Pizza Portal

Introduction

The Pizza Portal Project is a web application aimed at managing pizza orders. It allows users to view available pizza options, select pizzas, specify quantities, provide delivery addresses, and make payments. The project is built using ASP.NET MVC and C# programming languages.

Project Structure

PizzaLibrary Namespace

PizzaProperties Class:

- Defines the properties of a pizza including its ID, type, and price.
- **CODE:**

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

PizzaManager Class:

- Manages a list of pizzas. It provides functionality to retrieve a list of available pizzas.
- **CODE:**

```
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 = "Pepperoni Pizza", Price = 250.99f };
            PizzaProperties pizza2 = new PizzaProperties() {
Id = 2, Type = "Margherita Pizza", Price = 200.50f };
            PizzaProperties pizza3 = new PizzaProperties()
            {
                Id = 3,
                Type = "Veggie Delight Pizza",
                Price = 325.66f
            };
            PizzaProperties pizza4 = new PizzaProperties()
            {
                Id = 4,
                Type = "Mushroom and Olive Pizza",
                Price = 440.99f
            };
            PizzaProperties pizza5 = new PizzaProperties()
            {
                Id = 5,
                Type = "Veggie Delight Pizza",
                Price = 325.66f
            };
            PizzaProperties pizza6 = new PizzaProperties()
            {
                Id = 6,
                Type = "Paneer Pizza",
                Price = 230.33f
            };
            PizzaProperties pizza7 = new PizzaProperties()
            {
                Id = 7,
                Type = "Cheese Pizza",
                Price = 375.00f
            };
            pizzaList.Add(pizza1);
            pizzaList.Add(pizza2);
            pizzaList.Add(pizza3);
            pizzaList.Add(pizza4);
            pizzaList.Add(pizza5);
            pizzaList.Add(pizza6);
            pizzaList.Add(pizza7);

            return pizzaList;
        }
    }
}
```

PizzaPortalProject Namespace

PizzaModel Class:

- Represents a pizza in the view. It includes properties like ID, type, and price.
- **CODE:**

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

PizzaController Class:

- Controller responsible for handling pizza-related actions such as displaying available pizzas, selecting items, processing orders, and managing payment.
- **CODE:**

```
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");
        }

    }
}
```

Views

Index.cshtml:

- Displays a list of available pizzas.
- **CODE:**

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

@{
    ViewBag.Title = "Index";
}

<h2>Pizza World!!🍕</h2>


<style>
    .card {
        margin: 50px;
        border: 1px solid #ccc;
        border-radius: 5px;
        height: 100px;
        background-color: pink;
        display: inline-block
    }

    .card-header {
        background-color: #f0f0f0;
        font-weight: bold;
        padding: 10px;
    }

    .card-body {
        padding: 15px;
    }
</style>

@foreach (var item in Model)
{
    <div class="card">
        <div class="card-header">
            @Html.DisplayFor(modelItem => item.Type)
        </div>
        <div class="card-body">
            <p>Price: @Html.DisplayFor(modelItem =>
item.Price)</p>
            <p>@Html.ActionLink("Details", "SelectedItems", new {
id = item.Id })</p>
        </div>
    </div>
}
```

SelectedItems.cshtml:

- Allows users to select pizza items and specify delivery addresses.
- **CODE:**

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

<style>
    form {
        width: 100%;
        border: 2px solid lightblue;
        background-color: aquamarine;
        text-align: center;
        margin: auto;
        padding: 10px;
    }
</style>

<h2>Selected Items</h2>

<form method="post">
    <br />
    Enter the quantity of the @TempData["PizzaType"]:
    <input type="number" name="itemQuantity" /> <br /><br />
    Enter the Delivery Address:
    <input type="text" name="deliveryAddress" />
    <br /><br />
    <input type="submit" />
</form>
```

PaymentMode.cshtml:

- Displays payment-related information such as total price and delivery address.
- **CODE:**

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

```
<h2>Payment Status</h2>
The Total Price of @TempData["PizzaType"] = Rs.
@TempData["TotalPrice"]
<br />
The order will be delivered at @TempData["Address"]

<p>
    @Html.ActionLink("Order", "OrderSuccess")
</p>
```

