**LAB # 11**

**Task # 01: Consume previous micro services.**

**Solution:**

**Controller Code:**

public class Cosuming\_MicroService\_Controller : Controller

{

public ActionResult Product()

{

return View();

}

public ActionResult Bank()

{

return View();

}

}

**Bank.cs.html:**

@{

ViewBag.Title = "Bank";

}

<h1>Bank Transaction</h1>

<div>

<h2>Transaction Details</h2>

<div>

<label for="transactionId">Transaction ID:</label>

<input type="text" id="transactionId" name="transactionId" />

</div>

<div>

<label for="accountNumber">Account Number:</label>

<input type="text" id="accountNumber" name="accountNumber" />

</div>

<div>

<label for="amount">Amount:</label>

<input type="text" id="amount" name="amount" />

</div>

<div>

<label for="transactionType">Transaction Type:</label>

<select id="transactionType" name="transactionType">

<option value="Withdrawal">Withdrawal</option>

<option value="Deposit">Deposit</option>

</select>

</div>

<div>

<button onclick="createTransaction()">Create</button>

<button onclick="getTransactionById()">Get By Id</button>

</div>

<div id="idTable" style="text-align:center;display:none" >

<h3>Get By Id</h3>

<table id="transactionTable" class="table">

<thead>

<tr><th>Transaction ID</th><th>Account Number</th>

<th>Amount</th><th>Transaction Date</th> th>Transaction Type</th>

</tr>

</thead>

<tbody id="transactionDetails">

</tbody>

</table>

</div>

<hr />

<table class="table">

<thead>

<tr>

<th>Transaction ID</th>

<th>Account Number</th>

<th>Amount</th>

<th>Transaction Date</th>

<th>Transaction Type</th>

</tr>

</thead>

<tbody id="tableBody">

</tbody>

</table>

</div>

<script>

async function getTransactions() {

try {

const response = await fetch("https://localhost:44351/api/Bank", {

method: "GET",

headers: {

"Content-Type": "application/json"

}

});

if (!response.ok) {

throw new Error("Failed to fetch transactions");

}

const transactions = await response.json();

console.log(transactions);

var tableBody = document.getElementById("tableBody");

tableBody.innerHTML = "";

transactions.forEach(transaction => {

tableBody.innerHTML += `

<tr>

<td>${transaction.transactionId}</td>

<td>${transaction.accountNumber}</td>

<td>${transaction.amount}</td>

<td>${transaction.transactionDate}</td>

<td>${transaction.transactionType}</td>

</tr>

`;

});

} catch (error) {

console.error(error);

}

}

document.addEventListener("DOMContentLoaded", getTransactions);

async function createTransaction() {

var accountNumber = document.getElementById("accountNumber").value;

var amount = document.getElementById("amount").value;

var transactionType = document.getElementById("transactionType").value;

var currentDate = new Date();

var formattedDate = currentDate.toLocaleDateString();

var formattedTime = currentDate.toLocaleTimeString();

var dateTimeString = formattedDate + ' ' + formattedTime;

var transaction = {

AccountNumber: accountNumber,

Amount: amount,

TransactionDate: dateTimeString,

TransactionType: transactionType

};

try {

const response = await fetch("https://localhost:44351/api/Bank", {

method: "POST",

headers: {

"Content-Type": "application/json"

},

body: JSON.stringify(transaction)

});

if (response.ok) {

console.log("Transaction created successfully");

getTransactions();

} else {

console.error("Failed to create transaction");

}

} catch (error) {

console.error(error);

}

}

async function getTransactionById() {

try {

const transactionId = document.getElementById("transactionId").value;

const response = await fetch(`https://localhost:44351/api/Bank/${transactionId}`, {

method: "GET",

headers: {

"Content-Type": "application/json"

}

});

if (!response.ok) {

throw new Error("Failed to fetch transaction");

}

const transaction = await response.json();

const transactionDetails = document.getElementById("transactionDetails");

transactionDetails.innerHTML = `

<tr>

<td>${transaction.transactionId}</td>

<td>${transaction.accountNumber}</td>

<td>${transaction.amount}</td>

<td>${transaction.transactionDate}</td>

<td>${transaction.transactionType}</td>

</tr>

`;

const idTable = document.getElementById("idTable");

idTable.style.display = "block";

} catch (error) {

console.error(error);

}

}

</script>

**Product.cs.html:**

@{

ViewBag.Title = "Product";

