

GARAGE MANAGEMENT SYSTEM LAUNCH

| | |
|-------------------|-----------------------------------|
| NAME | : AMODA RASHMIKA MENDIS |
| UGC NUMBER | : UGC0323017 |
| UNIVERSITY | : LNBTI |
| MODULE | : PROGRAMMING FUNDAMENTALS |

TABLE OF CONTENTS

INTRODUCTION

PROBLEM STATEMENT

METHODOLOGY

IMPLEMENTATION

TESTING

RESULT OF

DISCUSSION

FUTURE WORK

REFERENCES

APPENDICES

INTRODUCTION

This project is about developing a computer program that enhances the efficiency of garage operations. It aims to streamline vehicle check-ins, maintenance tracking, inventory management, and customer-mechanic communication. The primary goal is to improve how garages operate, making everything more organized and hassle-free for both staff and customers.

OBJECTIVE AND GOALS

The main objectives and goals of this project are:

- To simplify the process of managing vehicle check-ins and service schedules.
- To maintain an organized and easily accessible inventory of parts and tools.
- To improve communication between mechanics and customers, making it more efficient and effective.
- To ensure a better experience at garages for customers and staff by making the service delivery system more organized and efficient

01. PROBLEM STATEMENT

WHAT'S THE PROBLEM?

The project focuses on addressing the challenges garages face, such as difficulties in managing service schedules, keeping track of inventory, and maintaining clear communication between mechanics and customers. These issues can lead to longer wait times, customer dissatisfaction, and inefficiencies in garage operations.

WHY IT MATTERS?

This issue is crucial because it directly impacts the quality of service garages can provide. When garages don't operate smoothly, it can result in delays in vehicle repairs, mistakes in inventory management, and overall poor customer experience. Solving these problems means customers receive better service faster, garages operate more efficiently, and both customer satisfaction and business profitability can improve. Ensuring garages run efficiently is key to providing quality automotive services.

02. METHODOLOGY

In developing my project, I adopted a step-by-step approach to create a program that enhances garage operations. Initially, I analyzed the main challenges garages encounter, such as scheduling services and managing inventory. Then, I designed a computer program to address these issues by allowing staff and customers to handle appointments and inventory more effectively. I incorporated measures to ensure data security and validate all information entered into the system.

TECHNOLOGIES AND TOOLS USED Programming Language:

C++ was chosen for its efficiency and ability to handle complex operations. Data Storage: Text files store all data (vehicles, customers, services, inventory) for simplicity and easy access. Security: Passwords are protected using a hashing function to enhance security, making the data files more secure. User Interface: A console-based interface offers clear menus and options, simplifying user interaction. Validation: Regular expressions validate input formats for dates, times, vehicle registration numbers, and contact numbers, ensuring accuracy.

03. IMPLEMENTATION PROCESS

I built the project incrementally. Starting with the basic structure, I then added features one by one, such as user registration, login functionality, service scheduling, inventory management, and billing. I tested each component as I developed it to ensure proper functionality.

KEY FEATURES User Accounts:

People can sign up as customers or staff members, and log in to access their specific services. Service Scheduling: Customers can book or cancel service appointments. Mechanics can view their schedules and mark services as completed. Inventory Management: Staff can track parts and tools, update quantities, and receive low stock alerts. Billing: Automatic generation of invoices based on services provided and parts used.

DATA VALIDATION EXAMPLES

```
4 using namespace std;
5
6 // Function prototypes for input validation(DATA VALIDATION)
7 bool isValidPhoneNumber(const string& phoneNumber);
8 bool isValidDate(const string& date);
9 int stringToInt(const string& str);
10 bool isAllDigits(const string& str);
11
```

```
do {  
    cout << "\t\t\t\t\t Enter your appointment Date (YYYY-MM-DD): ";  
    cin >> appointmentDate;  
} while (!isValidDate(appointmentDate));
```

```
// Function to validate date
bool isValidDate(const string& date) {
    if (date.length() != 10 || date[4] != '-' || date[7] != '-' ||
        !isdigit(date[0]) || !isdigit(date[1]) || !isdigit(date[2]) || !isdigit(date[3]) ||
        !isdigit(date[5]) || !isdigit(date[6]) || !isdigit(date[8]) || !isdigit(date[9])) {
        cout << "Invalid date format. Please use YYYY-MM-DD." << endl;
        return false;
    }

    int year = stringToInt(date.substr(0, 4));
    int month = stringToInt(date.substr(5, 2));
    int day = stringToInt(date.substr(8, 2));

    if (year < 2000 || year > 2100 || month < 1 || month > 12 || day < 1 || day > 31) {
        cout << "Invalid date. Please enter a valid date." << endl;
        return false;
    }

    return true;
}
```