OrderSuccess.cshtml:

- Confirms successful order placement and displays the order ID.
- **CODE:**

```
@{
    ViewBag.Title = "OrderSuccess";
}
<h2>Order Success</h2>
<h1>
    Your order has been placed successfully with the order id as
@TempData["RandomOrderId"]!!
    Hope to see you soon..!!
</h1>
```

Tests

ControllerTest Class:

- Contains NUnit tests to ensure proper functionality of controller actions.
- **CODE:**

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

namespace TestLibraryForPizzaProject
{
    [TestFixture]
    public class ControllerTest
    {
        [Test]
        public void IndexActionReturnsIndexView()
        {
            string expected = "Index";
            HomeController controller = new HomeController();
            var result = controller.Index() as ViewResult;
            ClassicAssert.AreEqual(expected, result.ViewName);
        }

        [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);
        }
    }
}
```

Project Workflow

Index Page:

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

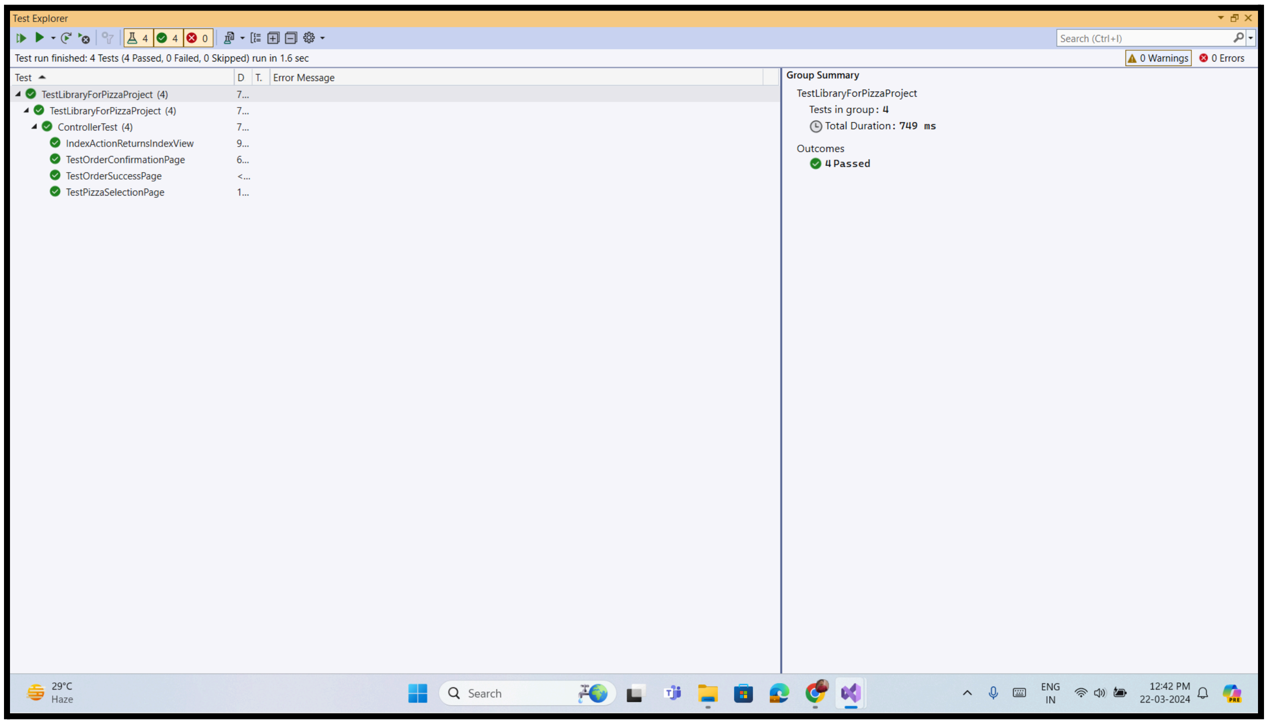
1. **Selecting Pizza:**
   Users can select a pizza from the list, which redirects them to the Selected Items page.
2. **Selected Items Page:**
   Users specify the quantity of the selected pizza and provide a delivery address.
3. **Payment Mode Page:**
   Displays the total price and delivery address. Users can proceed to payment.
