PHASE PROJECT 4

Title: Pizza Portal

The Pizza Portal project aims to provide users with a seamless online platform for ordering pizzas. The system consists of several interconnected pages facilitating the selection of pizzas, customization of orders, payment processing, and order confirmation.

Project Workflow

Index Page:

The Index page displays a list of available pizzas fetched from the PizzaManager.

1. **Selecting Pizza:**
   This serves as the landing page displaying a comprehensive list of available pizzas sourced from the PizzaManager. Users can browse through the options and select their desired pizza, initiating the ordering process.
2. **Selected Items Page:**
   Upon selecting a pizza, users are redirected to this page where they can specify the quantity of the chosen pizza and provide essential delivery information such as address details.
3. **Payment Mode Page:**
   This page displays a summary of the order including the total price and the delivery address. Users can proceed to choose their preferred payment method for completing the transaction.
4. **Order Success Page:**
   Upon successful completion of the payment process, users are directed to this page which confirms the placement of their order and provides them with a unique order ID for reference.

Deployment

The project is deployed on Microsoft Azure using Azure App Service, ensuring accessibility and scalability. The deployment process involves the following steps:

a) **Creating an Azure Instance:** An instance is provisioned on Azure to host the application, providing a stable environment for its execution.

b) **Publishing from Visual Studio:** The application is published directly from Visual Studio, streamlining the deployment process and ensuring seamless integration with Azure services.

c) **Deployment Center:** The GitHub repository link is connected to the Deployment Center of the Azure App Service, enabling continuous deployment. This ensures that any updates or modifications pushed to the GitHub repository are automatically reflected in the deployed application, enhancing agility and efficiency in the development lifecycle.

SOURCE CODE

PizzaLibrary Namespace:

PizzaProperties Class:

```
namespace PizzaLibrary
{
   public class PizzaProperties
   {
      public int Id { get; set; }
      public string Type { get; set; }
      public float Price { get; set; }
   }
}
```

PizzaManager Class:

```
using System.Collections.Generic;

namespace PizzaLibrary
{
   public class PizzaManager
   {
      public List<PizzaProperties> pizzaList;
      public PizzaManager()
      {
         pizzaList = new List<PizzaProperties>();
      }
      public List<PizzaProperties> ListOfPizza()
      {
         PizzaProperties pizza1 = new PizzaProperties() {
            Id = 1,
            Type = "Chicken Delight Pizza",
            Price = 350.99f
         };
         PizzaProperties pizza2 = new PizzaProperties() {
            Id = 2,
  Type = "Margherita Pizza",
  Price = 169.90f
};
PizzaProperties pizza3 = new PizzaProperties()
{
  Id = 3,
  Type = "Chocolate and Banana Pizza",
  Price = 225.66f
};
PizzaProperties pizza4 = new PizzaProperties()
{
  Id = 4,
  Type = "Chicken Pepporoni Pizza",
  Price = 290.99f
};
PizzaProperties pizza5 = new PizzaProperties()
{
  Id = 5,
  Type = "Mushroom and Olive  Pizza",
  Price = 235.66f
};
PizzaProperties pizza6 = new PizzaProperties()
{
  Id = 6,
  Type = "Farm House Pizza",
  Price = 240.13f
};
PizzaProperties pizza7 = new PizzaProperties()
{
  Id = 7,
  Type = "Cheese and Paneer Pizza",
  Price = 275.90f
};
PizzaProperties pizza8= new PizzaProperties()
{
  Id = 8,
  Type = "Cheese and Corn Pizza",
  Price = 234.90f
};
pizzaList.Add(pizza1);
pizzaList.Add(pizza2);
pizzaList.Add(pizza3);
pizzaList.Add(pizza4);
pizzaList.Add(pizza5);
```

```
            pizzaList.Add(pizza6);
            pizzaList.Add(pizza7);
            pizzaList.Add(pizza8);

            return pizzaList;
        }
    }
}
```

PizzaPortalProject Namespace

Model

PizzaModel Class:

```
namespace PizzaPortalProject.Models
{
    public class PizzaModel
    {
        public int Id { get; set; }
        public string Type { get; set; }
        public float Price { get; set; }
    }
}
```

Controller

PizzaController:

```
using PizzaLibrary;
using PizzaPortalProject.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;

namespace PizzaPortalProject.Controllers
{
    public class PizzaController : Controller
    {
```

```
public PizzaManager pizzaManager;
public PizzaController()
{
   pizzaManager = new PizzaManager();
}
public ActionResult Index()
{
   List<PizzaModel> pizzaModelList = new List<PizzaModel>();
   List<PizzaProperties> pizzaList = pizzaManager.ListOfPizza();
   foreach (PizzaProperties pizza in pizzaList)
   {
      PizzaModel pModel = new PizzaModel()
      {
         Id = pizza.Id,
         Type = pizza.Type,
         Price = pizza.Price
      };
      pizzaModelList.Add(pModel);
   }
   return View(pizzaModelList);
}

public ActionResult SelectedItems(int id)
{
   List<PizzaProperties> pizzaList = pizzaManager.ListOfPizza();
   PizzaProperties pizzaItem = pizzaList.Find(p => p.Id == id);
   PizzaModel model = new PizzaModel()
   {
      Id = pizzaItem.Id,
      Type = pizzaItem.Type,
      Price = pizzaItem.Price
   };
   TempData["Price"] = pizzaItem.Price;
   TempData["PizzaType"] = pizzaItem.Type;
   TempData.Keep();

   return View(model);
}

[HttpPost]
public ActionResult SelectedItems(string deliveryAddress, int itemQuantity)
{
   string price = TempData["Price"].ToString();
   float totalPrice = float.Parse(price) * itemQuantity;
```

```
            TempData["TotalPrice"] = totalPrice;
            TempData["Address"] = deliveryAddress;
            TempData.Keep();

            return RedirectToAction("PaymentMode");
        }

        public ActionResult PaymentMode()
        {
            Random random = new Random();
            const string chars =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
            int length = 10;
            string randomString = new string(Enumerable.Repeat(chars, length).Select(s =>
s[random.Next(s.Length)]).ToArray());
            TempData["orderid"] = randomString;
            ViewBag.TotalPrice = Convert.ToSingle(TempData["TotalPrice"]);
            ViewBag.Address = TempData["Address"].ToString();
            return View();
        }

        public ActionResult OrderSuccess()
        {
            TempData["RandomOrderId"] = TempData["orderid"];
            return View("OrderSuccess");
        }

    }
}
```