}

<h1>Product Management</h1>

<div>

<div>

<label for="productId">ID:</label>

<input type="text" id="productId" name="productId" />

</div>

<div>

<label for="productName">Name:</label>

<input type="text" id="productName" name="productName" />

</div>

<div>

<label for="productDescription">Description:</label>

<input type="text" id="productDescription" name="productDescription" />

</div>

<div>

<label for="productPrice">Price:</label>

<input type="text" id="productPrice" name="productPrice" />

</div>

<div>

<label for="categoryId">Category ID:</label>

<input type="text" id="categoryId" name="categoryId" />

</div>

<div>

<button onclick="createProduct()">Create</button>

<button onclick="updateProduct()">Update</button>

<button onclick="deleteProduct()">Delete</button>

<button onclick="getProductById()">Get By Id</button>

</div>

<div id="idTable" style=" display:none;">

<h3 style="text-align: center;">Get By Id</h3>

<table id="productTable" class="table">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Description</th>

<th>Price</th>

<th>Category ID</th>

</tr>

</thead>

<tbody id="productDetails">

</tbody>

</table>

</div>

<table class="table">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Description</th>

<th>Price</th>

<th>Category ID</th>

</tr>

</thead>

<tbody id="tableBody">

</tbody>

</table>

</div>

<script>

async function getProducts() {

try {

const response = await fetch("https://localhost:44343/api/Product", {

method: "GET",

headers: {

"Content-Type": "application/json"

}

});

if (!response.ok) {

throw new Error("Failed to fetch products");

}

const products = await response.json();

console.log(products)

var tableBody = document.getElementById("tableBody");

tableBody.innerHTML = "";

products.forEach(product => {

tableBody.innerHTML += `

<tr>

<td>${product.id}</td>

<td>${product.name}</td>

<td>${product.description}</td>

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

<td>${product.categoryId}</td>

</tr>

`;

});

} catch (error) {

console.error(error);

}

}

document.addEventListener("DOMContentLoaded", getProducts)

async function createProduct() {

var productId = document.getElementById("productId").value;

var productName = document.getElementById("productName").value;

var productDescription = document.getElementById("productDescription").value;

var productPrice = document.getElementById("productPrice").value;

var categorytId = document.getElementById("categoryId").value;

var product = {

Name: productName,

Description: productDescription,

Price: productPrice,

CategoryId: categorytId

};

try {

const response = await fetch("https://localhost:44343/api/Product", {

method: "POST",

headers: {

"Content-Type": "application/json"

},

body: JSON.stringify(product)

});

if (response.ok) {

console.log("Product created successfully");

getProducts();

} else {

console.error("Failed to create product");

}

} catch (error) {

console.error(error);

}

}

async function updateProduct() {

var productId = document.getElementById("productId").value;

var productName = document.getElementById("productName").value;

var productDescription = document.getElementById("productDescription").value;

var productPrice = document.getElementById("productPrice").value;

var categorytId = document.getElementById("categoryId").value;

var product = {

Id: productId,

Name: productName,

Description: productDescription,

Price: productPrice,

CategoryId: categorytId

};

try {

const response = await fetch(`https://localhost:44343/api/Product`, {

method: "PUT",

headers: {

"Content-Type": "application/json"

},

body: JSON.stringify(product)

});

if (response.ok) {

console.log("Product updated successfully");

getProducts();

} else {

console.error("Failed to update product");

}

} catch (error) {

console.error(error);

}

}

async function deleteProduct() {

var productId = document.getElementById("productId").value;

try {

const response = await fetch(`https://localhost:44343/api/Product/${productId}`, {

method: "DELETE"

});

if (response.ok) {

console.log("Product deleted successfully");

getProducts();

} else {

console.error("Failed to delete product");

}

} catch (error) {

console.error(error);

}

}

async function getProductById() {

var productId = document.getElementById("productId").value;

try {

const response = await fetch(`https://localhost:44343/api/Product/${productId}`, {

method: "GET",

headers: {

"Content-Type": "application/json"

}

});

if (!response.ok) {

throw new Error("Failed to fetch product by ID");

}

const product = await response.json();

console.log(product);

var productDetails = document.getElementById("productDetails");

productDetails.innerHTML = "";

productDetails.innerHTML += `

<tr>

<td>${product.id}</td>

<td>${product.name}</td>

<td>${product.description}</td>

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

<td>${product.categoryId}</td>

</tr>

`;