4. **Order Success Page:**
   Confirms successful order placement and displays the order ID.

Deployment

1. The project is deployed on Azure using Azure App Service. The deployment process involves:
   a. Creating an Azure Instance: An instance is created on Azure to host the application.
   b. Publishing from Visual Studio: The application is published from Visual Studio to the Azure instance.
   c. Deployment Center: The GitHub repository link is attached to the Deployment Center of the Azure App Service to enable continuous deployment.

OUTPUT Screenshot:

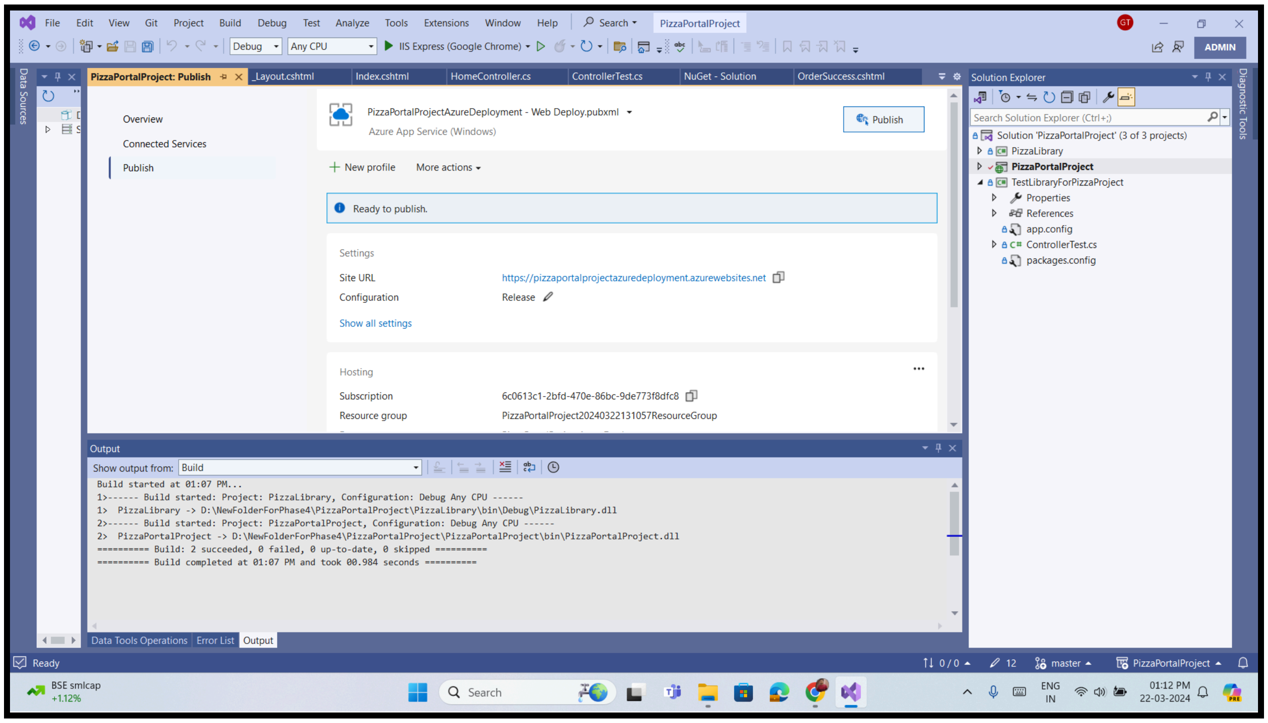
Test Case Results:



```
Test Explorer
4  4  0                                                        Search (Ctrl+I)
Test run finished: 4 Tests (4 Passed, 0 Failed, 0 Skipped) run in 1.6 sec      0 Warnings   0 Errors
Test                                    D     T.  Error Message        Group Summary
◢ TestLibraryForPizzaProject (4)        7...                           TestLibraryForPizzaProject
  ◢ TestLibraryForPizzaProject (4)      7...                             Tests in group: 4
    ◢ ControllerTest (4)                7...                             Total Duration: 749 ms
        IndexActionReturnsIndexView     9...
        TestOrderConfirmationPage       6...                           Outcomes
        TestOrderSuccessPage            <...                             4 Passed
        TestPizzaSelectionPage          1...