View

Index.cshtml:

```
@model IEnumerable<PizzaPortalProject.Models.PizzaModel>

@{
    ViewBag.Title = "Pizza World!!🍕";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2 style="text-align:left; margin-bottom: 30px;">Pizza World!!🍕</h2>
```

```
<div class="container">
    <div class="row">
        @foreach (var item in Model)
        {
            <div class="col-md-4">
                <div class="card">
                    <div class="card-header" style="color:crimson">
                        @Html.DisplayFor(modelItem => item.Type)
                    </div>
                    <div class="card-body">
                        <p class="price" style="color:deepskyblue">Price: $@Html.DisplayFor(modelItem => item.Price)</p>
                        <p>
                            <a href="@Url.Action("SelectedItems", new { id = item.Id })" class="btn btn-primary">Select</a>
                        </p>
                    </div>
                </div>
            </div>
        }
    </div>
</div>

<style>
    .card {
        margin-bottom: 30px;
        border: 1px solid #ddd;
        border-radius: 10px;
        background-color:antiquewhite;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        transition: transform 0.3s;
    }

        .card:hover {
            transform: translateY(-5px);
        }

    .card-header {
        background-color:bisque;
        font-weight: bold;
        padding: 15px;
        border-bottom: 1px solid #ddd;
        border-radius: 10px 10px 0 0;
    }

    .card-body {
        padding: 15px;
    }

    .price {
        font-size: 18px;
        margin-bottom: 10px;
    }

    .btn-primary {
        background-color: #ff6347;
        border-color: #ff6347;
    }

        .btn-primary:hover {
            background-color: #ff4c32;
            border-color: #ff4c32;
        }
</style>
```

SelectedItems.cshtml:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title >@ViewBag.Title</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f8f9fa;
            padding: 20px;
            text-align: center;
        }
        h2 {
            color: #ff6347;
            font-size: 28px;
            margin-bottom: 20px;
        }

        form {
            max-width: 400px;
            margin: auto;
            background-color: #fff;
            border: 1px solid #ccc;
            border-radius: 10px;
            padding: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }

        input[type="number"],
        input[type="text"],
        input[type="submit"] {
            width: calc(100% - 20px);
            padding: 10px;
            margin-top: 10px;
            margin-bottom: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-sizing: border-box;
        }

        input[type="submit"] {
            background-color: #ff6347;
            color: #fff;
            cursor: pointer;
            transition: background-color 0.3s;
        }

            input[type="submit"]:hover {
                background-color: #ff4c32;
            }
    </style>
</head>
<body>
```

```
  <form method="post">

    <h2 style="margin-bottom: 30px;">Selected Items 🍕</h2>
    <label for="itemQuantity" style="color: #ff6347;">Enter the quantity of the
@TempData["PizzaType"]:</label>
    <input type="number" value="1" id="itemQuantity" name="itemQuantity" required>

    <label for="deliveryAddress" style="color: #ff6347;">Enter the Delivery Address 🚚:</label>
    <input type="text" id="deliveryAddress" name="deliveryAddress" required>

    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

PaymentMode.cshtml:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>@ViewBag.Title</title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f8f9fa;
      padding: 20px;
      text-align: center;
    }

    h2 {
      color: #007bff;
      font-size: 28px;
      margin-bottom: 20px;
    }

    .payment-info {
      background-color:whitesmoke;
```

```
            border: 1px solid #ccc;
            border-radius: 10px;
            padding: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            margin-bottom: 20px;
        }

            .payment-info p {
                margin-bottom: 10px;
                color: #333;
                font-size: 18px;
            }

        .btn-success {
            background-color: #28a745;
            border-color: #28a745;
            color: #fff;
            padding: 10px 20px;
            font-size: 16px;
            border-radius: 5px;
            cursor: pointer;
            transition: background-color 0.3s;
        }

            .btn-success:hover {
                background-color: #218838;
            }
    </style>
</head>
<body>
    <h2>Payment Status</h2>

    <div class="payment-info">
        <p>The Total Price of @TempData["PizzaType"] = Rs. @TempData["TotalPrice"]</p>
        <p>The order will be delivered at @TempData["Address"]</p>
    </div>

    <input type="button" class="btn btn-success" value="Order"
onclick="redirectToOrderSuccess()" />

