Kitchen Story

Github Repository Link:
https://github.com/Gauravtripathi12345/Kitchen-Story/tree/main

Project Overview:

Kitchen Story is an e-commerce platform designed to facilitate the online purchase of basic food items. The website offers a user-friendly interface for customers to search for, select, and purchase food items. Additionally, it provides an administrative backend for managing food items and user accounts.

Features:

Customer Interface:

- **Home Page:** The home page features a search form allowing customers to enter food items they wish to purchase.
- **Search Functionality:** Customers can search for specific food items, which displays available items along with their prices.
- **Item Selection:** Upon selecting an item, customers are redirected to a page listing available items for purchase.
- **Order Summary:** Customers can view a comprehensive breakdown of their order, including item details and total price.
- **Payment Gateway Integration:** After reviewing the order, customers proceed to the payment gateway to complete their purchase securely.
- **Order Confirmation:** Upon successful payment, customers receive a confirmation page displaying details of their order.

Admin Backend:

- **Login Page:** Admins have a dedicated login page for accessing the backend functionalities.
- **Password Change:** Admins can change their passwords securely after logging in.
- **Food Item Management:**
    - **Master List:** The backend maintains a master list of available food items for purchase.
    - **Addition and Removal:** Admins have the ability to add new food items or remove existing ones from the platform.

Technologies Used:

- **ASP.NET MVC:** The project utilizes the ASP.NET MVC framework for building the web application.
- **C#:** Backend logic and data access are implemented using C# programming language.
- **SQL Server:** Data storage and retrieval are managed through SQL Server databases.
- **Bootstrap and CSS:** Proper styling and layout enhancements are achieved using Bootstrap and CSS for a visually appealing user interface.

Program:

Class Library (KitchenDALLib):

AdminMaster.cs:

```
namespace KitchenDALLib
{
    public class AdminMaster
    {
        public string EmailId { get; set; }
        public string Password { get; set; }
    }
}
```

AdminManager.cs:

```
using System.Configuration;
using System.Data.SqlClient;

namespace KitchenDALLib
{
    public class AdminManager
    {
        public string kitchenStr;
        public SqlConnection con;
        public AdminManager()
        {
            kitchenStr =
ConfigurationManager.ConnectionStrings["KitchenStoryDB"].Connecti
onString;
            con = new SqlConnection(kitchenStr);
        }

        public bool AdminLogin(AdminMaster adminMaster)
        {
            SqlCommand cmd = new SqlCommand("Select Password from
Admin where EmailId = @EmailId", con);
            cmd.Parameters.AddWithValue("@EmailId",
adminMaster.EmailId);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();

            if (reader.Read())
            {
                string strPassword =
reader["Password"].ToString();
                if (strPassword == adminMaster.Password)
                {
                    return true;
                }
            }
            return false;
        }


        public bool ChangePassword(AdminMaster adminMaster)
        {
            SqlCommand cmd = new
SqlCommand("dbo.sp_updatePassword", con);
            cmd.CommandType =
System.Data.CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@p_EmailId",
adminMaster.EmailId);
            cmd.Parameters.AddWithValue("@p_Password",
adminMaster.Password);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            con.Dispose();
            return true;
        }

        public bool validEmail(string EmailId)
        {
            SqlCommand cmd = new SqlCommand("SELECT COUNT(*) FROM
Admin WHERE EmailId = @EmailId", con);
            cmd.Parameters.AddWithValue("@EmailId", EmailId);

            con.Open();
            int count = (int)cmd.ExecuteScalar();
            con.Close();

            return count > 0;
        }
    }
}
```

FoodMaster.cs:

```
namespace KitchenDALLib
{
    public class FoodMaster
    {
        public int Id { get; set; }
        public string FoodName { get; set; }
        public float Price { get; set; }
    }
}
```

FoodItem.cs:

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;

namespace KitchenDALLib
{
    public class FoodItem
    {
        public string kitchenStr;
        public SqlConnection con;

        public FoodItem()
        {
            kitchenStr = 
ConfigurationManager.ConnectionStrings["KitchenStoryDB"].Connecti
onString;
            con = new SqlConnection(kitchenStr);
        }

        public List<FoodMaster> GetAllFoodItem()
        {
            List<FoodMaster> foodItemList = new 
List<FoodMaster>();
            SqlCommand cmd = new SqlCommand("Select * from 
FoodItem",con);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            while(reader.Read())
            {
                FoodMaster item = new FoodMaster();
                item.Id = Convert.ToInt32(reader["Id"]);
                item.FoodName = reader["FoodName"].ToString();
                item.Price = Convert.ToSingle(reader["price"]);
                foodItemList.Add(item);
            }
            con.Close();
            con.Dispose();
            return foodItemList;
        }

        public FoodMaster GetFoodItemById(int id)
        {
            SqlCommand cmd = new SqlCommand("Select * from 
FoodItem where Id = @Id", con);
            cmd.Parameters.AddWithValue("@Id", id);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            FoodMaster item = null;

            if (reader.HasRows)
            {
                reader.Read();

                item = new FoodMaster
                {
                    Id = Convert.ToInt32(reader["Id"]),
                    FoodName = reader["FoodName"].ToString(),
                    Price = Convert.ToSingle(reader["price"])
                };
            }
            con.Close();
            con.Dispose();
            return item;
        }


        public bool AddFoodItem(FoodMaster foodMaster)
        {
            SqlCommand cmd = new
SqlCommand("dbo.sp_insertFoodItem", con);
            cmd.CommandType =
System.Data.CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@p_FoodName",
foodMaster.FoodName);
            cmd.Parameters.AddWithValue("@p_Price",
foodMaster.Price);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            con.Dispose();
            return true;
        }

        public bool UpdateFoodItem(FoodMaster foodMaster) {
            SqlCommand cmd = new
SqlCommand("dbo.sp_updateFoodItem", con);
            cmd.CommandType =
System.Data.CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@p_Id", foodMaster.Id);
            cmd.Parameters.AddWithValue("@p_FoodName",
foodMaster.FoodName);
```

```
            cmd.Parameters.AddWithValue("@p_Price",
foodMaster.Price);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            con.Dispose();
            return true;
        }

        public bool DeleteFoodItem(int id) {
            SqlCommand cmd = new SqlCommand("Delete * from
FoodItem where Id="+id, con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            con.Dispose();
            return true;
        }
    }
}
```

MVC Controller

AdminController:

```
using KitchenDALLib;
using KitchenStoryProject.Models;
using System;
using System.Web.Mvc;

namespace KitchenStoryProject.Controllers
{
    public class AdminController : Controller
    {
        AdminManager adminManagerDal = new AdminManager();
        // GET: Admin
        public ActionResult Login()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Login(AdminModel adminModel)
        {
            AdminMaster adminMaster = new AdminMaster()
            {
                EmailId = adminModel.EmailId,
                Password = adminModel.Password,
            };
            try
            {
                bool result =
adminManagerDal.AdminLogin(adminMaster);
                if (result)
                {
                    return RedirectToAction("Index","FoodItem");
                }
                else
                {
                    return Content("Invalid Login Credentials");
                }
            }
            catch (Exception)
            {
                return Content("Invalid Login");
            }
        }

        public ActionResult PasswordChange()
        {
            return View();
        }

        [HttpPost]
        public ActionResult PasswordChange(AdminModel adminModel)
        {
            bool validEmail =
adminManagerDal.validEmail(adminModel.EmailId);
            if (validEmail)
            {
                AdminMaster adminMaster = new AdminMaster()
                {
                    EmailId = adminModel.EmailId,
                    Password = adminModel.Password,
                };
                try
                {
                    bool result =
adminManagerDal.ChangePassword(adminMaster);
                    if (result)
                    {
                        return RedirectToAction("SuccessPage");
                    }
                }
                catch (Exception)
                {
                    return Content("Invalid Login");
                }
            }
            return View();
        }

        public ActionResult SuccessPage()
        {
            return View();
        }
    }
}
```

FoodItemController.cs:

```
using KitchenDALLib;
using KitchenStoryProject.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;

namespace KitchenStoryProject.Controllers
{
    public class FoodItemController : Controller
    {
        FoodItem foodItemDal = new FoodItem();
        List<FoodItemModel> foodItemModelList = new
List<FoodItemModel>();
        List<FoodMaster> foodItemsList;

        // GET: FoodItem
        public ActionResult Index()
        {
            foodItemsList = foodItemDal.GetAllFoodItem();
            foreach (var item in foodItemsList)
            {
                FoodItemModel foodItemModel = new
FoodItemModel();

                foodItemModel.Id = item.Id;
                foodItemModel.FoodName = item.FoodName;
                foodItemModel.Price = item.Price;

                foodItemModelList.Add(foodItemModel);
            }
            return View(foodItemModelList);
        }

        // GET: FoodItem/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return Content("Id not found");
            }

            try
            {
                FoodMaster foodMaster =
foodItemDal.GetFoodItemById(id.Value);
                if (foodMaster != null)
                {
                    FoodItemModel foodItemModel = new
FoodItemModel()
                    {
                        Id = foodMaster.Id,
                        FoodName = foodMaster.FoodName,
                        Price = foodMaster.Price
                    };
                    return View(foodItemModel);
                }
                else
                {
                    return Content("Invalid FoodItem id");
                }
            }
            catch (Exception)
            {
                return Content("Error fetching FoodItem
details");
            }
        }

        // GET: FoodItem/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: FoodItem/Create
        [HttpPost]
        public ActionResult Create(FormCollection collection)
        {
            try
            {
                // TODO: Add insert logic here
                FoodItemModel foodItemModel = new FoodItemModel()
                {
                    FoodName = collection["FoodName"].ToString(),
                    Price = Convert.ToSingle(collection["Price"])
                };
                FoodMaster foodMaster = new FoodMaster()
                {
                    FoodName = foodItemModel.FoodName,
                    Price = foodItemModel.Price
                };

                bool result =
foodItemDal.AddFoodItem(foodMaster);
                if (result)
                {
                    return RedirectToAction("Index");
                }
            }
            catch (Exception)
            {
                return Content("Invalid Item entry");
            }
            return View();
        }

        // GET: FoodItem/Edit/5
        public ActionResult Edit(int id)
        {
            FoodMaster foodMaster =
foodItemDal.GetFoodItemById(id);
            FoodItemModel foodItemModel = new FoodItemModel()
            {
                Id = foodMaster.Id,
                FoodName = foodMaster.FoodName,
                Price = foodMaster.Price
            };
            return View(foodItemModel);
        }

        // POST: FoodItem/Edit/5
        [HttpPost]
        public ActionResult Edit(int id, FormCollection 
collection)
        {
            try
            {
                // TODO: Add update logic here
                FoodItemModel foodItemModel = new FoodItemModel()
                {
                    Id = int.Parse(collection["Id"]),
                    FoodName = collection["FoodName"].ToString(),
                    Price = Convert.ToSingle(collection["Price"])
                };
                FoodMaster foodMaster = new FoodMaster()
                {
                    Id = foodItemModel.Id,
                    FoodName = foodItemModel.FoodName,
                    Price = foodItemModel.Price
                };
                bool result = 
foodItemDal.UpdateFoodItem(foodMaster);
                if (result)
                {
                return RedirectToAction("Index");
                }
            }
            catch(Exception)
            {
                return Content("Invalid entry of the item to be 
Updated");
            }
            return View();
        }

        // GET: FoodItem/Delete/5
        public ActionResult Delete(int id)
        {
            FoodMaster foodMaster = 
foodItemDal.GetFoodItemById(id);
            FoodItemModel foodItemModel = new FoodItemModel()
            {
                Id = foodMaster.Id,
                FoodName = foodMaster.FoodName,
                Price = foodMaster.Price
            };
            return View(foodItemModel);
        }

        // POST: FoodItem/Delete/5
        [HttpPost]
        public ActionResult Delete(int id, FormCollection 
collection)
        {
            try
            {
                // TODO: Add delete logic here
                bool result = foodItemDal.DeleteFoodItem(id);
                if (result)
                {
                return RedirectToAction("Index");
                }
            }
            catch(Exception)
            {
                return Content("No item with this id");
            }
            return View();
        }

        public ActionResult FoodMenu()
        {
            var foodItems = foodItemDal.GetAllFoodItem();

            var foodItemModels = new List<FoodItemModel>();
            foreach (var foodItem in foodItems)
            {
                var foodItemModel = new FoodItemModel
                {
                    Id = foodItem.Id,
                    FoodName = foodItem.FoodName,
                    Price = foodItem.Price
                };
                foodItemModels.Add(foodItemModel);
            }
            return View(foodItemModels);
        }