29°C   Search     ENG   12:42 PM
Haze              IN    22-03-2024
```

Publish using Visual Studio



PizzaPortalProjectAzureDeployment - Web Deploy.pubxml

Azure App Service (Windows)

Ready to publish.

Settings

Site URL https://pizzaportalprojectazuredeployment.azurewebsites.net

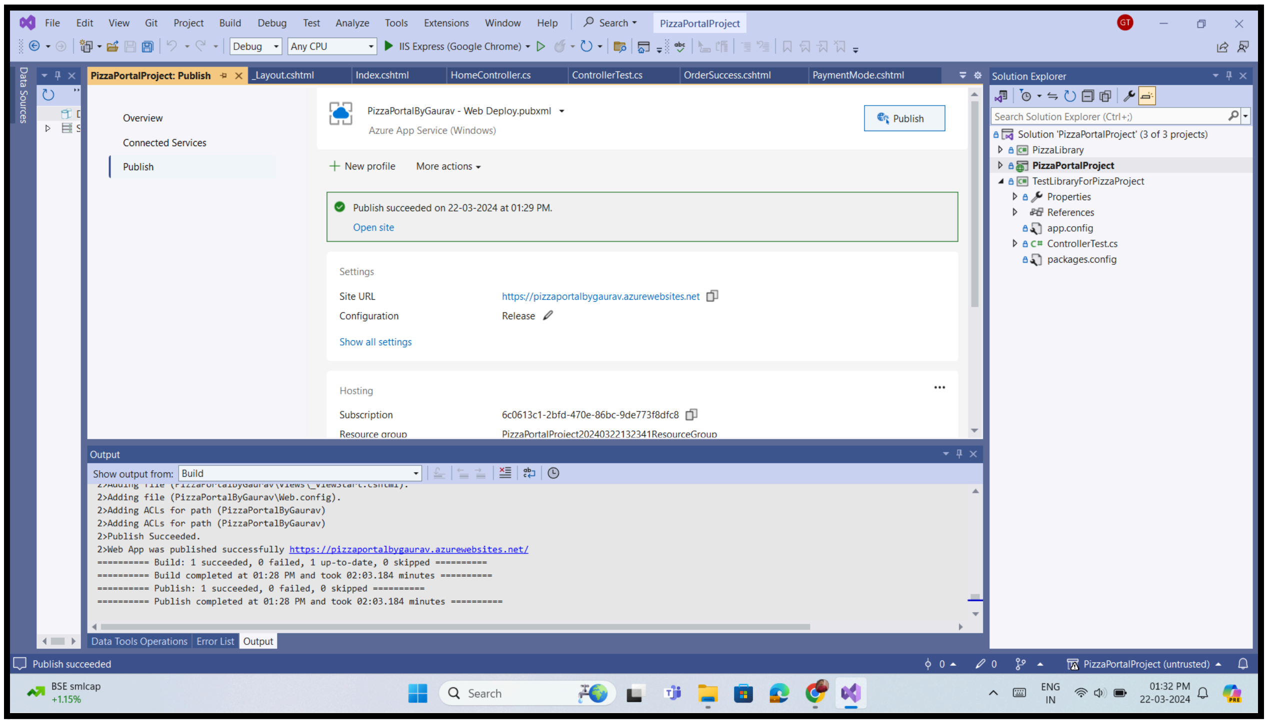
Configuration Release

Hosting

Subscription 6c0613c1-2bfd-470e-86bc-9de773f8dfc8

Resource group PizzaPortalProject20240322131057ResourceGroup