    <script>
```

```
        function redirectToOrderSuccess() {
            window.location.href = '@Url.Action("OrderSuccess")';
        }
    </script>
</body>
</html>
```

OrderSuccess.cshtml:

```
@{
    ViewBag.Title = "OrderSuccess";
}

<h2>Order Success</h2>
<h1>
    Your order has been placed successfully with the order id as
@TempData["RandomOrderId"]!!
</h1>
```

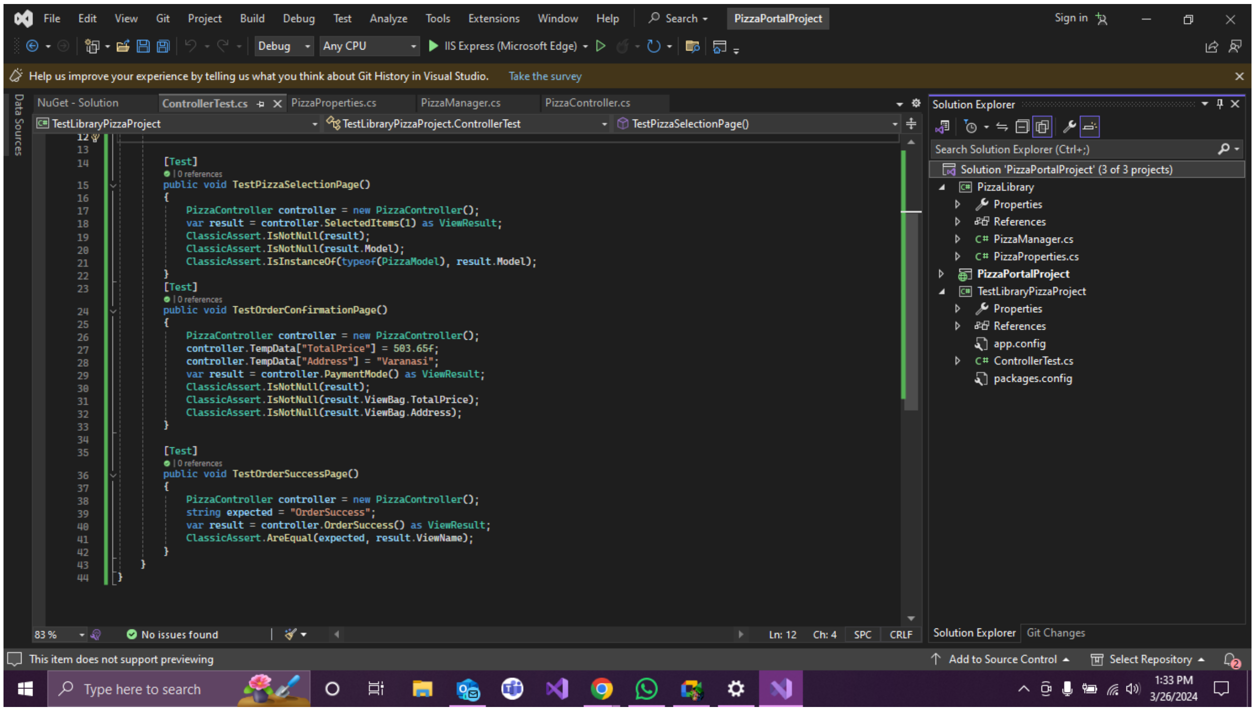
Tests

ControllerTest Class:

```
using NUnit.Framework;
using NUnit.Framework.Legacy;
using PizzaPortalProject.Controllers;
using PizzaPortalProject.Models;
using System.Web.Mvc;

namespace TestLibraryPizzaProject
{
    [TestFixture]
    public class ControllerTest
    {


        [Test]
        public void TestPizzaSelectionPage()
        {
            PizzaController controller = new PizzaController();
            var result = controller.SelectedItems(1) as ViewResult;
            ClassicAssert.IsNotNull(result);
            ClassicAssert.IsNotNull(result.Model);
            ClassicAssert.IsInstanceOf(typeof(PizzaModel), result.Model);
        }
        [Test]
        public void TestOrderConfirmationPage()
        {
            PizzaController controller = new PizzaController();
            controller.TempData["TotalPrice"] = 503.65f;
            controller.TempData["Address"] = "Varanasi";
            var result = controller.PaymentMode() as ViewResult;
            ClassicAssert.IsNotNull(result);
            ClassicAssert.IsNotNull(result.ViewBag.TotalPrice);
            ClassicAssert.IsNotNull(result.ViewBag.Address);
        }

