Name:-Jitendra Gupta

EmailID:-jitendra.g.gupta@pwc.com

**Foundation Assignment: Calculator(Part-i)**

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

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

  <title>Calculator</title>

</head>

<body>

  <table>

    <th colspan="4"><input type="text" id="Display" disabled></th>

    <tr>

        <td><input type="button" value="1" onclick="insertElem(1)"></td>

        <td><input type="button" value="2" onclick="insertElem(2)"></td>

        <td><input type="button" value="3" onclick="insertElem(3)"></td>

        <td><input type="button" value="+" onclick="insertElem('+')"></td>

    </tr>

    <tr>

        <td><input type="button" value="4" onclick="insertElem(4)"></td>

        <td><input type="button" value="5" onclick="insertElem(5)"></td>

        <td><input type="button" value="6" onclick="insertElem(6)"></td>

        <td><input type="button" value="-" onclick="insertElem('-')"></td>

    </tr>

    <tr>

        <td><input type="button" value="7" onclick="insertElem(7)"></td>

        <td><input type="button" value="8" onclick="insertElem(8)"></td>

        <td><input type="button" value="9" onclick="insertElem(9)"></td>

        <td><input type="button" value="\*" onclick="insertElem('\*')"></td>

    </tr>

    <tr>

        <td><input type="button" value="/" onclick="insertElem('/')"></td>

        <td><input type="button" value="0" onclick="insertElem('0')"></td>

        <td><input type="button" value="." onclick="insertElem('.')"></td>

        <td><input type="button" value="=" onclick="Equal()"></td>

    </tr>

    <th colspan="4"><input type="button" value="Clear All" class="Clear" onclick="Clear()"></th>

  </table>

  <script src="script.js"></script>

</body>

</html>

CSS:

\*{

    margin: 0;

    padding: 0;

}

body{

    background-image: url(1.jpg);

    background-position: center;

    background-attachment: fixed;

    background-repeat: no-repeat;

    background-size: cover;

    margin:auto;

}

table{

    text-align: center;

    margin: auto;

    position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

    border: 1px solid black;

    border-radius: 5px;

    padding: 10px 30px;

    background: rgba(255, 255, 255,0.8) ;

}

table tr td input[type=button]{

    width: 85%;

    padding: 18px 22px;

    margin: 10px 5px;

    box-sizing: border-box;

    cursor: pointer;

}

table tr td input[type=button]:hover{

    background-color: rgb(77, 75, 75);

}

table tr td input{

    text-align: center;

    font-weight: bolder;

    font-style: normal;

    font-size: 20px;

    color: white;

    background-color: rgb(134, 134, 134);

    border: 1px solid rgb(134, 134, 134);

    border-radius: 5px;

}

.Clear{

    width: 98%;

    padding: 18px 22px;

    text-align: center;

    font-weight: bolder;

    font-style: normal;

    color: white;

    background-color: rgb(134, 134, 134);

    border: 1px solid rgb(134, 134, 134);

    border-radius: 5px;

    cursor: pointer;

}

.Clear:hover{

    background-color: rgb(77, 75, 75);

}

#Display{

    margin-bottom: 5px;

    padding: 15px 22px;

    width:80%;

    font-weight: bolder;

    font-size: 20px;

    color: white;

    background-color: rgb(134, 134, 134);

    border: 1px solid rgb(134, 134, 134);

    border-radius: 5px;

}

JAVASCRIPT CODE:

document.getElementById("Display").value="";

function insertElem(num){

    return document.getElementById("Display").value+=num;

}

function Equal(){

    return document.getElementById("Display").value=eval(document.getElementById("Display").value);

