**Program 1:**

#include <iostream>

using namespace std;

int main()

{

int a=2,b=3,tem;

cout<<"Before swiping:"<<endl<<"a="<<a<<endl<<"b="<<b<<endl;

tem=a;

a=b;

b=tem;

cout<<"After swiping:"<<endl<<"a="<<a<<endl<<"b="<<b<<endl;

return 0;

}

**Program:2**

#include <iostream>

#

using namespace std;

int main()

{

int x=3;

double y=5;

double z=(x+10)/(3\*y);

cout<<z;

return 0;

}

**Program:3**

#include <iostream>

#include <ctime>

#include <cstdlib>

using namespace std;

int main()

{

const int minValue=1;

const int maxValue=6;

srand(time(0));

short first=(rand()%(maxValue-me+1))+minValue;

short second=(rand()%(maxValue-minValue+1))+minValue;

cout<<first<<", "<<second;

return 0;

}

**Program:4**

#include <iostream>

#include<string>

using namespace std;

int main()

{

string str,reversedstr;

cout<<"Enter a string:";

cin>>str;

reversedstr=string(str.rbegin(),str.rend());

cout<<reversedstr;

return 0;

}

**Program:5**

#include <iostream>

#include<string>

using namespace std;

int main()

{

string str,reversedstr;

cout<<"Enter a string:";

cin>>str;

reversedstr=string(str.begin(),str.end());

if(str==reversedstr)

cout<<"The string is palindrome."<<endl;

else

cout<<"The string not palindrome."<<endl;

cout<<reversedstr;

return 0;

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Car Service</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<div class="container">

<h1>Buffer free car is our Pride</h1>

<p>Your trusted car service provider</p>

<title>Explore Services Button</title>

<style>

.explore-button {

display: inline-block;

background-color: #007bff; /\* Blue color \*/

color: white;

text-decoration: none;

padding: 10px 20px;

font-size: 18px;

border-radius: 5px; /\* Rounded corners \*/

text-align: center;

font-family: Arial, sans-serif;

transition: background-color 0.3s ease;

}

.explore-button:hover {

background-color: #0056b3; /\* Darker blue on hover \*/

}

</style>

</head>

<body>

<a href="#services" class="explore-button">Explore Services</a>

</body>

</html>

<title>Social Media Links</title>

<style>

.social-icons {

display: flex;

gap: 10px;

}

.social-icons a {

text-decoration: none;

color: #000;

font-size: 24px;

}

.social-icons a:hover {

color: #0077b5; /\* Change hover color \*/

}

</style>

<title>Social Media Icons</title>

<!-- Add Font Awesome CDN -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">

<style>

.social-icons {

display: flex;

gap: 15px;

margin-top: 20px;

}

.social-icons a {

font-size: 24px;

color: #000; /\* Default icon color \*/

text-decoration: none;

transition: color 0.3s ease;

}

.social-icons a:hover {

color: #0077b5; /\* Hover color \*/

}

</style>

</head>

<body>

<div class="social-icons">

<a href="https://instagram.com" target="\_blank" title="Instagram">

<i class="fab fa-instagram"></i>

</a>

<a href="https://facebook.com" target="\_blank" title="Facebook">

<i class="fab fa-facebook-f"></i>

</a>

<a href="https://twitter.com" target="\_blank" title="Twitter">

<i class="fab fa-twitter"></i>

</a>

</div>

</body>

</html>

<title>Services Section</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

box-sizing: border-box;

}

.services-section {

padding: 50px;

max-width: 1200px;

margin: 0 auto;

}

.services-header {

margin-bottom: 30px;

}

.services-header h2 {

font-size: 36px;

color: #333;

position: relative;

}

.services-header h2::before {

content: "";

position: absolute;

left: 0;

bottom: -10px;

width: 50px;

height: 3px;

background-color: #007bff;

}

.services-header p {

font-size: 16px;

color: #666;

max-width: 600px;

margin-top: 10px;

}

.services-container {

display: flex;

gap: 20px;

justify-content: space-between;

aligne-items:flex-start;

}

.service-box {

flex: 1;

max-width: 300px;

border: 1px solid #007bff;

padding: 20px;

border-radius: 10px;

text-align: center;

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.service-box:hover {

transform: translateY(-5px);

box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);

}

.service-box h3 {

font-size: 20px;

color: #333;

margin-bottom: 10px;

}

.service-box p {

font-size: 14px;

color: #666;

}

</style>

</head>

<body>

<div class="services-section">

<div class="services-header">

<h2>Services</h2>

<p>We provide a variety of high-quality services to meet your needs.Whether it's car maintenance, cleaning, or repair, we ensure the best results with our expert team.</p>

</div>

<div class="services-container">

<div class="service-box">

<h3>Car Wash</h3>

<p>Professional car cleaning services with the latest equipment and eco-friendly products.Keep your vehicle shining like new.</p>