        [Test]
        public void TestOrderSuccessPage()
        {
            PizzaController controller = new PizzaController();
            string expected = "OrderSuccess";
            var result = controller.OrderSuccess() as ViewResult;
            ClassicAssert.AreEqual(expected, result.ViewName);
        }
    }
}
```

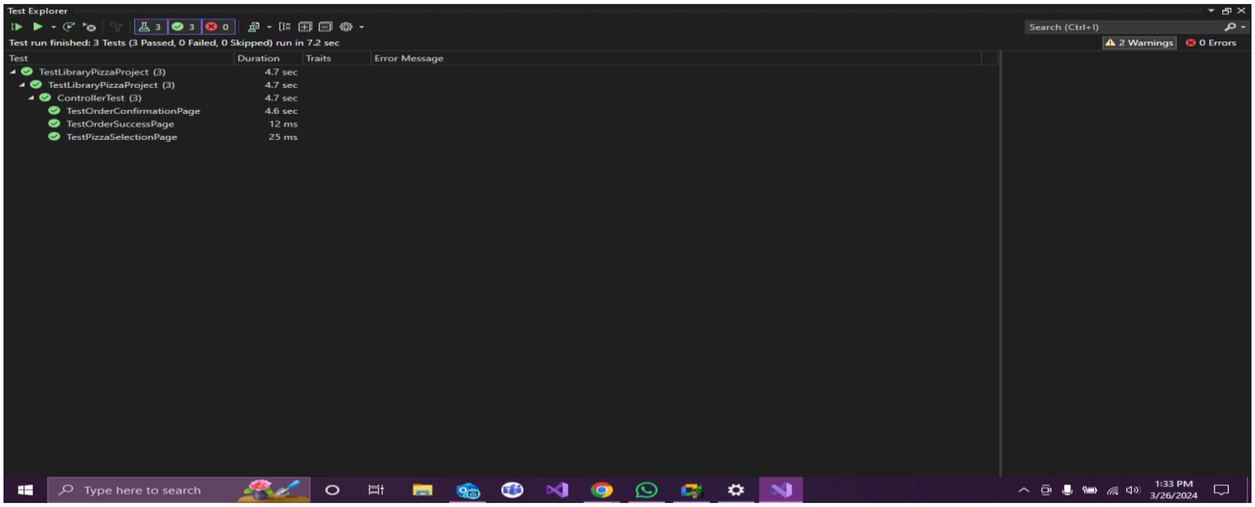
<u>OUTPUT SCREENSHOTS:</u>

Test Case:



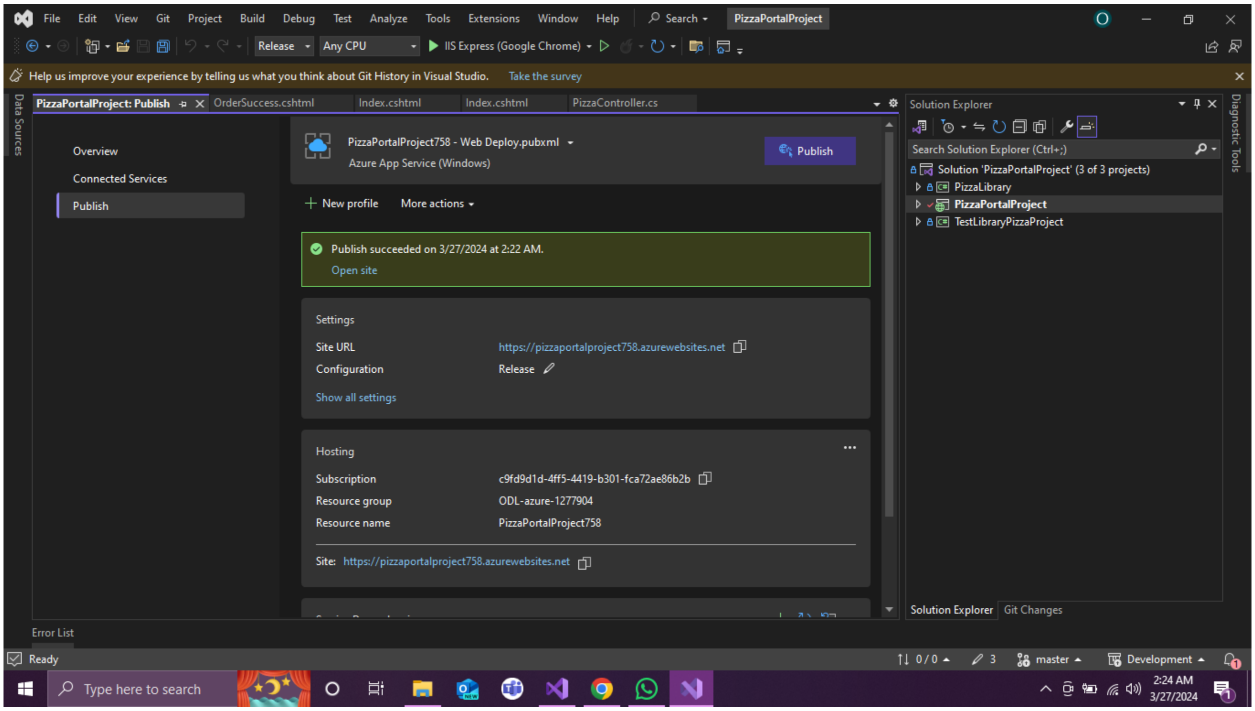
```
        [Test]
        public void TestPizzaSelectionPage()
        {
            PizzaController controller = new PizzaController();
            var result = controller.SelectedItems(1) as ViewResult;
            ClassicAssert.IsNotNull(result);
            ClassicAssert.IsNotNull(result.Model);
            ClassicAssert.IsInstanceOf(typeof(PizzaModel), result.Model);
        }
        [Test]
        public void TestOrderConfirmationPage()
        {
            PizzaController controller = new PizzaController();
            controller.TempData["TotalPrice"] = 503.65f;
            controller.TempData["Address"] = "Varanasi";
            var result = controller.PaymentMode() as ViewResult;
            ClassicAssert.IsNotNull(result);
            ClassicAssert.IsNotNull(result.ViewBag.TotalPrice);
            ClassicAssert.IsNotNull(result.ViewBag.Address);
        }