        [HttpPost]
        public ActionResult FoodMenu(string searchString)
        {
            var foodItems = foodItemDal.GetAllFoodItem();

            if (!string.IsNullOrEmpty(searchString))
            {
                string searchLower = searchString.ToLower();
                foodItems = foodItems.Where(f =>
f.FoodName.ToLower().Contains(searchLower)).ToList();
            }

            var foodItemModels = foodItems.Select(foodItem => new
FoodItemModel
            {
                Id = foodItem.Id,
                FoodName = foodItem.FoodName,
                Price = foodItem.Price
            }).ToList();

            if (!foodItemModels.Any())
            {
                ViewBag.NoResultsMessage = "No products found for
the given search criteria.";
            }
            return View(foodItemModels);
        }

        public ActionResult SelectedItems(int id)
        {
            foodItemsList = foodItemDal.GetAllFoodItem();
            FoodMaster foodItem = foodItemsList.Find(f => f.Id ==
id);
            TempData["Price"] = foodItem.Price;
            TempData["FoodItem"] = foodItem.FoodName;
            TempData.Keep();
            return View();
        }

        [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()
        {
            return View();

        }

        public ActionResult OrderSuccess()
        {
            return View();

        }
    }
}
```

MVC Model

AdminModel.cs:

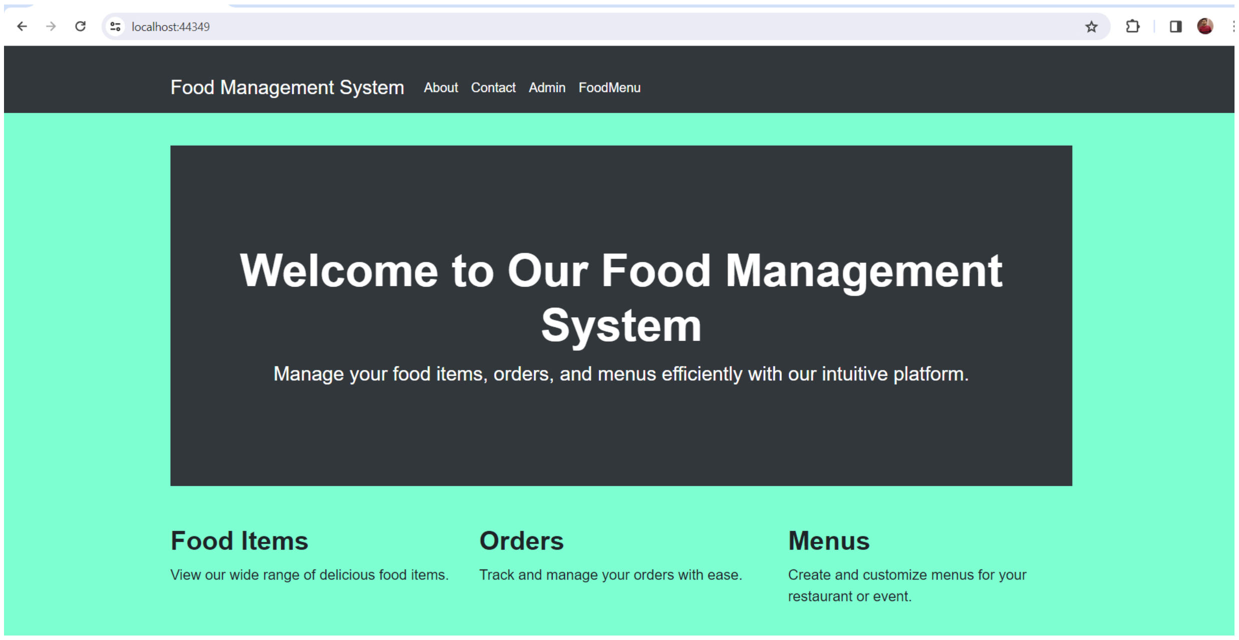
```
namespace KitchenStoryProject.Models
{
    public class AdminModel
    {
        public string EmailId { get; set; }
        public string Password { get; set; }
    }
}
```

FoodItemModel.cs:

```
namespace KitchenStoryProject.Models
{
    public class FoodItemModel
    {
        public int Id { get; set; }
        public string FoodName { get; set; }
        public float Price { get; set; }
    }
}
```

Output Screenshots:

1. Home Page


localhost:44349

Food Management System About Contact Admin FoodMenu

Welcome to Our Food Management System

Manage your food items, orders, and menus efficiently with our intuitive platform.

Food Items

View our wide range of delicious food items.

Orders

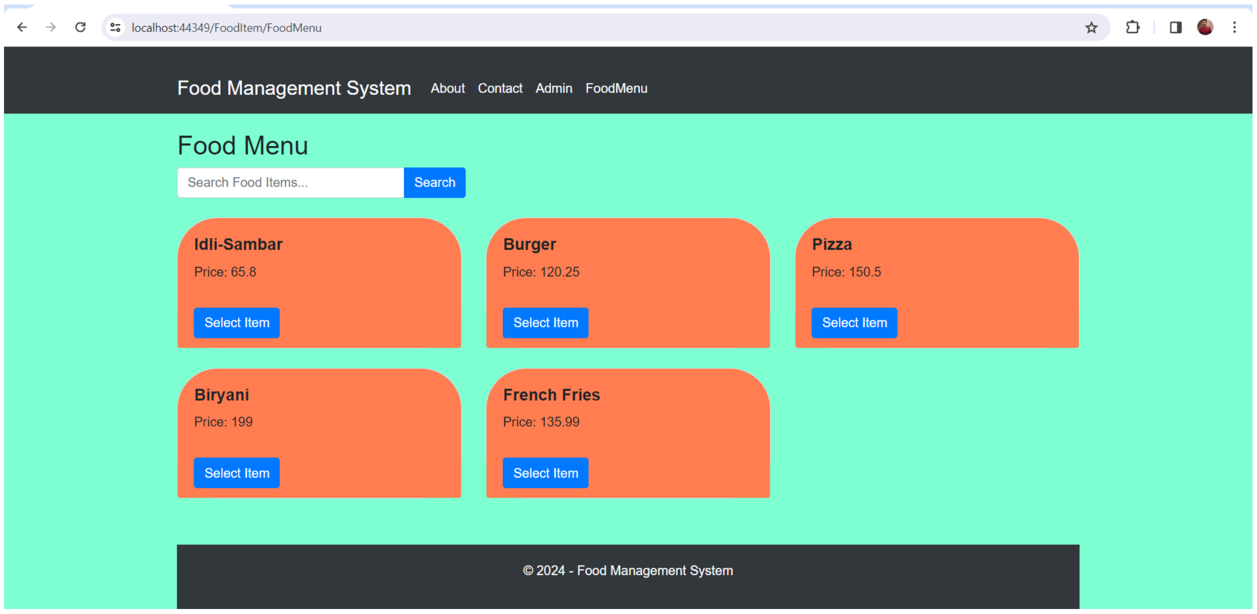
Track and manage your orders with ease.

Menus

Create and customize menus for your restaurant or event.

2. Customer View

a. FoodMenu


localhost:44349/FoodItem/FoodMenu

Food Management System About Contact Admin FoodMenu

Food Menu

Search Food Items... Search