// Show the table

document.getElementById("idTable").style.display = "block";

} catch (error) {

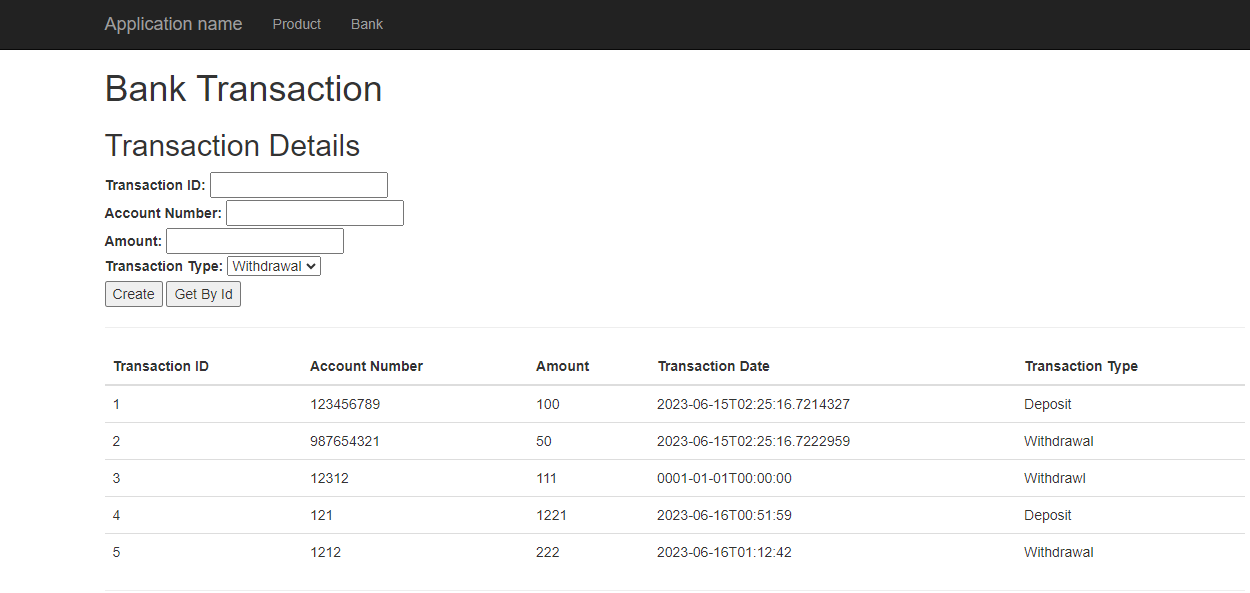
console.error(error);