        [Test]
        public void TestOrderSuccessPage()
        {
            PizzaController controller = new PizzaController();
            string expected = "OrderSuccess";
            var result = controller.OrderSuccess() as ViewResult;
            ClassicAssert.AreEqual(expected, result.ViewName);
        }
    }
}
```

Test Case Results:



```
Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 7.2 sec
Test                                 Duration   Traits   Error Message
TestLibraryPizzaProject (3)           4.7 sec
  TestLibraryPizzaProject (3)         4.7 sec
    ControllerTest (3)                4.7 sec
      TestOrderConfirmationPage       4.6 sec
      TestOrderSuccessPage              12 ms
      TestPizzaSelectionPage            25 ms
```

Publish using Visual Studio



PizzaPortalProject758 - Web Deploy.pubxml

Azure App Service (Windows)

Publish succeeded on 3/27/2024 at 2:22 AM.

Open site

Settings

Site URL https://pizzaportalproject758.azurewebsites.net

Configuration Release

Show all settings

Hosting

Subscription c9fd9d1d-4ff5-4419-b301-fca72ae86b2b

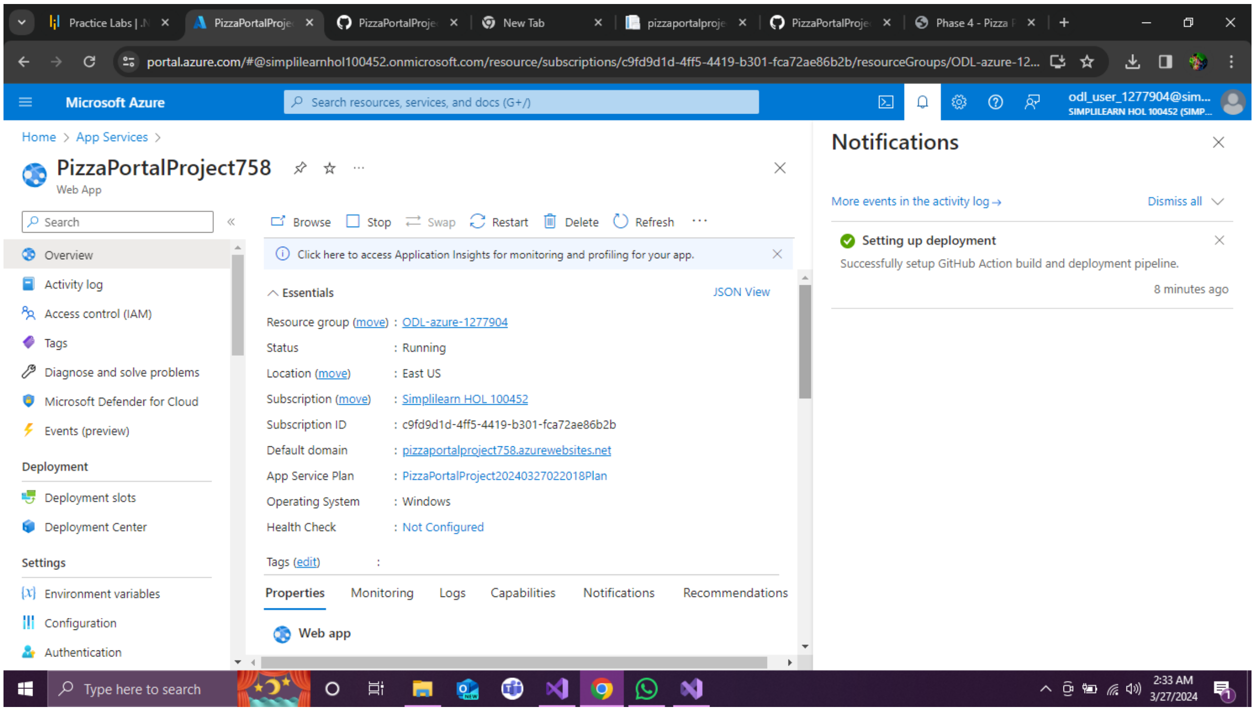
Resource group ODL-azure-1277904

Resource name PizzaPortalProject758