I HAVE USED DATE VALIDATION AND PHONE NUMBER VALIDATION
HERE AND UPPER ONE IS A EXAMPLE FOR DATE VALIDATION

4. TESTING TESTING OVERVIEW

I thoroughly tested my program to ensure its functionality. I performed various tests to identify potential issues. I created a comprehensive test plan and documented the results meticulously.

TEST PLAN AND TEST CASES I

planned to test various components of my program, including user registration and login, service scheduling and cancellation, inventory management, and billing. Each test case was designed as a specific task with step-by-step instructions

- SHOWING TEST CASES
- SHOWING RESULTS

Client side of the GMS

1. User Registration

| Test Case | Input | Expected Output |
|--------------------|--|--------------------------------|
| Valid Registration | Username: cl001 Phone: 1234567890 License ID: ABC123 Vehicle Number: XYZ789 Vehicle Type: Sedan Password: pass123 | Registration successful |
| Invalid Username | Username: admin (other fields filled correctly) | Error: Invalid username format |

2. User Login

| Test Case | Input | Expected Output |
|------------------------|--|---|
| Valid Client Login | Username: cl001 Password: pass123 | Login successful, Client Menu displayed |
| Invalid Login | Username: cl001 Password: wrongpass | Login error, return to main menu |
| Service Provider Login | Username: se001 Password: servpass | Login successful, Service Provider Menu displayed |

3. Book a Service

| Test Case | Input | Expected Output |
|------------------------|---|-----------------------------|
| Valid Booking | Username: cl001 Service Type: Full Service Date: 2024-09-15 | Service booked successfully |
| Booking with Past Date | Username: cl001 Service Type: Tune Up Date: 2023-01-01 | Error: Invalid date |

4. Make an Appointment

| Test Case | Input | Expected Output |
|-------------------------|---|------------------------------------|
| Valid Appointment | Name: John Doe Date: 2024-10-01 Details: Vehicle inspection Phone: 9876543210 Request Call: Yes | Appointment submitted successfully |
| Overlapping Appointment | Date: 2024-10-01 (already booked) (other fields filled) | Error: Date already booked |

5. Cancel Appointment

| Test Case | Input | Expected Output |
|--------------------------|--|-----------------------------------|
| Valid Cancellation | Name: John Doe Date: 2024-10-01 Details: Vehicle inspection Reason: Schedule conflict | Appointment canceled successfully |
| Non-existent Appointment | Name: Jane Smith Date: 2024-11-15 (other fields filled) | Error: Appointment not found |

6. View Service Status

| Test Case | Input | Expected Output |
|-------------------------|---------------------------|---|
| Client with Bookings | Username: cl001 | Display of all booked services and appointments for cl001 |
| Client without Bookings | Username: cl002 (new user | No bookings or appointments found |

7. Update Customer Details

| Test Case | Input | Expected Output |
|-------------------|--|------------------------------|
| Valid Update | Username: cl001 New Password: newpass123 New Phone: 5555555555 (other fields updated) | Details updated successfully |
| Non-existent User | Username: cl999 | User not found |

8. Service Provider Functions

| Test Case | Input | Expected Output |
|---------------------------|--|-------------------------------------|
| View Bookings | Select option in Service Provider Menu | Display of all client bookings |
| Search Vehicle | Enter Registration: ABC123 | Display vehicle and owner details |
| Update Service Records | Enter details for a completed service | Service record added successfully |
| Update Service Completion | Mark a service as completed | Service status updated successfully |
| | | |

These test cases cover the main functionalities of the Garage Management System.