</div>

<div class="service-box">

<h3>Chemical Wash</h3>

<p>Deep cleaning for stubborn stains and grims.Perfect for removing tough dirt from hard-to-reach areas.</p>

</div>

<div class="service-box">

<h3>Steam Wash</h3>

<p>An advance cleaning services using high-pressure steam to sanitize and clean your car without harming the environment .</p>

</div>

</div>

</div>

</body>

</html>

<section id="process">

<h2>Process</h2>

<div class="process-steps">

<div class="step">1. Book Service</div>

<div class="step">2. Car Cleaning</div>

<div class="step">3. Interior Detailing</div>

<div class="step">4. Final Polishing</div>

</div>

</section>

<section id="pricing">

<h2>Pricing</h2>

<div class="pricing-options">

<div class="price">Basic Cleaning: $29</div>

<div class="price">Premium Cleaning: $49</div>

<div class="price">Full Cleaning: $69</div>

</div>

</section>

<section id="team">

<h2>Team & Reviews</h2>

<p>Meet our professional team and see what our clients say!</p>

</section>

<footer>

<p>&copy; 2025 Car Service Co.</p>

</footer>

</body>

</html>

#include <stdio.h>

#include <string.h>

#define MAX\_PRODUCTS 100

// Structure to represent a product

struct Product {

char name[50];

int id;

float price;

int quantity;

};

// Array of products to hold the inventory

struct Product inventory[MAX\_PRODUCTS];

int productCount = 0; // Global variable to keep track of the number of products in inventory

// Function to add a new product to the inventory

void addProduct() {

if (productCount >= MAX\_PRODUCTS) {

printf("Inventory is full! Cannot add more products.\n");

return;

}

struct Product newProduct;

printf("Enter product name: ");

getchar(); // To clear newline from previous input

fgets(newProduct.name, sizeof(newProduct.name), stdin);

newProduct.name[strcspn(newProduct.name, "\n")] = '\0'; // Remove newline character

printf("Enter product ID: ");

scanf("%d", &newProduct.id);

printf("Enter product price: ");

scanf("%f", &newProduct.price);

printf("Enter quantity: ");

scanf("%d", &newProduct.quantity);

inventory[productCount] = newProduct;

productCount++;

printf("Product added successfully!\n");

}

// Function to find a product by ID and display its details

void findProductByID(int id) {

int found = 0;

for (int i = 0; i < productCount; i++) {

if (inventory[i].id == id) {

printf("Product Found:\n");

printf("Name: %s\n", inventory[i].name);

printf("ID: %d\n", inventory[i].id);

printf("Price: %.2f\n", inventory[i].price);

printf("Quantity: %d\n", inventory[i].quantity);

found = 1;

break;

}

}

if (!found) {

printf("Product with ID %d not found!\n", id);

}

}

// Function to update product quantity (pass by reference using pointers)

void updateProductQuantity(int id, int quantityChange) {

int found = 0;

for (int i = 0; i < productCount; i++) {

if (inventory[i].id == id) {

inventory[i].quantity += quantityChange; // Update quantity directly

printf("Product quantity updated! New quantity: %d\n", inventory[i].quantity);

found = 1;

break;

}

}

if (!found) {

printf("Product with ID %d not found!\n", id);

}

}

// Function to print all products in the inventory

void printAllProducts() {

if (productCount == 0) {

printf("No products in the inventory.\n");

return;

}

for (int i = 0; i < productCount; i++) {

printf("Product Name: %s, ID: %d, Price: %.2f, Quantity: %d\n",

inventory[i].name, inventory[i].id, inventory[i].price, inventory[i].quantity);

}

}

// Function to calculate the total inventory value

void calculateTotalValue() {

float totalValue = 0;

for (int i = 0; i < productCount; i++) {

totalValue += inventory[i].price \* inventory[i].quantity; // Sum of price \* quantity

}

printf("Total inventory value: %.2f\n", totalValue);

}

// Main function to drive the program

int main() {

int choice, productID, quantityChange;

printf("Welcome to the Inventory Management System!\n");

do {

// Display menu options

printf("\n1. Add Product\n");

printf("2. Find Product\n");

printf("3. Update Quantity\n");

printf("4. Print All Products\n");

printf("5. Calculate Total Value\n");

printf("6. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addProduct();

break;

case 2:

printf("Enter product ID to find: ");

scanf("%d", &productID);

findProductByID(productID);

break;

case 3:

printf("Enter product ID to update quantity: ");

scanf("%d", &productID);

printf("Enter quantity change (positive or negative): ");

scanf("%d", &quantityChange);

updateProductQuantity(productID, quantityChange);

break;

case 4:

printAllProducts();

break;

case 5:

calculateTotalValue();

break;

case 6:

printf("Exiting the system...\n");

break;

default:

printf("Invalid choice! Please try again.\n");

}

} while (choice != 6);

return 0;

}