**Name : Nishi Sharma**

**Assignment 1:**

**Calculator using HTML, CSS and Javascript**

**File: first.html**

<!DOCTYPE html>

<html lang="en">

<head>

<!--<meta charset="UTF-8" />-->

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

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

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

<title>Calculator</title>

</head>

<body>

<div class="container">

<div class="calculator">

<input type="text" id="inputBox" placeholder="0" />

<div>

<button class="button operator">CE</button>

<button class="button operator">C</button>

<button class="button operator">%</button>

<button class="button operator">/</button>

</div>

<div>

<button class="button">7</button>

<button class="button">8</button>

<button class="button">9</button>

<button class="button operator">\*</button>

</div>

<div>

<button class="button">4</button>

<button class="button">5</button>

<button class="button">6</button>

<button class="button operator">-</button>

</div>

<div>

<button class="button">1</button>

<button class="button">2</button>

<button class="button">3</button>

<button class="button operator">+</button>

</div>

<div>

<button class="button">00</button>

<button class="button">0</button>

<button class="button">.</button>

<button class="button equalBtn">=</button>

</div>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

**CSS file: style.css**

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

body{

width: 100%;

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

background: linear-gradient(45deg, #e6cccc, #95b1d9);

}

.calculator{

border: 1px solid #134ec2;

padding: 20px;

border-radius: 20px;

background: transparent;

box-shadow: 0px 3px 15px rgba(166, 184, 221, 0.5);

}

input{

width: 320px;

border: none;

padding: 24px;

margin: 10px;

border-radius: 8%;

background: transparent;

box-shadow: -8px -8px 15px rgba(149, 146, 210, 0.906);

font-size: 45px;

text-align: right;

cursor: pointer;

color: #5039f9;

}

input::placeholder{

color: #5039f9;

}

button{

border: none;

width: 60px;

height: 60px;

margin: 10px;

border-radius: 50%;

background: transparent;

color: #4216de;

font-size: 25px;

box-shadow: -8px -8px 15px rgba(149, 146, 210, 0.906);

cursor: pointer;

}

.equalBtn{

background-color: #4b8dcf;

}

.operator{

color: #8d079f;

}

**JAVASCRIPT file: script.js**

let input = document.getElementById('inputBox');

let buttons = document.querySelectorAll('button');

let string = "";

let arr = Array.from(buttons);

arr.forEach(button => {

button.addEventListener('click', (e) =>{

if(e.target.innerHTML == '='){

string = eval(string);

input.value = string;

}

else if(e.target.innerHTML == 'CE'){

string = "";

input.value = string;

}

else if(e.target.innerHTML == 'C'){

string = string.substring(0, string.length-1);

input.value = string;

}

else{

string += e.target.innerHTML;

input.value = string;

}

}

}

**OUTPUT:**



**Assignment 2:**

**E commerce UML diagram into .net:**



using System;

namespace Project

{

public class user {

private bool role;

private string loginID;

private string password;

public user() { }

public user(bool role, string loginID, string password)

{

this.role = role;

this.loginID = loginID;

this.password = password;

}

public void verifyUser(string login\_user, string password\_user) {

if(loginID == login\_user && password== password\_user)

Console.WriteLine("Verified User");

}

}

public class seller : user

{

private string id;

private string address;

private string phone;

private string Email;

private int sellerRating;

Product prod = new Product();

public seller() { }

public seller(string id, string address, string phone, string email, int sellerRating, Product prod)

{

this.id = id;

this.address = address;

this.phone = phone;

this.Email = email;

this.sellerRating = sellerRating;

this.prod = prod;

}

public seller(bool role, string loginID, string password) : base(role, loginID, password)

{

}

public void login(string id\_emp)

{

if(id==id\_emp)

Console.WriteLine("Successfully Logged into Seller's account");

}

public void register()

{

Console.WriteLine("Successfully registered into seller's account");

}

public void updateProfile(string id, string address, string phone, string email, int sellerRating, Product prod)

{

this.id = id;

this.address = address;

this.phone = phone;

this.Email = email;

this.sellerRating = sellerRating;

this.prod = prod;

Console.WriteLine("Profile updated");

}

}

public class Customer : user {

ShoppingCart shop = new ShoppingCart();

Reviews review = new Reviews();

Orders order = new Orders();

private string id;

private string address;

private string phone;

private string Email;

public Customer() { }

public Customer(bool role, string loginID, string password) : base(role, loginID, password)

{

}

public Customer(ShoppingCart shop, Reviews review, Orders order, string id, string address, string phone, string email)