Server side of GMS

1. View Bookings

| Test Case | Action | Expected Output |
|-------------------|--|--|
| Multiple Bookings | Select "View Bookings" option | Display of all booked services with client names, service types, dates, and statuses |
| No Bookings | Select "View Bookings" option when no bookings exist | Message: "No bookings found" |

2. View Appointments

| Test Case | Action | Expected Output |
|-----------------------|--|--|
| Multiple Appointments | Select "View Appointments" option | Display of all appointments with dates, names, details, phone numbers, and call request statuses |
| No Appointments | Select "View Appointments" option when no appointments exist | Message: "No appointments found" |

3. Search Vehicle By Registration

| Test Case | Input | Expected Output |
|----------------------|---------------|--|
| Valid Registration | Enter: ABC123 | Display of vehicle details including owner's information |
| Invalid Registration | Enter: XYZ999 | Message: "No vehicle found with the registration number: XYZ999" |

4. Search Bookings By Username

| Test Case | Input | Expected Output |
|---------------------------|--------------|--|
| Username with Bookings | Enter: cl001 | Display of all bookings for cl001 including service types, dates, and statuses |
| Username without Bookings | Enter: cl002 | Message: "No bookings found for the username: cl002" |

5. Update Service Records

| Test Case | Input | Expected Output |
|------------------|---|---|
| Valid Update | Username: cl001 Service: Oil Change Parts: Oil Filter Notes: Routine maintenance | Message: "Service records updated successfully!" |
| Invalid Username | Username: cl999 (other fields filled) | Service record added, but may not correspond to a real user |

6. Update Service Completion Status

| Test Case | Input | Expected Output |
|-------------------|--|---|
| Mark as Completed | Username: cl001 Status: yes | Message: "Service status updated successfully!" |
| Mark as Pending | Username: cl002 Status: no Expected Date: 2024-12-01 | Message: "Service status updated successfully!" |

7. Display Canceled Appointments

| Test Case | Input | Expected Output |
|------------------------|---|--|
| Multiple Cancellations | Select "Display Canceled Appointments" option | Display of all canceled appointments with names, dates, details, and reasons |
| No Cancellations | Select "Display Canceled Appointments" option when no cancellations exist | Message: "No canceled appointments found" |

5. RESULTS AND DISCUSSION

SUMMARY OF PROJECT OUTCOMES

My project now enables customers and garage staff to efficiently manage service appointments, inventory, and billing. It operates effectively, simplifying garage operations.

COMPARISON WITH INITIAL OBJECTIVES

I aimed to streamline garage visits by improving how service appointments and inventory are managed. The project meets these goals by offering easy ways for customers to book or cancel appointments and for staff to keep track of their schedules, vehicle details, and inventory levels.

CHALLENGES FACED AND LESSONS LEARNED

Challenges: Ensuring all components of the program work together seamlessly was challenging. Comprehensive testing of every feature was time-consuming.

Lessons Learned: I learned that meticulous planning and thorough testing are crucial. Breaking down complex tasks into smaller, manageable parts makes problem-solving easier. Incorporating user feedback was instrumental in improving the project significantly.

6. FUTURE WORK POSSIBLE EXTENSIONS OR ENHANCEMENTS

- Online Booking System: Implement a web-based calendar for easier service scheduling.
- Mobile App: Develop a mobile application for improved accessibility.
- Integration with Suppliers: Automate parts ordering when inventory is low.
- Customer Loyalty Program: Implement a points system for frequent customers.

AREAS FOR FUTURE RESEARCH OR DEVELOPMENT

- Predictive Maintenance: Utilize vehicle data to predict when services will be needed
- AI-powered Diagnostics: Implement an AI system to assist in vehicle problem diagnosis.
- IoT Integration: Connect with IoT devices in vehicles for real-time monitoring and diagnostics.
- Blockchain for Vehicle History: Implement a blockchain-based system for maintaining tamper-proof vehicle service histories.