Idli-Sambar
Price: 65.8
Select Item

Burger
Price: 120.25
Select Item

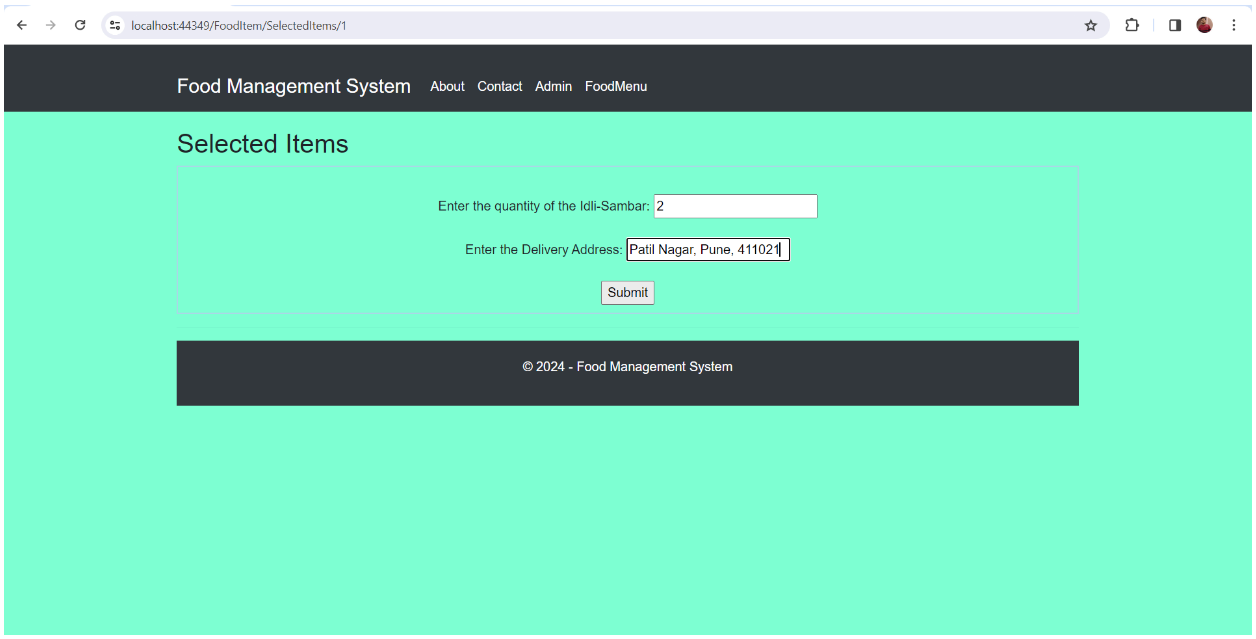
Pizza
Price: 150.5
Select Item

Biryani
Price: 199
Select Item

French Fries
Price: 135.99
Select Item

© 2024 - Food Management System

b. Selected Item



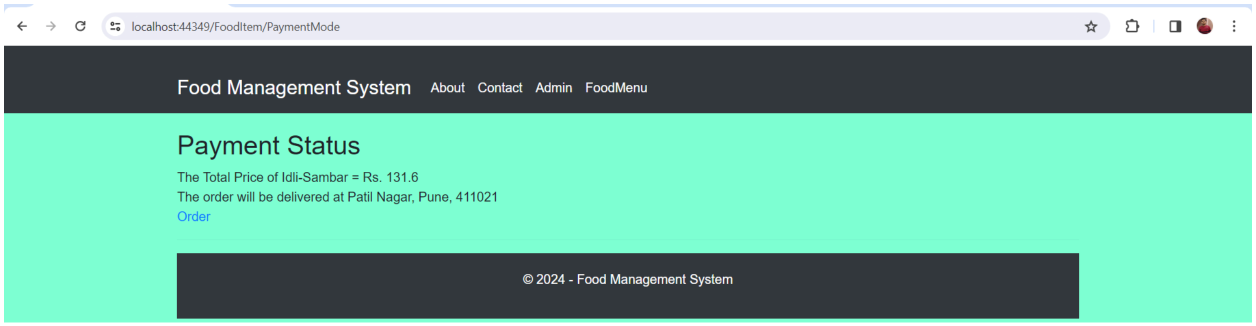
Selected Items

Enter the quantity of the Idli-Sambar: 2

Enter the Delivery Address: Patil Nagar, Pune, 411021

© 2024 - Food Management System

c. Payment Status

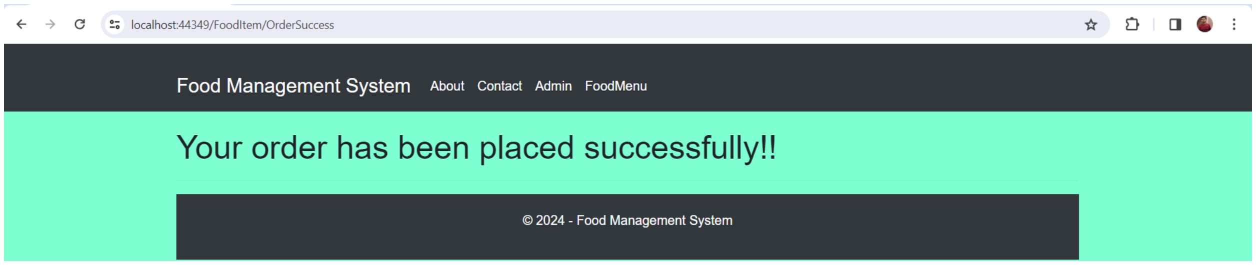


Payment Status

The Total Price of Idli-Sambar = Rs. 131.6
The order will be delivered at Patil Nagar, Pune, 411021
Order

© 2024 - Food Management System

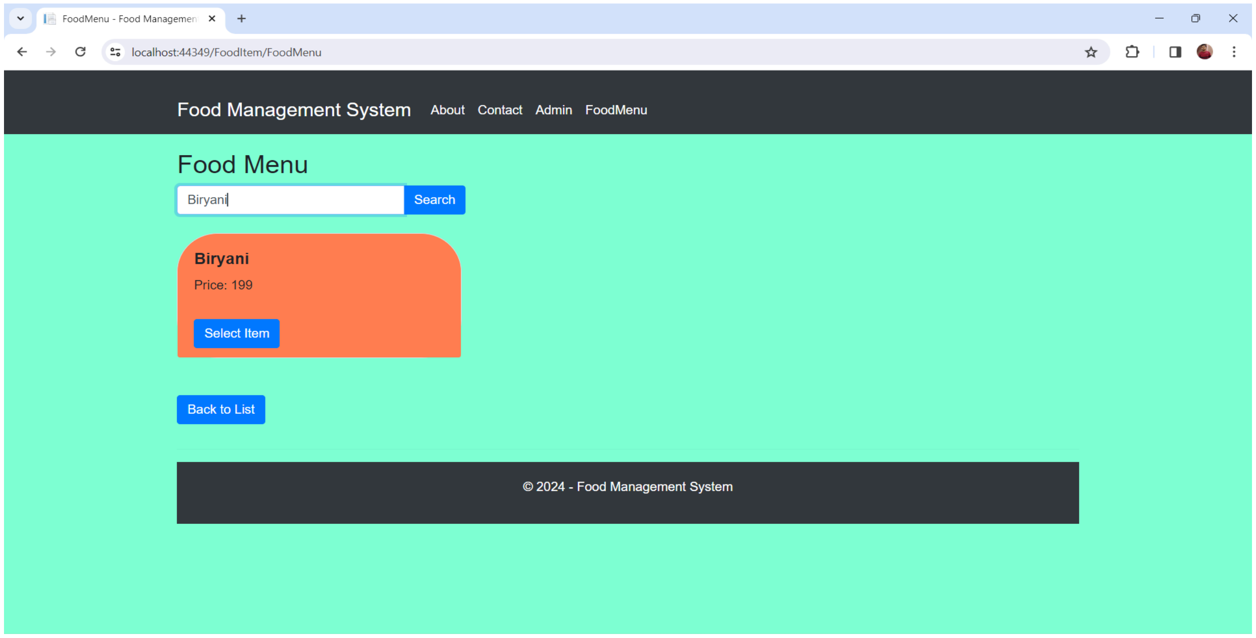
d. Order Status:



Your order has been placed successfully!!

© 2024 - Food Management System