```
Build started at 01:07 PM...
1>------ Build started: Project: PizzaLibrary, Configuration: Debug Any CPU ------
1>  PizzaLibrary -> D:\NewFolderForPhase4\PizzaPortalProject\PizzaLibrary\bin\Debug\PizzaLibrary.dll
2>------ Build started: Project: PizzaPortalProject, Configuration: Debug Any CPU ------
2>  PizzaPortalProject -> D:\NewFolderForPhase4\PizzaPortalProject\PizzaPortalProject\bin\PizzaPortalProject.dll
========== Build: 2 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
========== Build completed at 01:07 PM and took 00.984 seconds ==========
```



PizzaPortalByGaurav - Web Deploy.pubxml

Azure App Service (Windows)

Publish succeeded on 22-03-2024 at 01:29 PM.

Open site

Settings

Site URL https://pizzaportalbygaurav.azurewebsites.net

Configuration Release

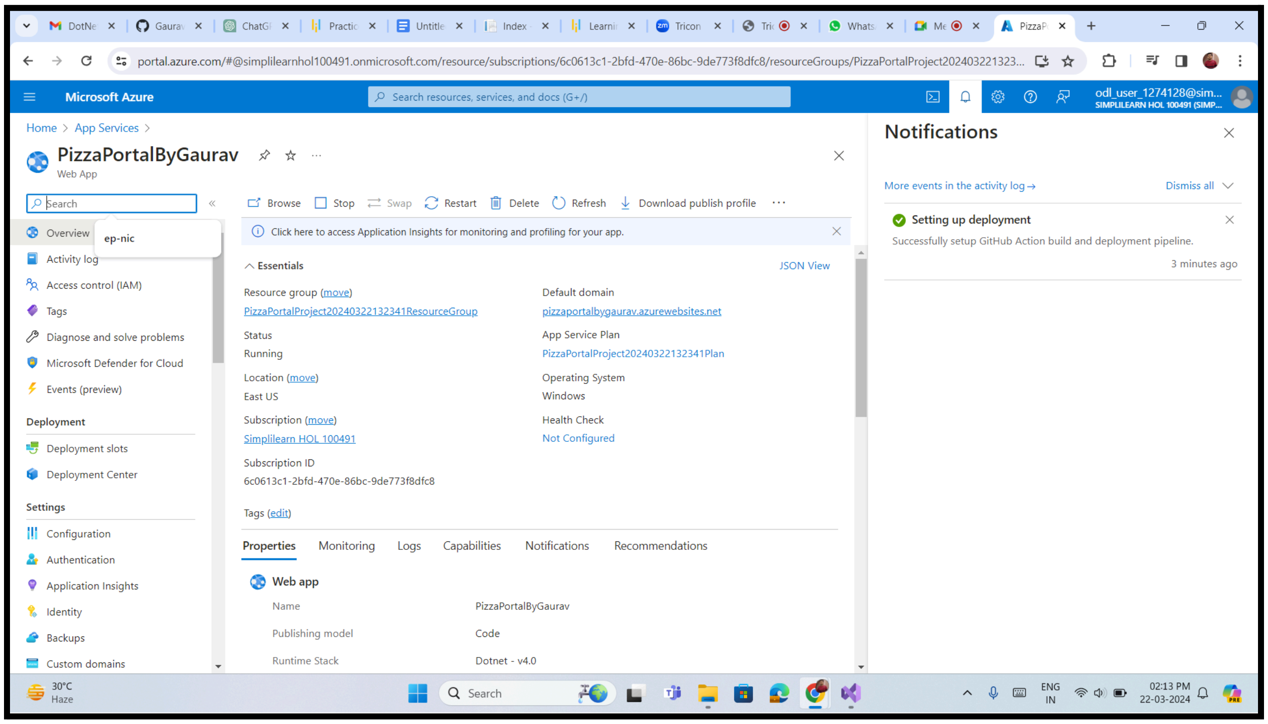
Hosting

Subscription 6c0613c1-2bfd-470e-86bc-9de773f8dfc8

Resource group PizzaPortalProject20240322132341ResourceGroup

