**1. HTML Files**

**main.html**

This page will have a form to enter a phone number and a button to validate it.

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Main Page</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f4f4f4;

}

.container {

text-align: center;

}

input {

padding: 10px;

font-size: 16px;

}

button {

padding: 10px 20px;

font-size: 16px;

margin-top: 10px;

}

</style>

</head>

<body>

<div class="container">

<h1>Enter Phone Number</h1>

<input type="text" id="phoneNumber" placeholder="Enter your phone number">

<button onclick="validateNumber()">Validate</button>

<p id="error" style="color: red; display: none;">Please enter a 10-digit phone number.</p>

</div>

<script>

function validateNumber() {

const phoneNumber = document.getElementById('phoneNumber').value;

if (phoneNumber.length === 10 && !isNaN(phoneNumber)) {

window.location.href = 'menu.html';

} else {

document.getElementById('error').style.display = 'block';

}

}

</script>

</body>

</html>

**menu.html**

This page will show burgers with dropdowns for categories and quantity, and a button to add items to the cart.

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Menu</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #e9ecef;

margin: 0;

padding: 0;

}

.container {

width: 80%;

margin: 0 auto;

}

.burger {

border: 1px solid #ccc;

padding: 10px;

margin: 10px 0;

background-color: #fff;

}

.burger select, .burger input {

margin-right: 10px;

}

button {

padding: 10px 20px;

margin-top: 10px;

}

.cart-button {

margin-top: 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>Menu</h1>

<div class="burger">

<h2>Burger 1</h2>

<label>Category:

<select class="category">

<option value="veg">Veg</option>

<option value="egg">Egg</option>

<option value="chicken">Chicken</option>

</select>

</label>

<label>Quantity:

<input type="number" class="quantity" min="1" value="1">

</label>

<button onclick="addToCart('Burger 1')">Add to Cart</button>

</div>

<!-- Repeat the above div for each burger item (5 total) -->

<button class="cart-button" onclick="viewCart()">View Cart</button>

</div>

<script>

let cart = [];

function addToCart(burgerName) {

const category = document.querySelector(`.burger h2:contains('${burgerName}') ~ select`).value;

const quantity = parseInt(document.querySelector(`.burger h2:contains('${burgerName}') ~ input`).value);

const price = 100; // Set a fixed price for simplicity

cart.push({ burgerName, category, quantity, price, total: quantity \* price });

alert(`${burgerName} added to cart!`);

}

function viewCart() {

localStorage.setItem('cart', JSON.stringify(cart));

window.location.href = 'cart.html';

}

</script>

</body>

</html>

**cart.html**

This page will show the cart and allow users to place an order.

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Cart</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.container {

width: 80%;

margin: 0 auto;

padding: 20px;

}

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

table, th, td {

border: 1px solid #ccc;

}

th, td {

padding: 10px;

text-align: center;

}

button {

padding: 10px 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>Cart</h1>

<table id="cartTable">

<thead>

<tr>

<th>Item</th>

<th>Category</th>

<th>Quantity</th>

<th>Price</th>

<th>Total</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<button onclick="placeOrder()">Place Order</button>

</div>

<script>

function loadCart() {

const cart = JSON.parse(localStorage.getItem('cart')) || [];

const tableBody = document.querySelector('#cartTable tbody');

tableBody.innerHTML = '';

cart.forEach(item => {

const row = document.createElement('tr');

row.innerHTML = `

<td>${item.burgerName}</td>

<td>${item.category}</td>

<td>${item.quantity}</td>

<td>${item.price}</td>

<td>${item.total}</td>

`;

tableBody.appendChild(row);

});

}

function placeOrder() {

fetch('https://your-api-url/api/orders', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(JSON.parse(localStorage.getItem('cart')))

})

.then(response => response.json())

.then(data => {

alert('Order placed successfully!');

localStorage.removeItem('cart');

window.location.href = 'main.html';

})

.catch(error => {

console.error('Error:', error);

alert('Failed to place the order.');

});

}

loadCart();

</script>

</body>

</html>

**2. ASP.NET Core API**

1. **Create a new ASP.NET Core Web API project:**
   * Open Visual Studio.
   * Go to File -> New -> Project.
   * Select ASP.NET Core Web API and click Next.
   * Name your project and click Create.
2. **Add a model for the order:**

Create a new file Order.cs in the Models folder.

csharp

Copy code

namespace YourNamespace.Models

{

public class Order

{

public string BurgerName { get; set; }

public string Category { get; set; }

public int Quantity { get; set; }

public decimal Price { get; set; }

public decimal Total { get; set; }

}

}

1. **Set up the database context:**

Create a new file OrderContext.cs in the Data folder.

csharp

Copy code

using Microsoft.EntityFrameworkCore;

using YourNamespace.Models;

namespace YourNamespace.Data

{

public class OrderContext : DbContext

{

public OrderContext(DbContextOptions<OrderContext> options) : base(options) { }

public DbSet<Order> Orders { get; set; }

}

}

1. **Configure the API in Startup.cs or Program.cs:**

Add the following to configure services and middleware.

csharp

Copy code

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<OrderContext>(options =>

options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

services.AddControllers();

// Other service configurations

}

Add the connection string to appsettings.json:

json

Copy code

"ConnectionStrings": {

"DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=OrderDb;Trusted\_Connection=True;MultipleActiveResultSets=true"

}

1. **Create a controller for orders:**

Create a new file OrdersController.cs in the Controllers folder.

csharp

Copy code

using Microsoft.AspNetCore.Mvc;

using YourNamespace.Data;

using YourNamespace.Models;

[Route("api/[controller]")]

[ApiController]

public class OrdersController : ControllerBase

{

private readonly OrderContext \_context;

public OrdersController(OrderContext context)

{

\_context = context;

}

[HttpPost]

public async Task<IActionResult> PostOrder([FromBody] List<Order> orders)

{

if (orders == null || !orders.Any())

{

return BadRequest("No orders to process.");

}

await \_context.Orders.AddRangeAsync(orders);

await \_context.SaveChangesAsync();

return Ok(new { Message = "Orders placed successfully." });

}

}

1. **Migrate the database:**
   * Open the Package Manager Console (Tools -> NuGet Package Manager -> Package Manager Console).
   * Run the following commands:

powershell

Copy code

Add-Migration InitialCreate

Update-Database

**3. Testing and Deployment**

1. **Test the entire flow:**
   * Open main.html in your browser and enter a valid phone number to go to the menu.html.
   * Add items to the cart and view the cart in cart.html.
   * Click "Place Order" and verify that the order is correctly placed in your SQL Server database.
2. **Deploy the API:**
   * Publish your ASP.NET Core API to your chosen hosting environment.
3. **Update cart.html with the actual API URL:**
   * Replace https://your-api-url/api/orders with the actual URL where your API is hosted.

This setup should cover the core functionalities you've described. Feel free to reach out if you need more specific details or run into any issues!