7. REFERENCES

During the project, I utilized these resources to help build and improve the system:

- Cplusplus.com: For in-depth C++ programming knowledge.
- Stack Overflow: A valuable resource for solving various coding challenges.
- GitHub: Source of useful code snippets and libraries.
- Microsoft Docs: Assisted in understanding file operations in C++.

EXTERNAL LIBRARIES AND TOOLS

- regex: Used for validating vehicle registration numbers, contact numbers, and dates.
- fstream: Facilitated file operations such as reading and writing data.
- iostream: For input and output operations.
- vector: For dynamic array management.

These resources were instrumental in successfully completing my project.

8. APPENDICES

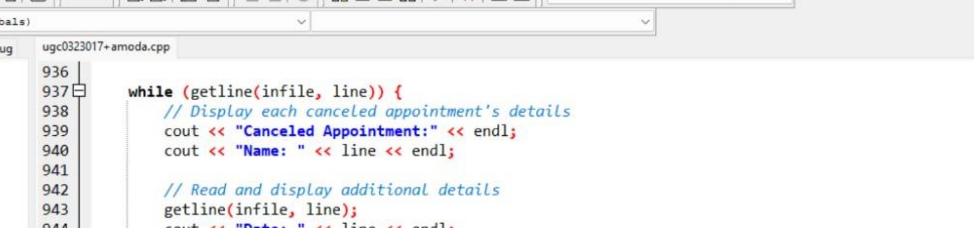

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
888 }
889 }
890 // Function to update service completion status
891 void updateServiceCompletionStatus() {
892     system("cls");
893     string username, serviceStatus, completionDate;
894
895     cout << "Enter the username for which you want to update service status: ";
896     cin.ignore();
897     getline(cin, username);
898
899     cout << "Is the service completed? (yes/no): ";
900     getline(cin, serviceStatus);
901
902     if (serviceStatus == "no") {
903         cout << "Enter the expected completion date (YYYY-MM-DD): ";
904         getline(cin, completionDate);
905     } else {
906         completionDate = "Completed";
907     }
908
909     ofstream outfile("service_completion.txt", ios::app);
910     if (!outfile) {
911         cout << "Error opening service completion.txt for writing." << endl;
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
857 // Function to update service records
858 void updateServiceRecords() {
859     system("cls");
860     string username, servicePerformed, partsReplaced, additionalNotes;
861
862     cout << "Enter the username for which you want to update service records: ";
863     cin.ignore();
864     getline(cin, username);
865
866     cout << "Enter the service performed: ";
867     getline(cin, servicePerformed);
868
869     cout << "Enter the parts replaced: ";
870     getline(cin, partsReplaced);
871
872     cout << "Enter any additional notes: ";
873     getline(cin, additionalNotes);
874
875     ofstream outfile("service_records.txt", ios::app);
876     if (!outfile) {
877         cout << "Error opening service_records.txt for writing." << endl;
878         return;
879     }
880 }
```



```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
520
521 if (!bookedServicesFile) {
522     cout << RED << "\t          Error opening clients.txt for reading." << RESET << endl;
523     return;
524 }
525 if (!appointmentsFile) {
526     cout << "Error opening appointments.txt for reading." << endl;
527     return;
528 }
529
530 cout << "Booked Services:" << endl;
531 string line;
532 while (getline(bookedServicesFile, line)) {
533     size_t pos1 = line.find(';');
534     size_t pos2 = line.find(',', pos1 + 1);
535     size_t pos3 = line.find(',', pos2 + 1);
536
537     if (pos1 != string::npos && pos2 != string::npos && pos3 != string::npos) {
538         string name = line.substr(0, pos1);
539         string serviceType = line.substr(pos1 + 1, pos2 - pos1 - 1);
540         string date = line.substr(pos2 + 1, pos3 - pos2 - 1);
541         string status = line.substr(pos3 + 1);
542
543         cout << "Service Tvpce: " << serviceTvpce << "\n" << " Date: " << date << "\n" << " Status: " << status << "\n"
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
366
367     cout << BLUE << "          VEHICLE REPAIRING   \n" << RESET;
368     cout << "\n";
369
370     cout << "    Professional vehicle detailing using world reputed\nchemicals and products to make your vehicle look
371     cout << "\n";
372     cout << "\n";
373
374 }
375 // Function to make an appointment
376 void appointment() {
377     system("cls");
378     string appointmentname, appointmentDate, appointmentDetails, phonenumber, requestacall;
379     bool overlap = false;
380     cout << BLUE << "          Appointment Form\n\n\n" << RESET;
381
382     cout << "\t          Enter your appointment sender's name: ";
383     cin.ignore();
384     getline(cin, appointmentname);
385
386     do {
387         cout << "\t          Enter your appointment Date (YYYY-MM-DD): ";
388         cin >> appointmentDate;
389     } while (!isValidDate(appointmentDate));
```



The screenshot shows a C++ IDE with the following components:

- Title Bar:** C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for file operations (open, save, print), editing (undo, redo), and execution (run, debug).
- Compiler:** TIM-GCC 4.9.2 64-bit Release
- Project Explorer:** Shows a project named 'ugc0323017+amoda.cpp'.
- Code Editor:** Displays the following C++ code:

```
936  
937  
938     while (getline(infile, line)) {  
939         // Display each canceled appointment's details  
940         cout << "Canceled Appointment:" << endl;  
941         cout << "Name: " << line << endl;  
942  
943         // Read and display additional details  
944         getline(infile, line);  
945         cout << "Date: " << line << endl;  
946  
947         getline(infile, line);  
948         cout << "Details: " << line << endl;  
949  
950         getline(infile, line);  
951         cout << "Reason for Canceling: " << line << endl;  
952  
953         cout << endl;  
954     }  
955     infile.close();  
956 }  
957 // end of the system  
958
```

The screenshot shows a C++ IDE with the following code:

```

245 cout << "\n";
246 cout << "\t" << "View the service status" << endl;
247 cout << "\n";
248 cout << "\t" << "View the Service Records" << endl;
249 cout << "\n";
250 cout << "\t" << "View the Service Completion Status" << endl;
251 cout << "\n";
252 cout << "\t" << "Back to Main Menu" << endl;
253 cout << "\n";
254 cout << "\n";
255 cout << "\t" << "Enter your choice"<<BLUE<<"(1/2/3/4/5/6/7/8/9): " <<RESET;
256 cin >> choice;
257 switch (choice) {
258     case 1:
259         ourservices();
260         break;
261     case 2:
262         bookService();
263         break;
264     case 3:
265         appointment();
266         break;
267     case 4:

```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
449     if (year < 2000 || year > 2100 || month < 1 || month > 12 || day < 1 || day > 31) {
450         cout <<RED<< "Invalid date. Please enter a valid date." <<RESET<< endl;
451         return false;
452     }
453
454     return true;
455 }
456
457 // Helper function to convert string to integer
458 int stringToInt(const string& str) {
459     int result = 0;
460     stringstream ss(str);
461     ss >> result;
462     return result;
463 }
464
465 // Helper function to check if all characters in a string are digits
466 bool isAllDigits(const string& str) {
467     for (size_t i = 0; i < str.length(); ++i) {
468         if (!isdigit(str[i])) {
469             return false;
470         }
471     }
472     return true;
473 }
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
617     } else {
618         cout << "User not found!" << endl;
619     }
620 }
621 // Function to view service records
622 void viewServiceRecords() {
623     system("cls");
624     ifstream infile("service_records.txt");
625     if (!infile) {
626         cout << "Error opening service_records.txt for reading." << endl;
627         return;
628     }
629
630     string line;
631     cout << "\nService Records:" << endl;
632     while (getline(infile, line)) {
633         size_t pos1 = line.find(';');
634         size_t pos2 = line.find(',', pos1 + 1);
635         size_t pos3 = line.find(',', pos2 + 1);
636
637         if (pos1 != string::npos && pos2 != string::npos && pos3 != string::npos) {
638             string username = line.substr(0, pos1);
639             string servicePerformed = line.substr(pos1 + 1, pos2 - pos1 - 1);
640             string partsReplaced = line.substr(pos2 + 1, pos3 - pos2 - 1);
641         }
642     }
643 }
```



```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
579         return;
580     }
581
582     string id, pass, phone, line;
583     bool found = false;
584
585     while (infile >> id >> pass >> phone) {
586         if (id == username) {
587             found = true;
588
589             cout << "Enter new password: ";
590             cin >> newPassword;
591             cout << "Enter new phone number: ";
592             cin >> phoneNumber;
593             cout << "Enter new license ID: ";
594             cin >> licenseId;
595             cout << "Enter new vehicle ID: ";
596             cin >> vehicleId;
597             cout << "Enter new vehicle type: ";
598             cin >> vehicleType;
599
600             tempFile << username << " " << newPassword << " " << phoneNumber << endl;