```
2>Adding file (PizzaPortalByGaurav\Web.config).
2>Adding ACLs for path (PizzaPortalByGaurav)
2>Adding ACLs for path (PizzaPortalByGaurav)
2>Publish Succeeded.
2>Web App was published successfully https://pizzaportalbygaurav.azurewebsites.net/
========== Build: 1 succeeded, 0 failed, 1 up-to-date, 0 skipped ==========
========== Build completed at 01:28 PM and took 02:03.184 minutes ==========
========== Publish: 1 succeeded, 0 failed, 0 skipped ==========
========== Publish completed at 01:28 PM and took 02:03.184 minutes ==========
```

Setting up Github Action with Azure



PizzaPortalByGaurav
Web App

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

Essentials

Resource group (move)	Default domain
PizzaPortalProject20240322132341ResourceGroup	pizzaportalbygaurav.azurewebsites.net
Status	App Service Plan
Running	PizzaPortalProject20240322132341Plan
Location (move)	Operating System
East US	Windows
Subscription (move)	Health Check
Simplilearn HOL 100491	Not Configured
Subscription ID	
6c0613c1-2bfd-470e-86bc-9de773f8dfc8	

Tags (edit)

Web app

Name	PizzaPortalByGaurav
Publishing model	Code
Runtime Stack	Dotnet - v4.0

Notifications

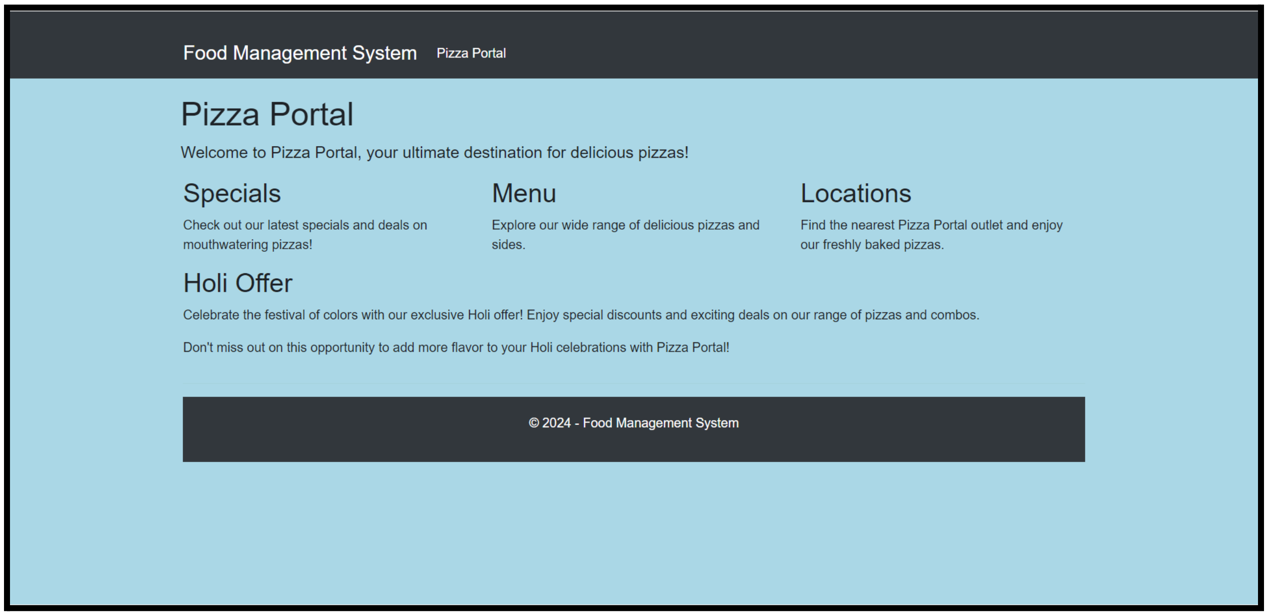
More events in the activity log → Dismiss all

Setting up deployment
Successfully setup GitHub Action build and deployment pipeline.
3 minutes ago

30°C Haze 02:13 PM 22-03-2024

Project View

Home Page



Food Management System Pizza Portal

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.

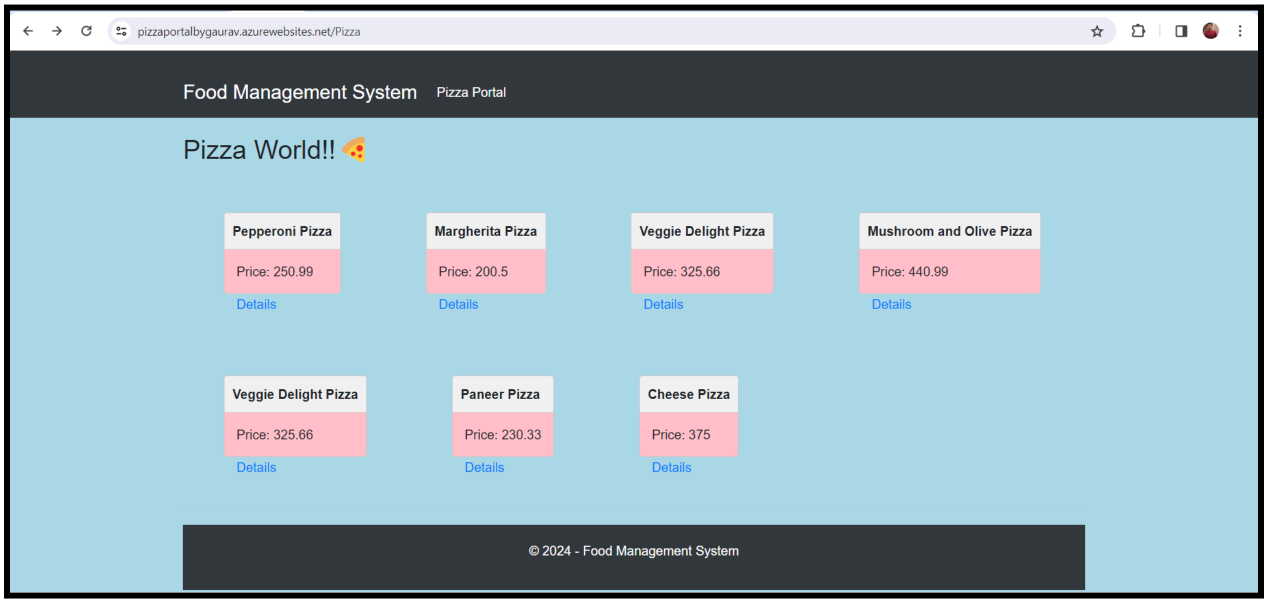
Holi Offer

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 - Food Management System

Index View



pizzaportalbygaurav.azurewebsites.net/Pizza

Food Management System Pizza Portal

Pizza World!! 🍕

Pepperoni Pizza
Price: 250.99
Details

Margherita Pizza
Price: 200.5
Details

Veggie Delight Pizza
Price: 325.66
Details