Site: https://pizzaportalproject758.azurewebsites.net

Solution 'PizzaPortalProject' (3 of 3 projects)
- PizzaLibrary
- PizzaPortalProject
- TestLibraryPizzaProject

Setting up Github Action with Azure



portal.azure.com/#@simplilearnhol100452.onmicrosoft.com/resource/subscriptions/c9fd9d1d-4ff5-4419-b301-fca72ae86b2b/resourceGroups/ODL-azure-12...

PizzaPortalProject758

Web App

Click here to access Application Insights for monitoring and profiling for your app.

Essentials

Resource group (move) : ODL-azure-1277904

Status : Running

Location (move) : East US

Subscription (move) : Simplilearn HOL 100452

Subscription ID : c9fd9d1d-4ff5-4419-b301-fca72ae86b2b

Default domain : pizzaportalproject758.azurewebsites.net

App Service Plan : PizzaPortalProject20240327022018Plan

Operating System : Windows

Health Check : Not Configured

Tags (edit) :

Notifications

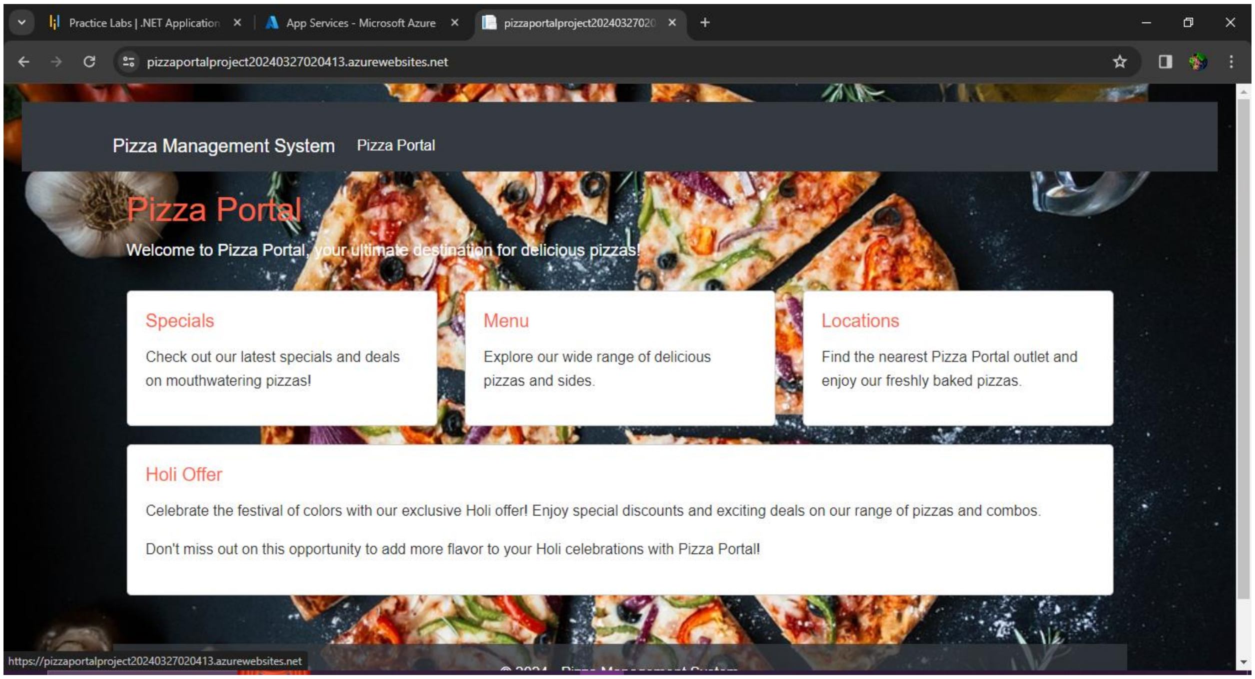
Setting up deployment

Successfully setup GitHub Action build and deployment pipeline.

8 minutes ago

Project View

Home page:


Practice Labs | .NET Application
App Services - Microsoft Azure
pizzaportalproject2024032702
pizzaportalproject20240327020413.azurewebsites.net
Pizza Management System
Pizza Portal