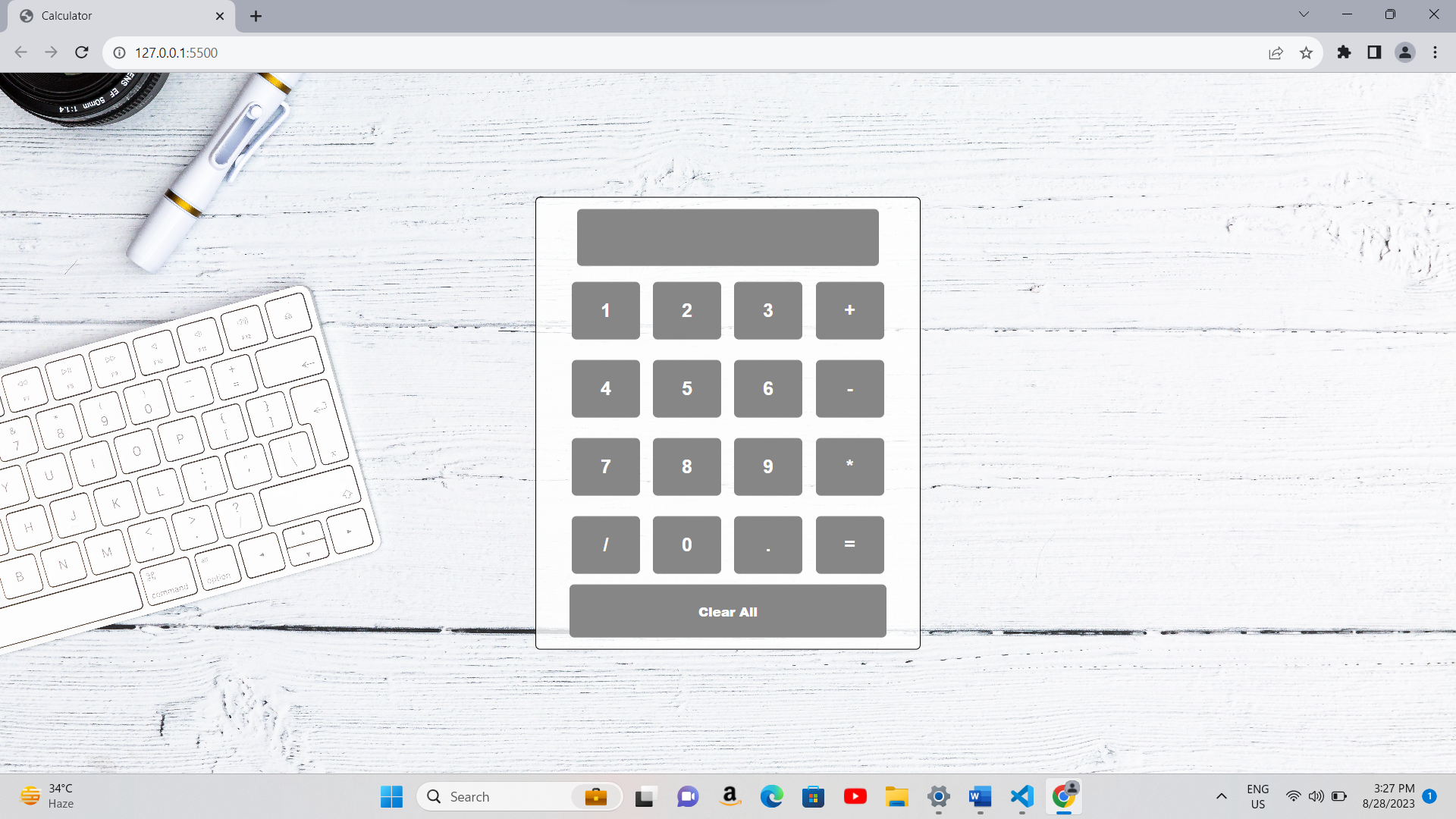
}

function Clear(){

    return document.getElementById("Display").value="";

}

OUTPUT:



**Foundation Assignment: e-COMMERCE(Part-Ii)**

C-Sharp Code:-

using System;

using System.Collections.Generic;

class User

{

    private bool role;

    private string loginId;

    private string password;

    public User(bool role, string loginId, string password)

    {

        this.role = role;

        this.loginId = loginId;

        this.password = password;

    }

    public bool VerifyUser(string inputLoginId, string inputPassword)

    {

        return loginId == inputLoginId && password == inputPassword;

    }

}

class Customer : User

{

    private int id;

    private string address;

    private string phone;

    private string email;

    public Customer(bool role, string loginId, string password, int id, string address, string phone, string email)

        : base(role, loginId, password)

    {

        this.id = id;

        this.address = address;

        this.phone = phone;

        this.email = email;

    }

    public void Login()

    {

        // Implement login logic for customers

    }

    public void Register()

    {

        // Implement registration logic for customers

    }

    public void UpdateProfile()

    {

        // Implement profile update logic for customers

    }

}

class Seller : User

{

    private int id;

    private string address;

    private string phone;

    private string email;

    private double sellerRating;

    public Seller(bool role, string loginId, string password, int id, string address, string phone, string email, double sellerRating)

        : base(role, loginId, password)

    {

        this.id = id;

        this.address = address;

        this.phone = phone;

        this.email = email;

        this.sellerRating = sellerRating;

    }

    public void Login()

    {

        // Implement login logic for sellers

    }

    public void Register()