601         }
    }
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
657         return;
658     }
659
660     string line;
661     cout << "\nService Completion Status:" << endl;
662     while (getline(infile, line)) {
663         size_t pos1 = line.find(';');
664         size_t pos2 = line.find(';', pos1 + 1);
665
666         if (pos1 != string::npos && pos2 != string::npos) {
667             string username = line.substr(0, pos1);
668             string serviceStatus = line.substr(pos1 + 1, pos2 - pos1 - 1);
669             string completionDate = line.substr(pos2 + 1);
670
671             cout << "Username: " << username << "\nService Status: " << serviceStatus
672                  << "\nCompletion Date: " << completionDate << "\n" << endl;
673         } else {
674             cout << "Error parsing service completion data." << endl;
675         }
676     }
677     infile.close();
678 }
679
680 }
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp

129     system("cls");
130     break;
131 }
132 }
133 input.close();
134
135 if (count == 1) {
136     if (userid.substr(0, 2) == "cl") {
137         cout << RED << "userid << RESET << "\nYour LOGIN is successful\n";
138         clientMenu();
139     } else if (userid.substr(0, 2) == "se") {
140         cout << RED << "userid << RESET << "\nYour LOGIN is successful\n";
141         serviceProviderMenu();
142     } else {
143         cout << RED << " \nLOGIN ERROR\nInvalid user type. Please check your username.\n" << RES
144     }
145     main();
146 } else {
147     cout << RED << " \nLOGIN ERROR\nPlease check your username and password\n" << RESET << endl
148     main();
149 }
150 }
151 }
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp

268     cancelappointment();
269     break;
270 case 5:
271     updateCustomerDetails();
272     break;
273 case 6:
274     viewServiceStatus();
275     break;
276 case 7:
277     viewServiceRecords();
278     break;
279 case 8:
280     viewServiceCompletionStatus();
281     break;
282 case 9:
283     main();
284     break;
285 default:
286     system("cls");
287     cout << " Invalid choice. Try again." << endl;
288 }
289 } while (choice != 9);
290 }
291 // client contents
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp

426 // Function to validate phone number
427 bool isValidPhoneNumber(const string& phoneNumber) {
428     if (phoneNumber.length() != 10 || !isAllDigits(phoneNumber)) {
429         cout << RED << "Invalid phone number. It should be exactly 10 digits long." << RESET << endl;
430         return false;
431     }
432     return true;
433 }
434
435 // Function to validate date
436 bool isValidDate(const string& date) {
437     if (date.length() != 10 || date[4] != '-' || date[7] != '-' ||
438         !isdigit(date[0]) || !isdigit(date[1]) || !isdigit(date[2]) || !isdigit(date[3]) ||
439         !isdigit(date[5]) || !isdigit(date[6]) || !isdigit(date[8]) || !isdigit(date[9])) {
440         cout << RED << "Invalid date format. Please use YYYY-MM-DD." << RESET << endl;
441         return false;
442     }
443
444     int year = stringToInt(date.substr(0, 4));
445     int month = stringToInt(date.substr(5, 2));
446     int day = stringToInt(date.substr(8, 2));
447
448     if (year < 2000 || year > 2100 || month < 1 || month > 12 || day < 1 || day > 31) {
449
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp

549     cout << "Booked Appointments:" << endl;
550     while (getline(appointmentsFile, line)) {
551         size_t pos1 = line.find(';');
552         if (pos1 != string::npos) {
553             string date = line.substr(0, pos1);
554
555             cout << "Date : " << date << "\n" << endl;
556         } else {
557             cout << "Error parsing appointment data." << endl;
558         }
559     }
560
561     bookedServicesFile.close();
562     appointmentsFile.close();
563 }
564
565 // Function to update customer details
566 void updateCustomerDetails() {
567     system("cls");