e. Search Food Item:



Food Menu

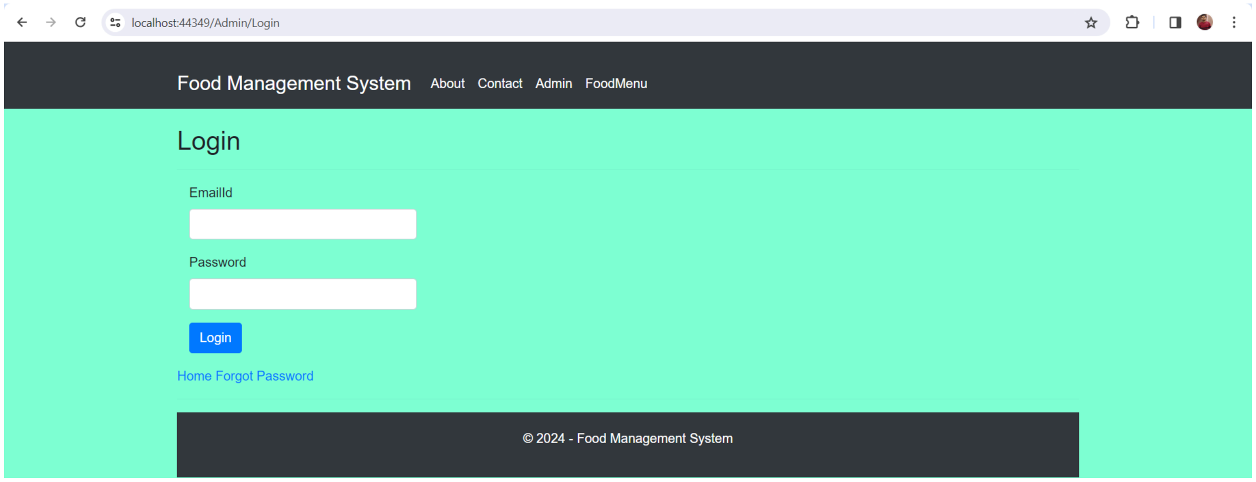
Biryani

Price: 199

© 2024 - Food Management System

3. Admin View:

a. Login:



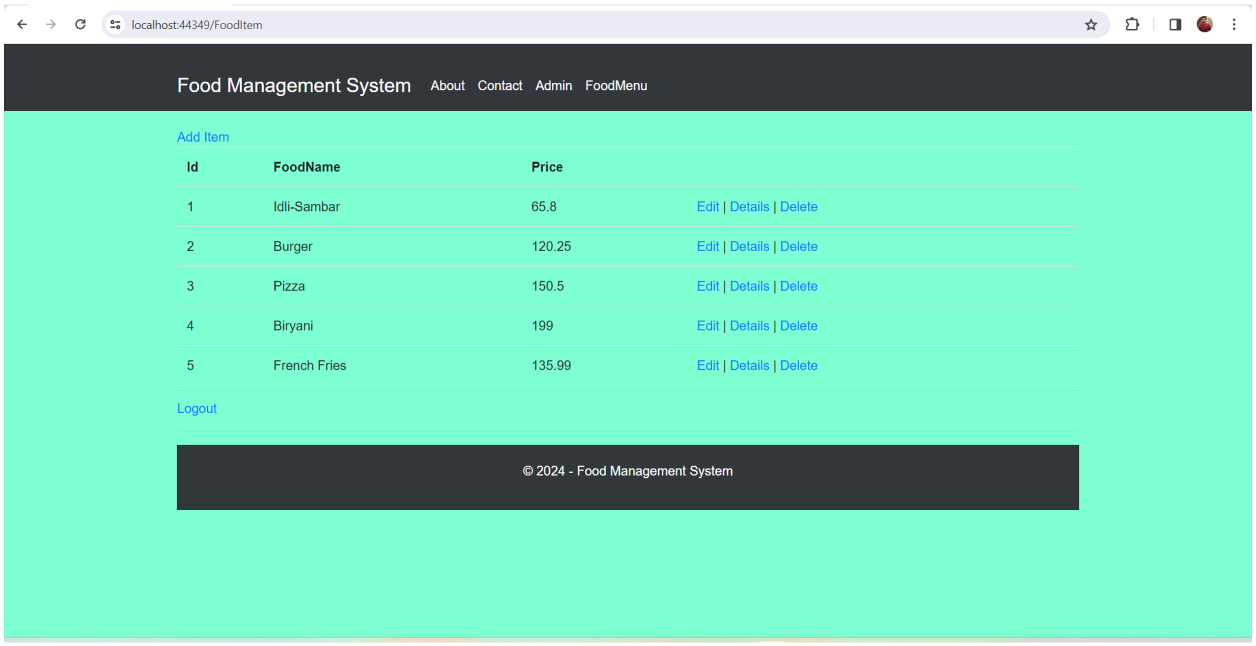
Login

EmailId

Password

© 2024 - Food Management System

b. Post Login View


localhost:44349/FoodItem

Food Management System About Contact Admin FoodMenu

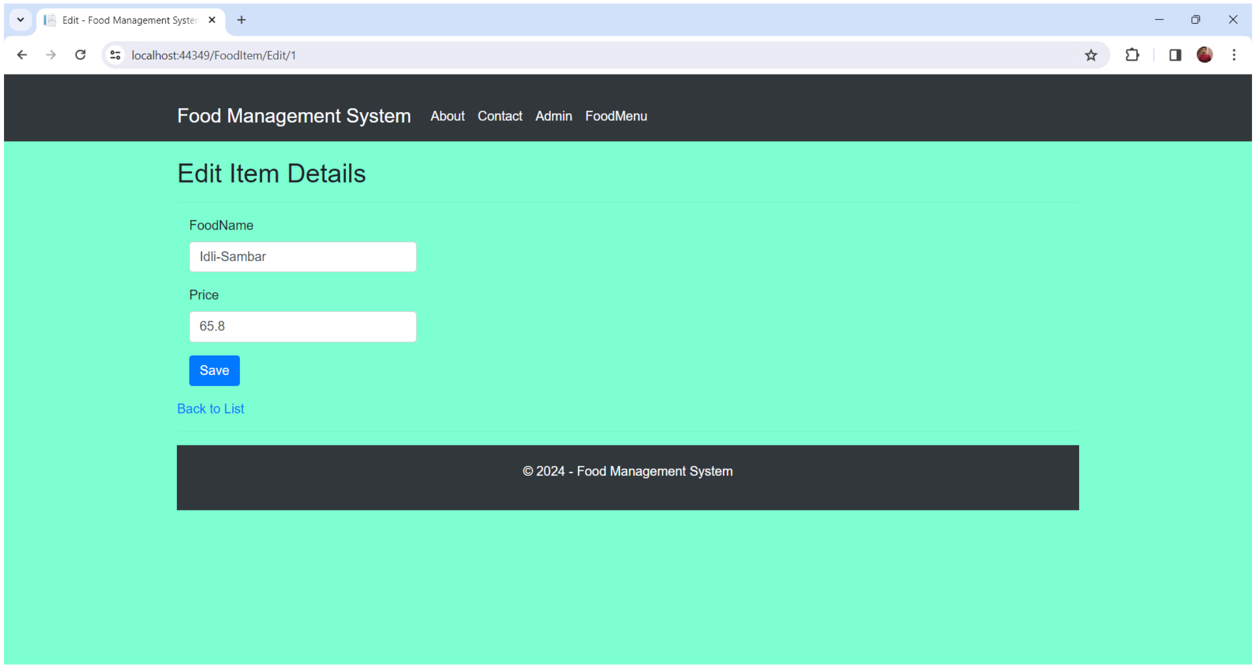
Add Item

Id	FoodName	Price	
1	Idli-Sambar	65.8	Edit | Details | Delete
2	Burger	120.25	Edit | Details | Delete
3	Pizza	150.5	Edit | Details | Delete
4	Biryani	199	Edit | Details | Delete
5	French Fries	135.99	Edit | Details | Delete

Logout

© 2024 - Food Management System

c. Edit View:


Edit - Food Management Syster

localhost:44349/FoodItem/Edit/1

Food Management System About Contact Admin FoodMenu

