



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Laundry Management System

Blog post introduction

In the fast-paced world we live in, managing day-to-day tasks efficiently is crucial. From mundane chores to complex operations, technology plays a vital role in simplifying our lives. In this blog post, we will explore a practical example of how C# console application can be leveraged to create a Laundry Management System. This project encompasses user authentication, customer information handling, service display, and even a simulated payment process.

About the System

Laundry management is an essential aspect of our daily routines, and streamlining the process can save time and effort. The Laundry Management System developed in C# console application provides a user-friendly interface to handle various tasks related to customer information and laundry services.

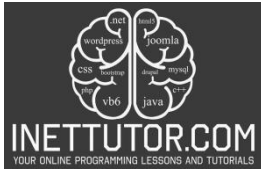
Flow of the System

1. Login using if else statement, display welcome message
2. Select option to execute using switch statement
 - a. 1 - for customer info
 - b. 2 for Services offered
3. If 1 then the system will ask to input the name, contact and address of the customer
4. Select option to execute:
 - 1 for customer info
 - 2 for Services offered

If 1 then the system will ask to input the name, contact and address

If 2 then the system will list the services offered with prices

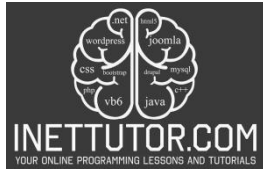
5. The system will ask the user to input the service choice, amount or number of dress then the system will compute the total amount
6. The system will then compute the total amount and the payment process will happen.



Example Source code

```
1. using System;
2.
3. class Program
4. {
5.     static void Main()
6.     {
7.         string username = "admin";
8.         string password = "password";
9.
10.        Console.WriteLine("Laundry Management System");
11.        Console.WriteLine("-----");
12.
13.        // Login
14.        Console.Write("Enter username: ");
15.        string enteredUsername = Console.ReadLine();
16.        Console.Write("Enter password: ");
17.        string enteredPassword = Console.ReadLine();
18.
19.        if (enteredUsername == username && enteredPassword == password)
20.        {
21.            Console.WriteLine("Welcome, {0}!", username);
22.
23.            // Display menu
24.            Console.WriteLine("\nSelect an option:");
25.            Console.WriteLine("1 - Customer Info");
26.            Console.WriteLine("2 - Services Offered");
27.            Console.WriteLine("0 - Exit");
28.
29.            // Get user input
30.            Console.Write("Enter option: ");
31.            string input = Console.ReadLine();
32.
33.            // Validate input
34.            if (int.TryParse(input, out int option))
35.            {
36.                if (option == 1)
37.                {
38.                    CustomerInfo();
39.                }
40.                else if (option == 2)
41.                {
42.                    DisplayServices();
43.
44.                    // Additional steps for service choice and payment
45.                    ProcessLaundryService();
46.                }
47.                else if (option == 0)
48.                {
49.                    Console.WriteLine("Exiting the program. Goodbye!");
50.                }
51.                else
52.                {
53.                    Console.WriteLine("Invalid option. Please try again.");
54.                }