568     string username, newPassword, phoneNumber, licenseId, vehicleId, vehicleType;
569
570     cout << "Enter the username of the account you want to update: ";
571     cin >> username;
```



```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
288 }
289 } while (choice != 9);
290 }
291 // client contents
292 // Function to book a service
293 void bookService() {
294     system("cls");
295     ofstream outfile("clients.txt", ios::app);
296     if (!outfile) {
297         cout << RED << "Error opening file for writing." << RESET << endl;
298         return;
299     }
300     cout << BLUE << "Choose & Book your service from the OUR SERVICES\n\n\n\n" << R
301     string name, serviceType, date;
302     cout << "\t Enter your username : ";
303     cin.ignore();
304     getline(cin, name);
305     cout << "\t Enter service type : ";
306     getline(cin, serviceType);
307     cout << "\t Enter preferred date "<< RED << "(YYYY-MM-DD) : "<< RESET;
308     getline(cin, date);
309
310     outfile << name << ";" << serviceType << ";" << date << ";Pending\n";
311     outfile.close();
}
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
204 }
205 f2.close();
206
207 if (count == 1) {
208     cout << BLUE << "\n\n Your account is found!\n" << RESET;
209     cout << BLUE << "\n\n Your password is : " << RESET << spass << endl;
210     main();
211 } else {
212     cout << RED << "\n\t Sorry! Your account is not found!\n" << RESET;
213     main();
214 }
215 break;
216 }
217 case 2:
218     main();
219     break;
220 default:
221     cout << RED << "\t\t\t Wrong choice! Please try again" << RESET << endl;
222     forgot();
223     main();
224 }
225 }
226 // access to client part
227 void clientMenu() {
}
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
318 }
319 // Function to display available services
320 void ourservices(){
321     system("cls");
322     cout<<RED<<"                OUR SERVICES                \n "<<RESET;
323     cout <<"\n";
324     cout <<"\n";
325     cout <<"\n";
326
327     cout<<BLUE<<"                RUNNING REPAIRS                \n"<<RESET;
328     cout <<"\n";
329
330     cout<<"    Day to day running repairs on all brands\nof vehicles using genuine parts\n";
331     cout <<"\n";
332     cout <<"\n";
333
334     cout<<BLUE<<"                TUNE UP'S                \n"<<RESET;
335     cout <<"\n";
336
337     cout<<"    Electronic injector cleaning and tune up's\nfor both petrol and diesel vehicles\n";
338     cout <<"\n";
339     cout <<"\n";
340
341 }
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
474
475 // Function to cancel an appointment
476 void cancelappointment() {
477     system("cls");
478     cout <<BLUE<<"                Appointment Cancel Form\n\n\n"<<RESET;
479     ofstream outfile("cancelappointments.txt", ios::app);
480     if (!outfile) {
481         cout <<RED<<"Error opening file for writing." <<RESET<< endl;
482         return;
483     }
484
485     string appointmentname;
486     cout <<"\t                Enter your appointment sender's name: ";
487     cin.ignore();
488     getline(cin, appointmentname);
489     outfile << appointmentname <<"\n";
490
491     string appointmentDate;
492     cout <<"\t                Enter your appointment & service Date "<<RED<<"(YYYY-MM-DD): "<<RESET;
493     cin.ignore();
494     getline(cin, appointmentDate);
495     outfile << appointmentDate <<"\n";
496
497     string appointmentDetails;
```



```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
397         overlap = true;
398         break;
399     }
400 }
401 infile.close();
402
403 if (overlap) {
404     cout << RED << "\t                An appointment is already booked on this date. Please choose a different da
405     return;
406 }
407 cout << RED << "sorry for the inconvenience there is small error here,plz neglect it"<< RESET;
408 // Get additional details and save appointment
409 cout << "\t                Enter your appointment Details: \n";
410 getline(cin, appointmentDetails);
411
412 cout << "\t                Enter your phone number: ";