{

this.shop = shop;

this.review = review;

this.order = order;

this.id = id;

this.address = address;

this.phone = phone;

this.Email = email;

}

public void login(string id\_emp , string email\_emp)

{

if(id==id\_emp && Email==email\_emp)

Console.WriteLine("Succesfully Logged in to Customer's account");

}

public void register(string id\_emp, string email\_emp, string address)

{

this.id=id\_emp; ;

this.Email=email\_emp;

this.address=address;

Console.WriteLine("Successfully Registered to Customer's account");

}

public void updateProfile(string id\_emp, string email\_emp, string address)

{

this.id = id\_emp; ;

this.Email = email\_emp;

this.address = address;

Console.WriteLine("Successfully updated Customer ");

}

}

public class ShoppingCart

{

private Product prod;

private Date created;

public ShoppingCart (Date d, Product prod)

{

created = d;

this.prod = prod;

}

public void addCartItem(Product prod){

Console.WriteLine("The item had been successfully added to cart");

}

public void checkOut(Product prod){

Console.WriteLine("Items cart checked out");

}

public void viewCartDetails(Product prod, Date d) {

Console.WriteLine("Viewed Cart Item");

}

public void updateQuantity(Product prod)

{

Console.WriteLine("Quantity of cart item has been updated");

}

}

public class Reviews{

private string reviewID;

private string customerID;

private string parentID;

private string productID;

private string reviewContent;

private int rating;

public Reviews(){}

public Reviews(string reviewID, string customerID, string parentID, string productID, string reviewContent, int rating)

{

this.reviewID = reviewID;

this.customerID = customerID;

this.parentID = parentID;

this.productID = productID;

this.reviewContent = reviewContent;

this.rating = rating;

}

public void addReview(string review\_user)

{

this.reviewContent = review\_user;

Console.WriteLine("Your review successfully added");

}

public void deleteReview(string delete\_review)

{

this.reviewContent = delete\_review;

Console.WriteLine("Your review successfully deleted");

}

public void editReview(string edit\_review)

{

this.reviewContent = edit\_review;

Console.WriteLine("The review has been edited");

}

}

public class Orders

{

private string id;

private string sellerID;

private string customerID;

private string productID;

private string totalAmount;

private Date orderDate;

private string address;

private Date deliveryDate;

private string DeliveryStatus;

Payment pay = new Payment();

public Orders() { }

public Orders(string id, string sellerID, string customerID, string productID, string totalAmount, Date orderDate, string address, Date deliveryDate, string deliveryStatus, Payment pay)

{

this.id = id;

this.sellerID = sellerID;

this.customerID = customerID;

this.productID = productID;

this.totalAmount = totalAmount;

this.orderDate = orderDate;

this.address = address;

this.deliveryDate = deliveryDate;

this.DeliveryStatus = deliveryStatus;

this.pay = pay;

}

public void placeOrder(String prod, string amt, Date delvry\_d)

{

this.productID = prod;

this.totalAmount = amt;

this.orderDate = delvry\_d;

Console.WriteLine("Your order has been placed");

}

}

public class Payment

{

private string id;

private string orderID;

private bool paid;

private int total;

private string details;

public Payment() { }

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

{

this.id = id;

this.orderID = orderID;

this.paid = paid;

this.total = total;

this.details = details;

}

public void sendOtp()

{

Console.WriteLine("Sending Otp");

}

public void confirmTransaction()

{

Console.WriteLine("Confirmed Transcation");

}

public void getPaymentDetails()

{

Console.WriteLine("Payment details received");

}

public void makeTrasnsaction()

{

Console.WriteLine("Transaction has been made");

}

}

public class Product

{

private string productID;

private string productName;

private int productCost;

private string sellerID;

private DateTime postedDate;

Reviews reviews = new Reviews();

Orders orders= new Orders();

public Product(string productID, string productName, int productCost, string sellerID, DateTime postedDate, Reviews reviews, Orders orders)

{

this.productID = productID;

this.productName = productName;

this.productCost = productCost;

this.sellerID = sellerID;

this.postedDate = postedDate;

this.reviews = reviews;

this.orders = orders;

public void addToCart(string id, string name)

{

this.productID = id;

this.productName = name;

Console.WriteLine("Product has been added to cart");

}

public void sellProduct(int cost, string id, string name)

{

this.sellerID = id;

this.productName = name;

this.productCost = cost;

Console.WriteLine("Product has been sold");

}

public void getProductDetails(string id, string name)

{

Console.WriteLine("Get the product details");

}

public void buyProducts(string id, string name)

{

Console.WriteLine("Get the products to buy");

}

}

}