Mushroom and Olive Pizza
Price: 440.99
Details

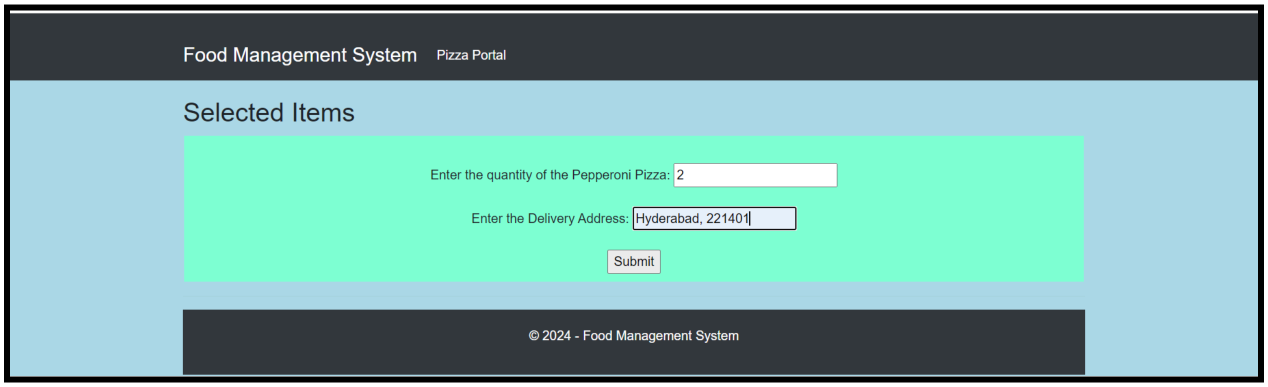
Veggie Delight Pizza
Price: 325.66
Details

Paneer Pizza
Price: 230.33
Details

Cheese Pizza
Price: 375
Details

© 2024 - Food Management System

Order View



Food Management System Pizza Portal

Selected Items

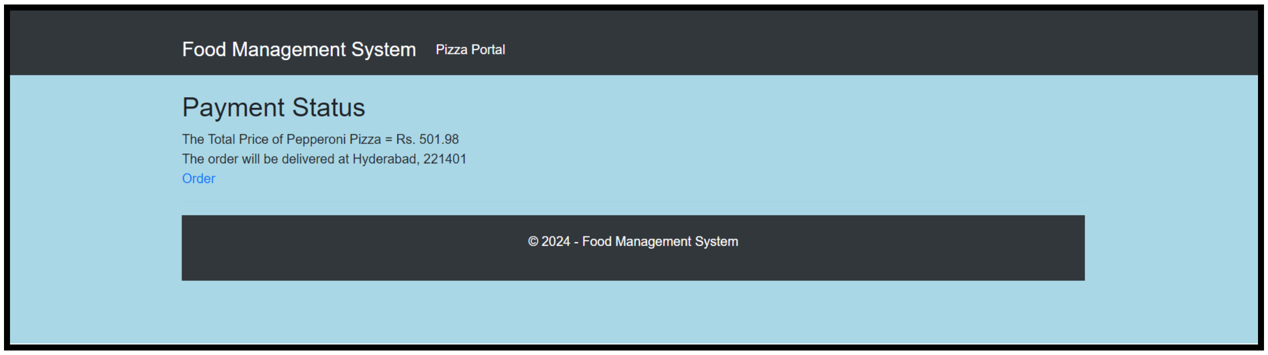
Enter the quantity of the Pepperoni Pizza: 2

Enter the Delivery Address: Hyderabad, 221401

Submit

© 2024 - Food Management System

Payment View:



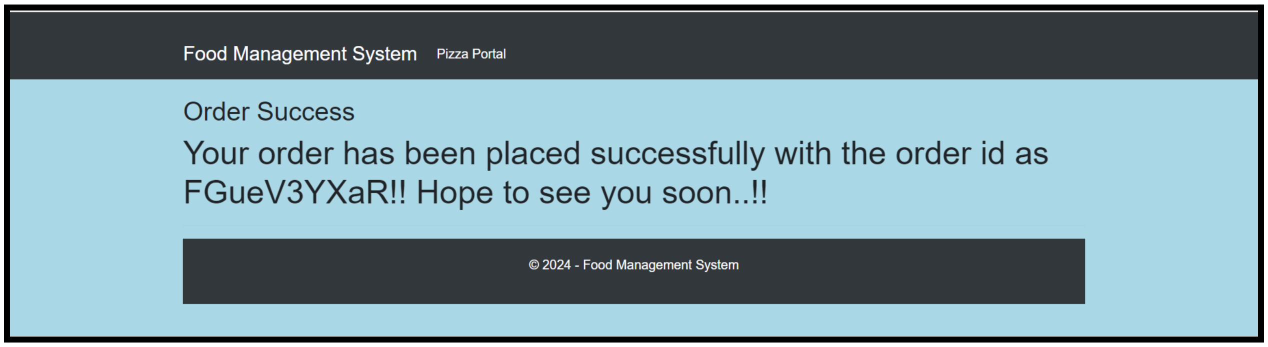
Food Management System Pizza Portal

Payment Status

The Total Price of Pepperoni Pizza = Rs. 501.98
The order will be delivered at Hyderabad, 221401
Order

© 2024 - Food Management System

Order Success View



Food Management System Pizza Portal

Order Success

Your order has been placed successfully with the order id as FGueV3YXaR!! Hope to see you soon..!!

© 2024 - Food Management System