    {

        // Implement registration logic for sellers

    }

    public void UpdateProfile()

    {

        // Implement profile update logic for sellers

    }

}

class Shopping\_Cart

{

    private Customer customer;

    private DateTime created;

    private List<CartItem> cartItems;

    public Shopping\_Cart(Customer customer)

    {

        this.customer = customer;

        created = DateTime.Now;

        cartItems = new List<CartItem>();

    }

    public void AddCartItems(CartItem item)

    {

        cartItems.Add(item);

    }

    public void CheckOut()

    {

        // Implement checkout logic

    }

    public void ViewCartDetails()

    {

        // Implement logic to display cart details

    }

    public void UpdateQuantity(CartItem item, int newQuantity)

    {

        // Implement logic to update the quantity of a cart item

    }

}

class CartItem

{

    public string ProductName { get; set; }

    public decimal Price { get; set; }

    public int Quantity { get; set; }

}

class Review

{

    private int reviewId;

    private int customerId;

    private string reviewContent;

    private int rating;

    private int parentId;

    private int productId;

    public Review(int reviewId, int customerId, string reviewContent, int rating, int parentId, int productId)

    {

        this.reviewId = reviewId;

        this.customerId = customerId;

        this.reviewContent = reviewContent;

        this.rating = rating;

        this.parentId = parentId;

        this.productId = productId;

    }

    public void AddReview(int reviewId, int customerId, string reviewContent, int rating, int parentId, int productId)

    {

        // Implement logic to add a new review

    }

    public void EditReview(int reviewId, string newReviewContent, int newRating)

    {

        // Implement logic to edit an existing review

    }

    public void DeleteReview(int reviewId)

    {

        // Implement logic to delete a review

    }

}

class Product

{

    private string productID;

    private string productName;

    private int productCost;

    private string sellerID;

    private DateTime postedDate;

    private List<Review> reviews;

    private List<Order> orders;

    private Shopping\_Cart shoppingCart; // Aggregation relationship

    public Product(string productID, string productName, int productCost, string sellerID, DateTime postedDate)

    {

        this.productID = productID;

        this.productName = productName;

        this.productCost = productCost;

        this.sellerID = sellerID;

        this.postedDate = postedDate;

        this.reviews = new List<Review>();

        this.orders = new List<Order>();

    }

    public void AddToCart(Shopping\_Cart cart)

    {

        // Implement logic to add the product to a shopping cart

        this.shoppingCart = cart;

    }

    public void SellProduct()

    {

        // Implement logic to sell the product

    }

    public void BuyProduct()

    {

        // Implement logic to buy the product

    }

    public string GetProductDetails()

    {

        // Implement logic to retrieve and return product details

        return $"{productName} - Cost: {productCost}";

    }

}

class Order

{

    private string id;

    private string sellerID;

    private string customerID;

    private string productID;

    private string totalAmount;

    private DateTime orderDate;

    private string address;

    private DateTime deliveredDate;

    private string deliveryStatus;

    public Order(string id, string sellerID, string customerID, string productID, string totalAmount, DateTime orderDate, string address, DateTime deliveredDate, string deliveryStatus)

    {

        this.id = id;

        this.sellerID = sellerID;

        this.customerID = customerID;

        this.productID = productID;

        this.totalAmount = totalAmount;

        this.orderDate = orderDate;

        this.address = address;

        this.deliveredDate = deliveredDate;

        this.deliveryStatus = deliveryStatus;

    }

    public void PlaceOrder()

    {

        // Implement logic to place an order

    }

}

class Payment

{

    private string id;

    private string orderID;

    private bool paid;

    private int total;

    private string details;

    public Payment(string id, string orderID, bool paid, int total, string details)

    {

        this.id = id;

        this.orderID = orderID;

        this.paid = paid;

        this.total = total;

        this.details = details;

    }

    public void SendOTP()

    {

        // Implement logic to send OTP for payment confirmation

    }

    public void ConfirmTransaction()

    {

        // Implement logic to confirm the payment transaction

    }

    public string GetPaymentDetails()

    {

        // Implement logic to retrieve and return payment details

        return $"Payment ID: {id}, Order ID: {orderID}, Total Amount: {total}";

    }

    public void MakeTransaction()

    {

        SendOTP();  // Send OTP for verification

        ConfirmTransaction(); // Confirm the payment transaction

        paid = true; // Mark the payment as paid

    }

}

class Program

{

    static void Main(string[] args)

    {

        // Example usage of the classes and their methods can be added here

    }

}