https://pizzaportalproject20240327020413.azurewebsites.net

Pizza Portal

Welcome to Pizza Portal, your ultimate destination for delicious pizzas!

Specials

Check out our latest specials and deals on mouthwatering pizzas!

Menu

Explore our wide range of delicious pizzas and sides.

Locations

Find the nearest Pizza Portal outlet and enjoy our freshly baked pizzas.

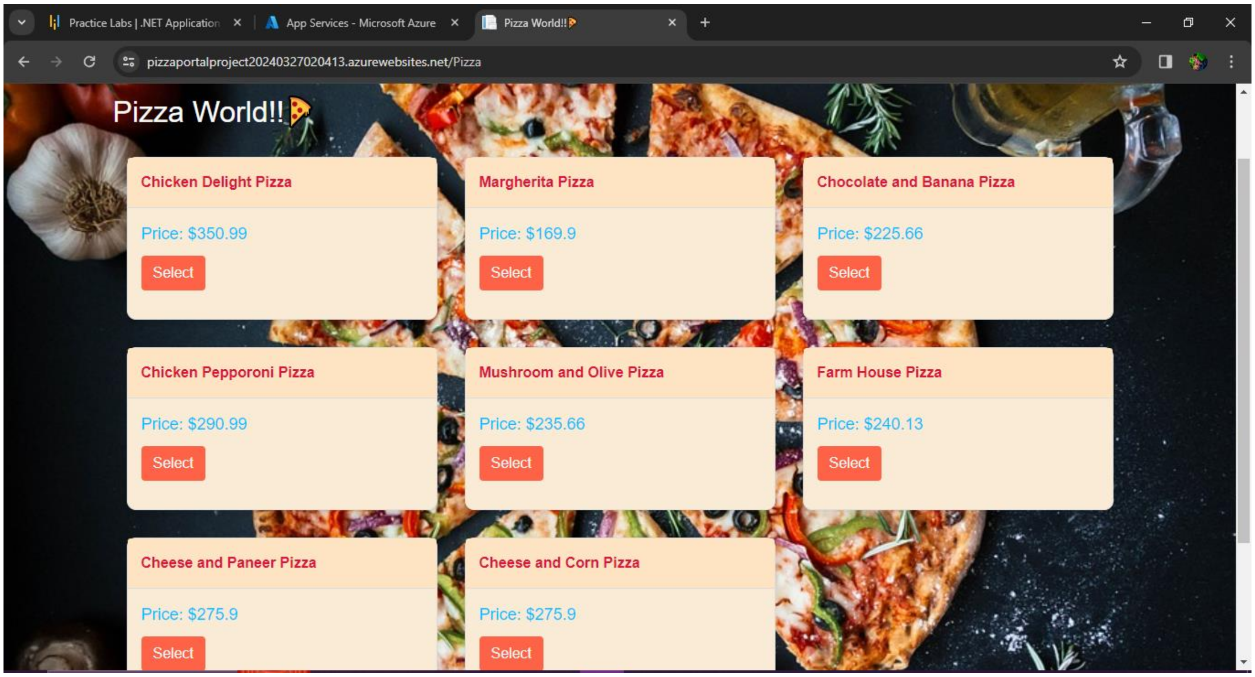
Holi Offer

Celebrate the festival of colors with our exclusive Holi offer! Enjoy special discounts and exciting deals on our range of pizzas and combos.

Don't miss out on this opportunity to add more flavor to your Holi celebrations with Pizza Portal!

© 2024 - Pizza Management System

Index Page:


Practice Labs | .NET Application
App Services - Microsoft Azure
Pizza World!!
pizzaportalproject20240327020413.azurewebsites.net/Pizza

Pizza World!! 🍕

Chicken Delight Pizza

Price: $350.99

Select

Margherita Pizza

Price: $169.9

Select

Chocolate and Banana Pizza

Price: $225.66

Select

Chicken Pepporoni Pizza

Price: $290.99

Select

Mushroom and Olive Pizza

Price: $235.66

Select

Farm House Pizza

Price: $240.13

Select

Cheese and Paneer Pizza

Price: $275.9

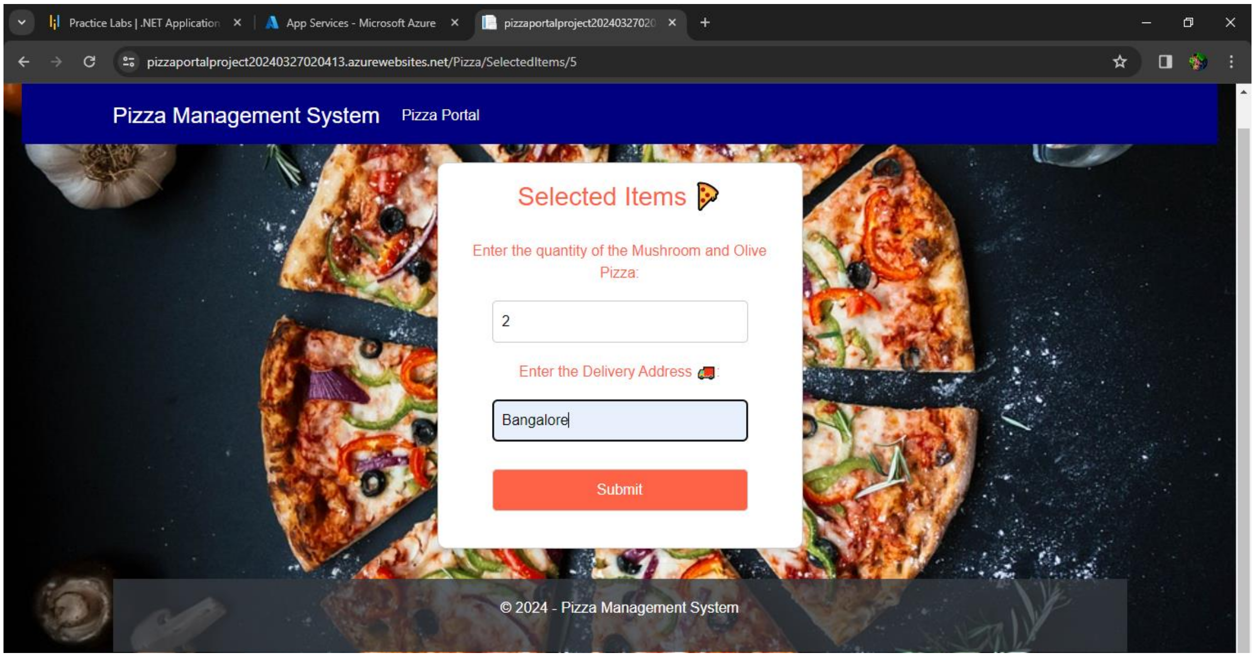
Select