```



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

```
55.         }
56.         else
57.         {
58.             Console.WriteLine("Invalid input. Please enter a valid option.");
59.         }
60.     }
61.     else
62.     {
63.         Console.WriteLine("Invalid username or password. Exiting program.");
64.     }
65. }
66.
67. static void CustomerInfo()
68. {
69.     Console.WriteLine("\nEnter customer details:");
70.
71.     Console.Write("Name: ");
72.     string name = Console.ReadLine();
73.
74.     Console.Write("Contact: ");
75.     string contact = Console.ReadLine();
76.
77.     Console.Write("Address: ");
78.     string address = Console.ReadLine();
79.
80.     Console.WriteLine("\nCustomer Information:");
81.     Console.WriteLine("Name: {0}", name);
82.     Console.WriteLine("Contact: {0}", contact);
83.     Console.WriteLine("Address: {0}", address);
84. }
85.
86. static void DisplayServices()
87. {
88.     Console.WriteLine("\nServices Offered:");
89.
90.     // You can add more services and their prices as needed
91.     Console.WriteLine("1. Regular Wash - $5.00");
92.     Console.WriteLine("2. Dry Cleaning - $8.00");
93.     Console.WriteLine("3. Express Service - $10.00");
94.     // Add more services as needed
95. }
96.
97. static void ProcessLaundryService()
98. {
99.     Console.Write("\nEnter the service choice (1, 2, 3): ");
100.    if (int.TryParse(Console.ReadLine(), out int serviceChoice))
101.    {
102.        Console.Write("Enter the number of dresses: ");
103.        if (int.TryParse(Console.ReadLine(), out int numberOfDresses))
104.        {
105.            double totalAmount = CalculateTotalAmount(serviceChoice,
106.                numberOfDresses);
107.            Console.WriteLine("\nTotal Amount: ${0}", totalAmount);
108.
109.            // Simulate payment process
110.            Console.Write("Enter payment amount: $");
```



```
111.             if (double.TryParse(Console.ReadLine(), out double
    paymentAmount) && paymentAmount >= totalAmount)
112.             {
113.                 Console.WriteLine("Payment successful. Thank you!");
114.             }
115.             else
116.             {
117.                 Console.WriteLine("Invalid payment amount. Payment
    failed.");
118.             }
119.         }
120.     else
121.     {
122.         Console.WriteLine("Invalid input for the number of dresses.
    Please enter a valid number.");
123.     }
124. }
125. else
126. {
127.     Console.WriteLine("Invalid input for the service choice. Please
    enter a valid number.");
128. }
129.     Console.ReadKey();
130. }
131.
132.     static double CalculateTotalAmount(int serviceChoice, int numberOfDresses)
133.     {
134.         // Prices for different services
135.         double[] servicePrices = { 5.00, 8.00, 10.00 };
136.
137.         if (serviceChoice >= 1 && serviceChoice <= servicePrices.Length)
138.         {
139.             double servicePrice = servicePrices[serviceChoice - 1];
140.             return servicePrice * numberOfDresses;
141.         }
142.         else
143.         {
144.             Console.WriteLine("Invalid service choice. Unable to calculate total
    amount.");
145.             return 0;
146.         }
147.         Console.ReadKey();
148.     }
149. }
```

Explanation

Login Section:

Security is paramount, even in a console application. The project begins with a simple yet effective user authentication mechanism. Users are required to input a username and password, providing a secure gateway to the laundry management system. This ensures that only authorized personnel can access and operate the system.



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

```
1. string username = "admin";
2. string password = "password";
3.
4. Console.WriteLine("Laundry Management System");
5. Console.WriteLine("-----");
6.
7. // Login
8. Console.Write("Enter username: ");
9. string enteredUsername = Console.ReadLine();
10. Console.Write("Enter password: ");
11. string enteredPassword = Console.ReadLine();
12.
13. if (enteredUsername == username && enteredPassword == password)
```

- The program starts by defining a username and password.
- The user is prompted to enter a username and password.
- If the entered credentials match the predefined ones, the program proceeds; otherwise, it exits.

Main Menu Section:

The project adopts a menu-driven interface to enhance user experience. Once authenticated, users are presented with a menu displaying options such as managing customer information, exploring laundry services, and exiting the system. The intuitive design ensures that users can easily navigate through the application.

```
1. Console.WriteLine("Welcome, {0}!", username);
2.
3. // Display menu
4. Console.WriteLine("\nSelect an option:");
5. Console.WriteLine("1 - Customer Info");
6. Console.WriteLine("2 - Services Offered");
7. Console.WriteLine("0 - Exit");
8.
9. // Get user input
10. Console.Write("Enter option: ");
11. string input = Console.ReadLine();
12.
13. // Validate input
14. if (int.TryParse(input, out int option))
```

- Displays a welcome message and the main menu with options.
- Prompts the user to enter an option.
- Validates whether the entered input is a valid integer.

Option Handling:

```
1. if (option == 1)
2. {
3.     CustomerInfo();
4. }
```



```
5. else if (option == 2)
6. {
7.     DisplayServices();
8.
9.     // Additional steps for service choice and payment
10.    ProcessLaundryService();
11. }
12. else if (option == 0)
13. {
14.    Console.WriteLine("Exiting the program. Goodbye!");
15. }
16. else
17. {
18.    Console.WriteLine("Invalid option. Please try again.");
19. }
```

- If the user chooses option 1, it calls the CustomerInfo method to gather and display customer details.
- If the user chooses option 2, it calls the DisplayServices method to show available services and then calls ProcessLaundryService to handle service choice and payment.
- If the user chooses option 0, it exits the program.
- If the user enters an invalid option, an error message is displayed.

Customer Information and Services Display:

The system enables the input and retrieval of customer information. Users can seamlessly enter customer details, including name, contact information, and address. The system then organizes and displays this information, providing a comprehensive overview of the clientele.