413 getline(cin, phonenumber);
414
415 cout << "\t                Request a phone call "<< RED << "(Yes/No):"<< RESET;
416 getline(cin, requestacall);
417
418 ofstream outfile("appointments.txt", ios::app);
419 outfile << appointmentDate << ";" << appointmentname << ";" << appointmentDetails << ";" << phonenumber << ";";
420 outfile.close();

C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
1 // Include necessary header files
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5 #include <cstdlib>
6 #include <cctype>
7 #include <sstream>
8
9 // Define color codes for console output
10 #define RESET "\033[0m"
11 #define RED "\033[31m"
12 #define BLUE "\033[34m"
13
14 using namespace std;
15
16 // Function prototypes for input validation(DATA VALIDATION)
17 bool isValidPhoneNumber(const string& phoneNumber);
18 bool isValidDate(const string& date);
19 int stringToInt(const string& str);
20 bool isAllDigits(const string& str);
21
22 // Function for login
23 void login();
24 void registration();

Compiler Resources Compile Log Debug Find Results
Line: 49 Col: 64 Sel: 0 Lines: 958 Length: 34678 Insert Done parsing in 0.062 seconds
```



```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
342 cout<<BLUE<<"                FULL SERVICE          \n"<<RESET;
343 cout <<"\n";
344
345 cout<<"    Full service for all brands of vehicles\nusing manufacturer recommended oils\n";
346 cout <<"\n";
347 cout <<"\n";
348
349
350
351 cout<<BLUE<<"                VEHICLE ACCESSORIES      \n"<<RESET;
352 cout <<"\n";
353
354 cout<<"    All vehicle accessories including leather seat covers,\ntinting, GPS/DVD players, etc.\n";
355 cout <<"\n";
356 cout <<"\n";
357
358
359 cout<<BLUE<<"                COLLISION REPAIRS      \n"<<RESET;
360 cout <<"\n";
361
362 cout<<"    2K heated booth painting with 3 year warranty\nnon all paint job\n";
363 cout <<"\n";
364 cout <<"\n";
365
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
81 cout << "*"
82 cout << "*****"
83 cout << "Enter your choice"<<BLUE<<"(1/2/3/4):"<<RESET;
84 cin >> choice;
85 switch (choice) {
86     case 1:
87         login();
88         break;
89     case 2:
90         registration();
91         break;
92     case 3:
93         forgot();
94         break;
95     case 4:
96         cout <<RED<< "                Exiting the system." <<RESET<< endl;
97         break;
98     default:
99         cout <<RED<< "                Invalid choice. Try again." <<RESET<< endl;
100 }
101 } while (choice != 4);
102
103 return 0;
104
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
21
22 // Function for login
23 void login();
24 void registration();
25 void forgot();
26
27 //functions for client
28 void clientMenu();
29 void bookService();
30 void ourservices();
31 void appointment();
32 void cancelappointment();
33 void viewServiceStatus();
34 void updateCustomerDetails();
35 void viewServiceCompletionStatus();
36 void viewServiceRecords();
37
38
39 //functions of serverside
40 void serviceProviderMenu();
41 void viewBookings();
42 void displayAppointments();
43 void searchVehicleByRegistration();
44 void searchBookingsByUsername();
```

```
C:\Users\Administrator\Desktop\night final\ugc0323017+amoda.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug ugc0323017+amoda.cpp
817
818 if (!found) {
819     cout << "No vehicle found with the registration number: " << registrationNumber << endl;
820 }
821
822 infile.close();
823 }
824 // Function to search bookings by username
825 void searchBookingsByUsername() {
826     system("cls");
827     string username, name, serviceType, date, status;
828     bool found = false;
829
830     cout << "Enter the username to search for bookings: ";
831     cin >> username;
832
833     ifstream infile("clients.txt");
834     if (!infile) {
835         cout << "Error opening clients.txt for reading." << endl;
836         return;
837     }
838
839     while (getline(infile, name, ';')) {
840         getline(infile, serviceType, ':');
```