Cheese and Corn Pizza

Price: $275.9

Select

Selected Items Order Page:



Practice Labs | .NET Application
App Services - Microsoft Azure
pizzaportalproject2024032702

pizzaportalproject20240327020413.azurewebsites.net/Pizza/SelectedItems/5

Pizza Management System Pizza Portal

Selected Items 🍕

Enter the quantity of the Mushroom and Olive Pizza:

2

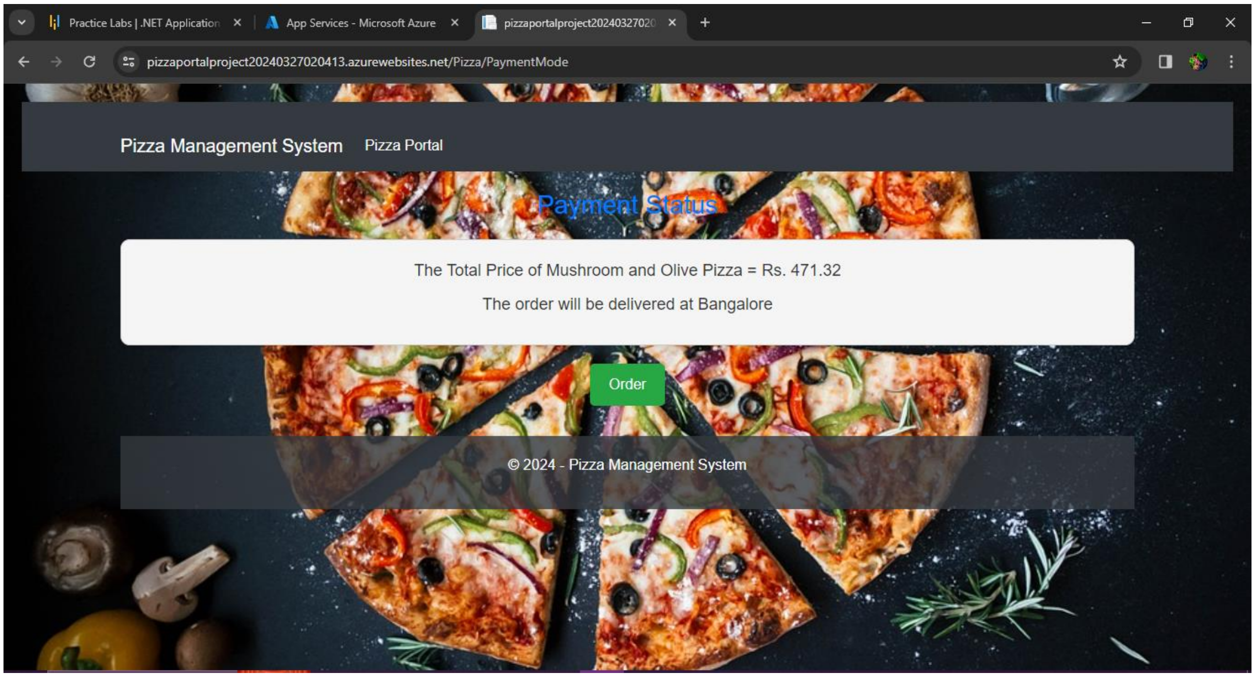
Enter the Delivery Address 🚚:

Bangalore

Submit

© 2024 - Pizza Management System

Payment Page:



Practice Labs | .NET Application
App Services - Microsoft Azure
pizzaportalproject2024032702

pizzaportalproject20240327020413.azurewebsites.net/Pizza/PaymentMode

Pizza Management System Pizza Portal

Payment Status

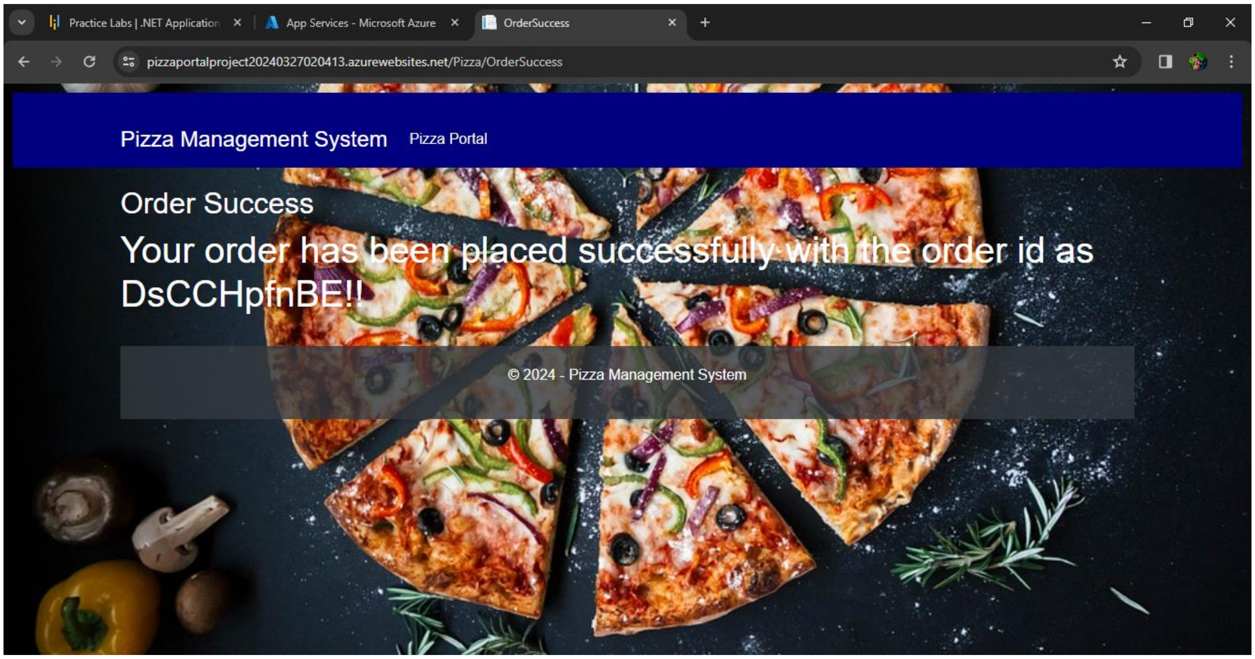
The Total Price of Mushroom and Olive Pizza = Rs. 471.32

The order will be delivered at Bangalore

Order

© 2024 - Pizza Management System

OrderSuccess Page:



Practice Labs | .NET Application
App Services - Microsoft Azure
OrderSuccess
pizzaportalproject20240327020413.azurewebsites.net/Pizza/OrderSuccess
Pizza Management System
Pizza Portal

Order Success

Your order has been placed successfully with the order id as DsCCHpfnBE!!

© 2024 - Pizza Management System