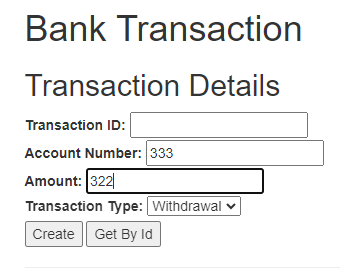
}

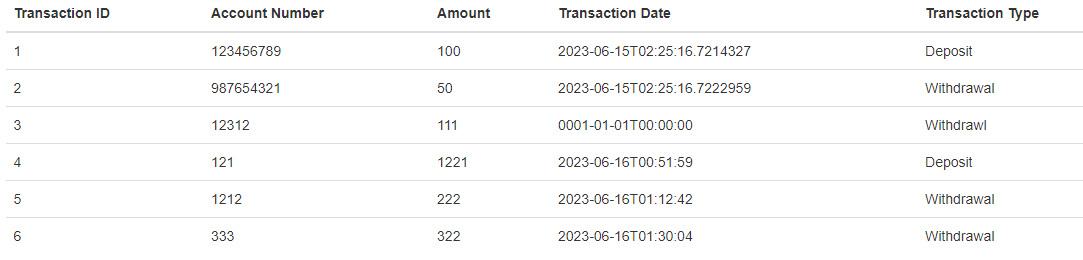
}

</script>

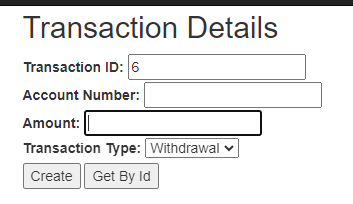
**Output:**

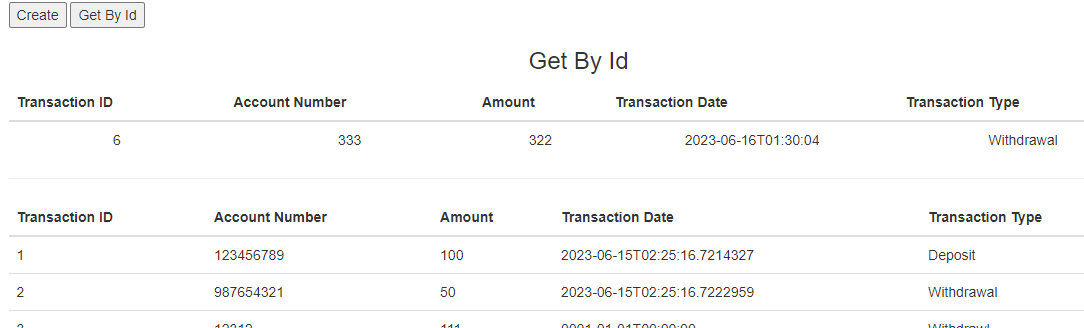




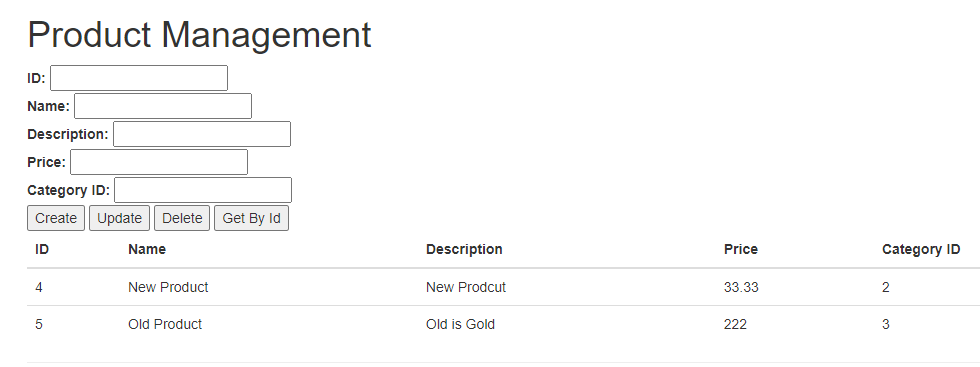


**Get By Id:**

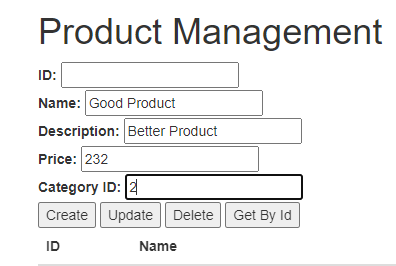




**Product:**

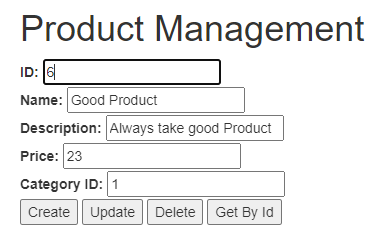


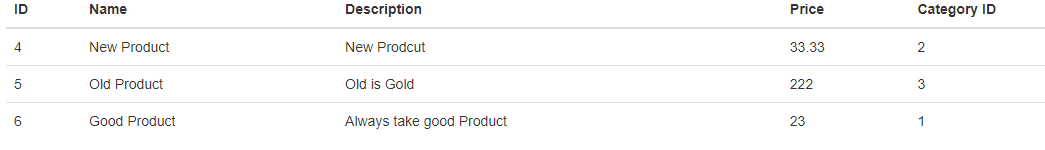
**Create:**



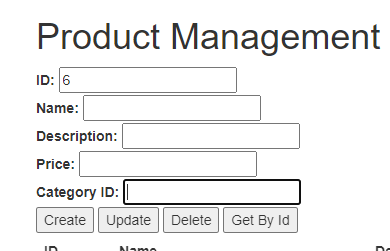


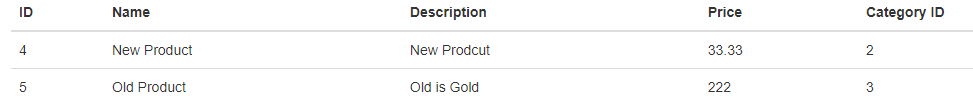
**Update:**





**Delete:**





**Get By Id:**