One of the key functionalities of the Laundry Management System is the ability to display available laundry services. Users are presented with a menu showcasing different services, each with its associated price. This feature ensures transparency and helps customers make informed choices.

```
1. static void CustomerInfo()
2. {
3.     Console.WriteLine("\nEnter customer details:");
4.
5.     Console.Write("Name: ");
6.     string name = Console.ReadLine();
7.
8.     Console.Write("Contact: ");
9.     string contact = Console.ReadLine();
10.
11.    Console.Write("Address: ");
12.    string address = Console.ReadLine();
13.
14.    Console.WriteLine("\nCustomer Information:");
15.    Console.WriteLine("Name: {0}", name);
16.    Console.WriteLine("Contact: {0}", contact);
17.    Console.WriteLine("Address: {0}", address);
```



```
18.     }
19.
20.     static void DisplayServices()
21.     {
22.         Console.WriteLine("\nServices Offered:");
23.
24.         // You can add more services and their prices as needed
25.         Console.WriteLine("1. Regular Wash - $5.00");
26.         Console.WriteLine("2. Dry Cleaning - $8.00");
27.         Console.WriteLine("3. Express Service - $10.00");
28.         // Add more services as needed
29.     }
```

- CustomerInfo() collects and displays customer details.
- DisplayServices() shows a list of available services and their prices.

Laundry Service Processing:

The project goes beyond mere information display by incorporating a streamlined process for handling laundry services. Users can input their service choice and the number of dresses they wish to launder. The system, in turn, computes the total amount based on the selected service and quantity.

Simulated Payment Process:

To mirror real-world scenarios, the application simulates a payment process. Users are prompted to enter the payment amount, and the system validates the payment. This step ensures that the Laundry Management System not only assists in service selection but also provides a seamless transaction experience.

```
1. static void ProcessLaundryService()
2.     {
3.         Console.Write("\nEnter the service choice (1, 2, 3): ");
4.         if (int.TryParse(Console.ReadLine(), out int serviceChoice))
5.         {
6.             Console.Write("Enter the number of dresses: ");
7.             if (int.TryParse(Console.ReadLine(), out int numberOfDresses))
8.             {
9.                 double totalAmount = CalculateTotalAmount(serviceChoice,
10.                    numberOfDresses);
11.                 Console.WriteLine("\nTotal Amount: ${0}", totalAmount);
12.
13.                 // Simulate payment process
14.                 Console.Write("Enter payment amount: $");
15.                 if (double.TryParse(Console.ReadLine(), out double paymentAmount) &&
16.                    paymentAmount >= totalAmount)
17.                 {
18.                     Console.WriteLine("Payment successful. Thank you!");
19.                 }
```



```
19.         else
20.         {
21.             Console.WriteLine("Invalid payment amount. Payment failed.");
22.         }
23.     }
24.     else
25.     {
26.         Console.WriteLine("Invalid input for the number of dresses. Please
enter a valid number.");
27.     }
28. }
29. else
30. {
31.     Console.WriteLine("Invalid input for the service choice. Please enter a
valid number.");
32. }
33. Console.ReadKey();
34. }
35.
36. static double CalculateTotalAmount(int serviceChoice, int numberOfDresses)
37. {
38.     // Prices for different services
39.     double[] servicePrices = { 5.00, 8.00, 10.00 };
40.
41.     if (serviceChoice >= 1 && serviceChoice <= servicePrices.Length)
42.     {
43.         double servicePrice = servicePrices[serviceChoice - 1];
44.         return servicePrice * numberOfDresses;
45.     }
46.     else
47.     {
48.         Console.WriteLine("Invalid service choice. Unable to calculate total
amount.");
49.         return 0;
50.     }
51.     Console.ReadKey();
52. }
```

- ProcessLaundryService() handles the service choice, number of dresses input, calculates the total amount, and simulates a payment process.
- CalculateTotalAmount() calculates the total amount based on the selected service and the number of dresses.

Conclusion:

In conclusion, this C# console application exemplifies the power and versatility of the programming language in creating practical solutions. The Laundry Management System, with its user-friendly interface and efficient functionalities, showcases how technology can simplify everyday tasks. This project serves as a foundation for further enhancements and customization, demonstrating the adaptability of C# in meeting specific requirements.



INetTutor.com

Online Programming Lessons, Tutorials and Capstone Project guide

Stay tuned for the next parts of this blog series where we delve deeper into the code, exploring individual components and understanding how each contributes to the overall functionality of the Laundry Management System.