Edit Item Details

FoodName

Idli-Sambar

Price

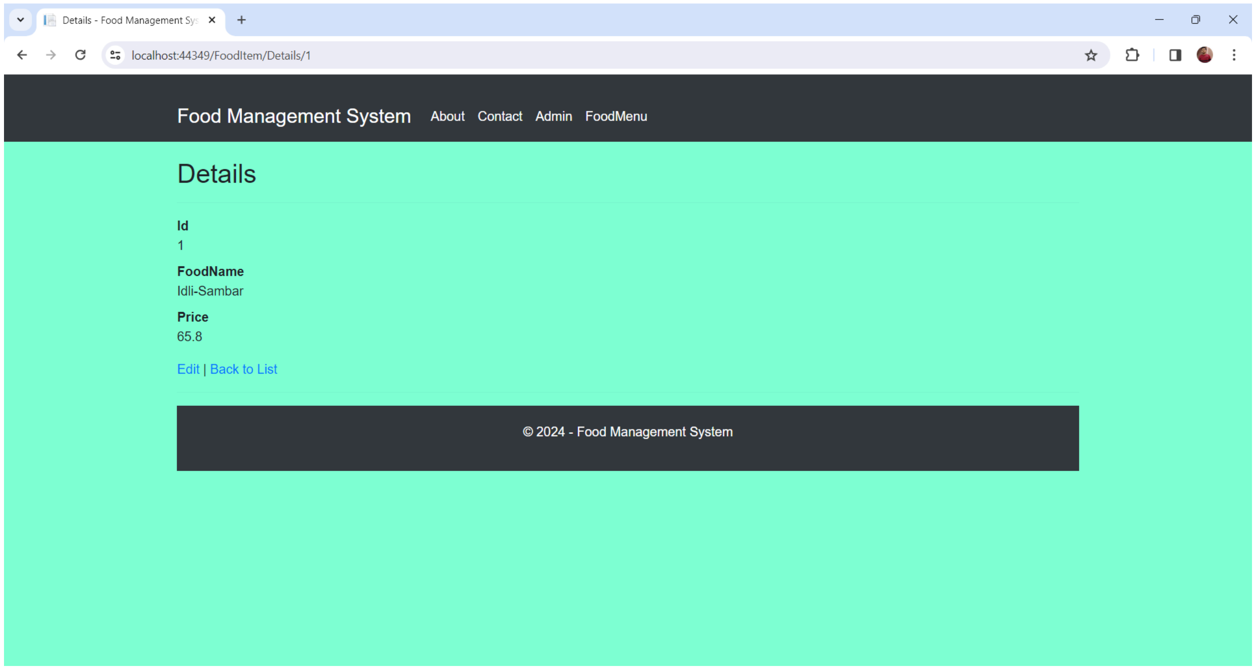
65.8

Save

Back to List

© 2024 - Food Management System

d. Details View:


Details - Food Management Sy

localhost:44349/FoodItem/Details/1

Food Management System About Contact Admin FoodMenu

Details

Id
1

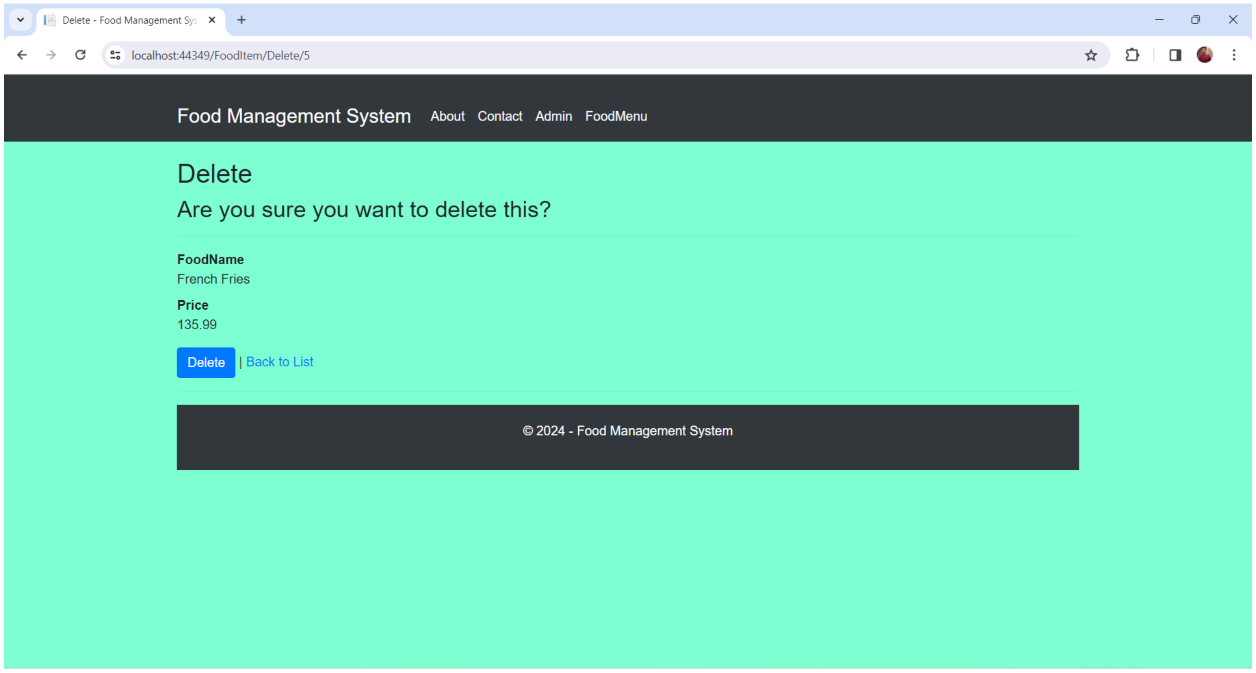
FoodName
Idli-Sambar

Price
65.8

Edit | Back to List

© 2024 - Food Management System

e. Delete Food Item View


Delete - Food Management Sy

localhost:44349/FoodItem/Delete/5

Food Management System About Contact Admin FoodMenu

Delete

Are you sure you want to delete this?

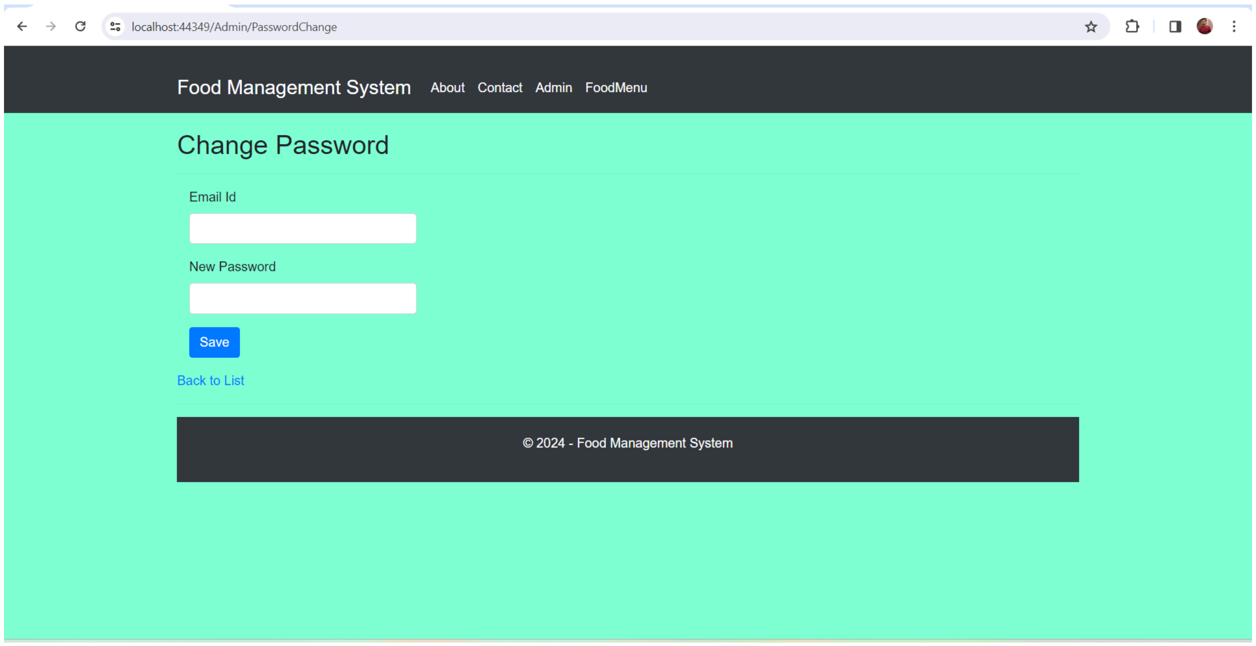
FoodName
French Fries

Price
135.99

Delete | Back to List

© 2024 - Food Management System

f. Forgot Password View:



