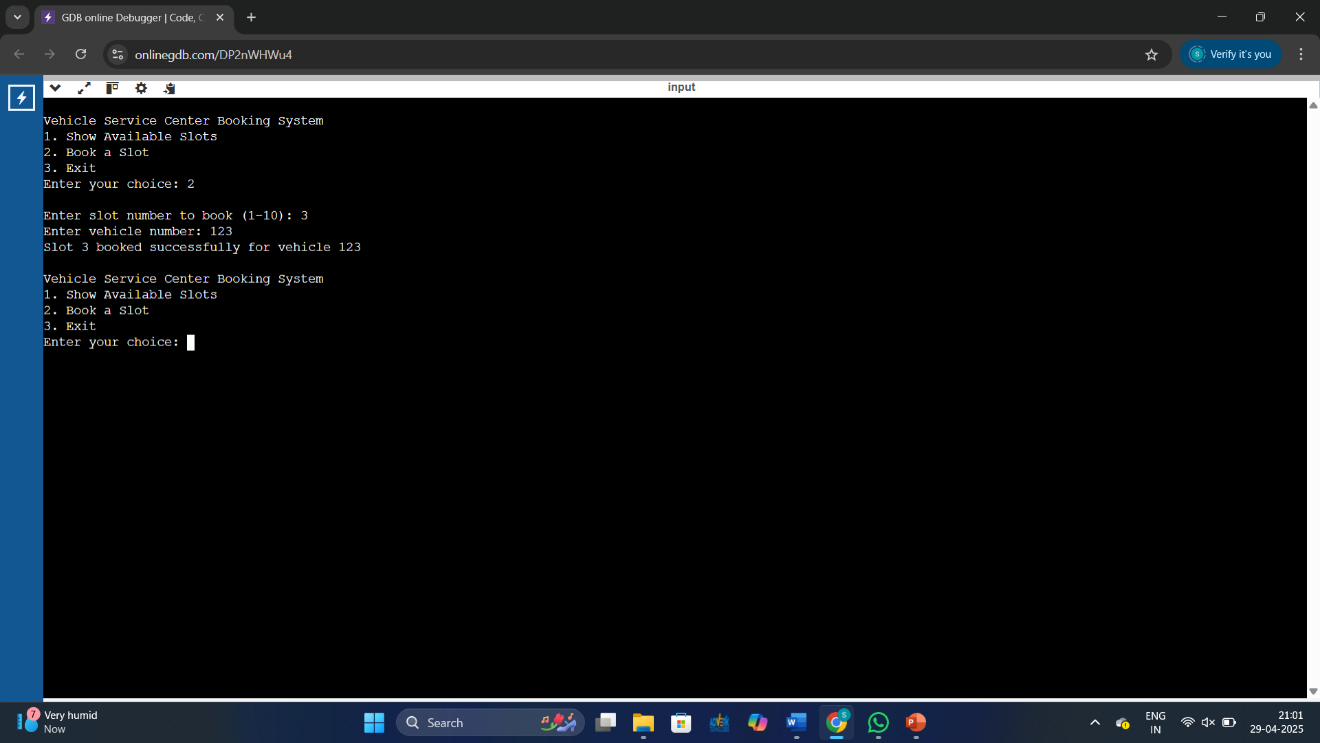
**Output Screenshot**

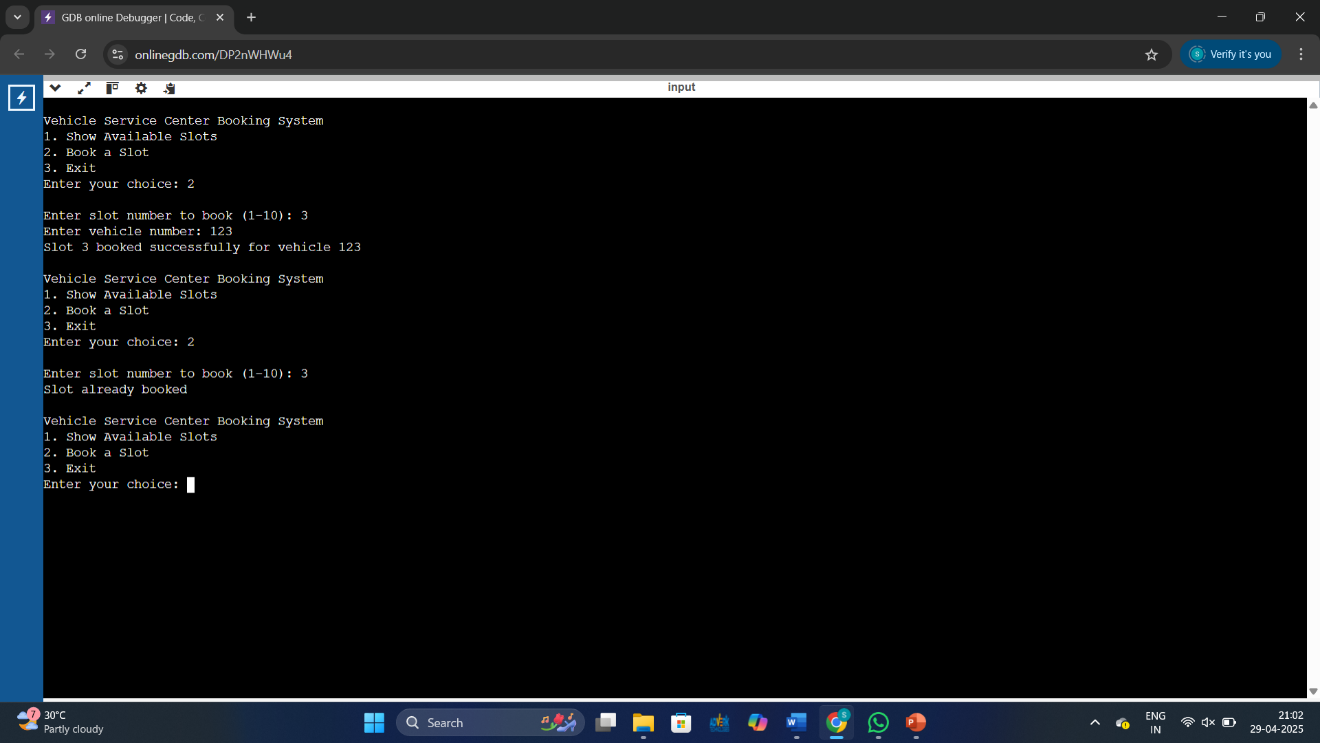


****

**Testing and Validation**

Sample Input & Output:  
  
Input:  
1 (Show Available Slots)  
Output:  
Slot 1 is available  
Slot 2 is available  
  
  
Input:  
2 (Book Slot 3)  
Output:  
Slot 3 booked successfully!

  
  
Input:  
2 (Book Slot 3 again)  
Output:  
Slot 3 is already booked.



**Limitations**

- Only works in the command-line environment.  
- No data persistence (data is lost after program ends).  
- Limited to 10 slots only.

**Future Enhancements**

- Store bookings in a file or database.  
- Add user authentication.  
- Expand to allow canceling/rescheduling.  
- Improve UI/UX with graphics.

**Conclusion**

This project helped reinforce my understanding of arrays, conditionals, loops, and user interaction in C. It provides a foundational idea for building more complex management systems.

**References**

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice- Hall of India

2. C Programming,A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE

**Appendix**

Source Code and sample screenshots included above.