Change Password

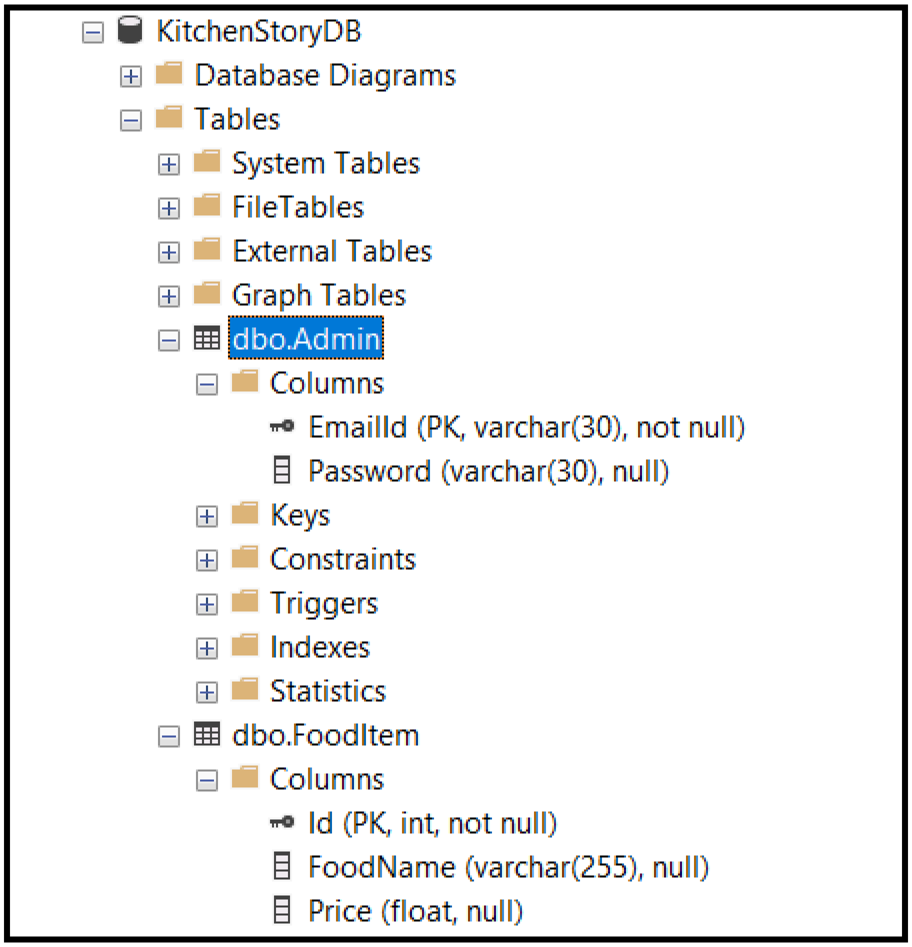
Email Id

New Password

© 2024 - Food Management System

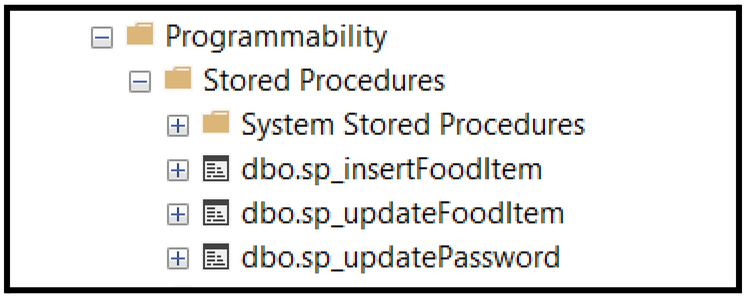
Database Overview:

- The Database consists of 2 tables:
  - Admin Table
  - FoodItem Table
- The columns and the description of the same are depicted in the image given below:



KitchenStoryDB
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Admin
Columns
EmailId (PK, varchar(30), not null)
Password (varchar(30), null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.FoodItem
Columns
Id (PK, int, not null)
FoodName (varchar(255), null)
Price (float, null)

- The Stored Procedure used in the Project are given below:



Programmability
Stored Procedures
System Stored Procedures
dbo.sp_insertFoodItem
dbo.sp_updateFoodItem
dbo.sp_updatePassword