

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2022

Management System for a Restaurant

Srikar Macha

Governors State University

Follow this and additional works at: <https://opus.govst.edu/capstones>

Recommended Citation

Macha, Srikar, "Management System for a Restaurant" (2022). *All Capstone Projects*. 570.
<https://opus.govst.edu/capstones/570>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

MANAGEMENT SYSTEM FOR A RESTAURANT

By

Srikar Macha

B. Tech, VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY, 2017

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science



Governors State University
University Park, IL 60484

2022

ABSTRACT

The Management System for a Restaurant is a web application that allows customers to make online orders by choosing their favorite food items from the restaurant's menu. This website's user-friendly layout organizes a variety of food items.

Customers can order food courses and items online and pick them up at a nearby location. This web application consists of a database that stores information about customers, the food menu, and inventory. This application provides a feasible way to order inventory for owners.

The motive for designing this web application is to encourage the cloud kitchen concept. Nowadays, due to COVID, most customers are attracted to ordering food online, which is more convenient. Due to this transition, it's imperative that restaurants provide online services to recover from their past financial crises. Customers and restaurants benefit equally from it.

This application has the scope of enhancing the existing application in terms of performance, efficiency, and an effective user interface by using the Model-View-Controller (MVC) framework and bootstrap technologies.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements.....	2
1.5	Definitions and Acronyms	2
2	<i>Project Technical Description</i>	2
2.1	Application Architecture	2
2.2	Application Information flows.....	5
2.3	Interactions with other Applications	5
2.4	Capabilities	5
2.5	Risk Assessment and Management.....	6
3	<i>Project Requirements</i>	6
3.1	Identification of Requirements	6
3.2	Operations, Administration, Maintenance and Provisioning (OAM&P)	7
3.3	Security and Fraud Prevention.....	7
3.4	Release and Transition Plan.....	8
4	<i>Project Design Description</i>	8
5	<i>Internal/external Interface Impacts and Specification</i>	11
6	<i>Design Units Impacts</i>	18
6.1	Functional Area A/Design Unit A	19
6.1.1	<i>Functional Overview</i>	19
6.1.2	<i>Impacts</i>	19
7	<i>Open Issues</i>	19
8	<i>Acknowledgements</i>	20
9	<i>References</i>	20

1 Project Description

The Management System for a Restaurant web application assists customers in ordering their favorite food items online by navigating to the website's food menu and categories. This application mainly consists of two roles, admin, and customer. The restaurant menu is available for all customers to browse on the website. Customers need to be authenticated whenever they try to add an item to the cart or register as new customers. Once the customer has logged into the website, they will add, modify, or delete the food items in the cart, and place an order. Administrators, who are usually restaurant owners, will accept the order and give regular updates about the order's status to the customer. A user with admin privileges can also modify orders, food items, and inventory. Admins can also view the weekly report, most ordered items, and most valued customers.

1.1 Competitive Information

Like many other food ordering systems, our application stands out among others in terms of design specifications and user interface. This application allows users to easily navigate food menus based on categories and displays warning messages wherever necessary in a pleasant manner.

1.2 Relationship to Other Applications/Projects

The sole objective of this application is to allow customers to order food by logging onto our website. Since this core functionality has been implemented by many applications hosted on the internet, they may offer different services to users depending on their interests.

1.3 Assumptions and Dependencies

We need to create APIs for this web application so that it can be easily integrated with other services. Admins have authorization to manage the permissions granted to customers.

1.4 Future Enhancements

In the future, this application can be enhanced in an effective way by improving user-friendly functionality with advanced filters and UX improvements. And we are planning to introduce live delivery tracking of orders by integrating with the Google Maps API and a payment gateway for successful money transactions.

1.5 Definitions and Acronyms

UI – User Interface.

Viz – Visualization.

UX – User Experience.

ASP – Active Server Pages.

MVC – Model-View-Controller.

ORM – Object Relational Mapping.

LINQ – Language Integrated Query.

2 Project Technical Description

This project comprises two main modules: Admin and Customer. The objective of this project is to deliver a customer-friendly web application so that customers can easily navigate between website links to create an online order.

2.1 Application Architecture

This Application was developed based on two main frameworks, Model-View-Controller and Entity Core framework [1],[2],[3].

Model-View-Controller (MVC)

MVC generally divides web applications into three parts: Model, View, and Controllers. The end user will initially send a request to the server, and then the controller will handle the request and pass it on to the model. After interacting with the database and performing some functionality, the model will then fetch the data from the table and return it to the controller. The controller will then send the data to the view. View will format the data, and this formatted data will be displayed to the client [1].

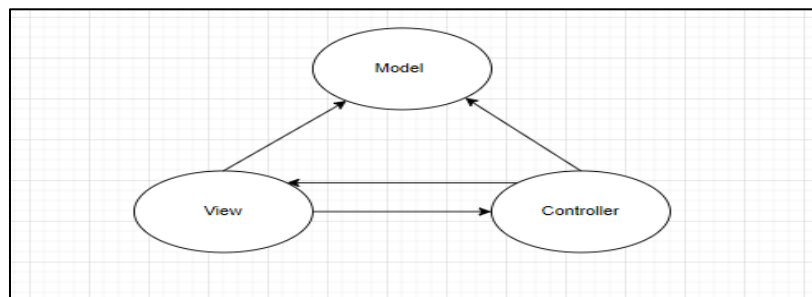


Figure 1: Design flow of MVC framework.

Below you can see the project directories categorized according to MVC framework.

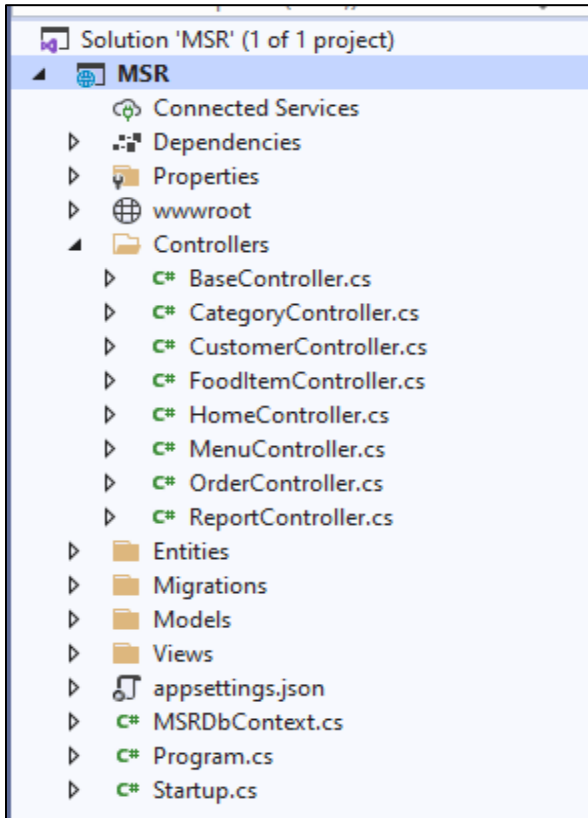


Figure 2: Controller Structure in the Solution.

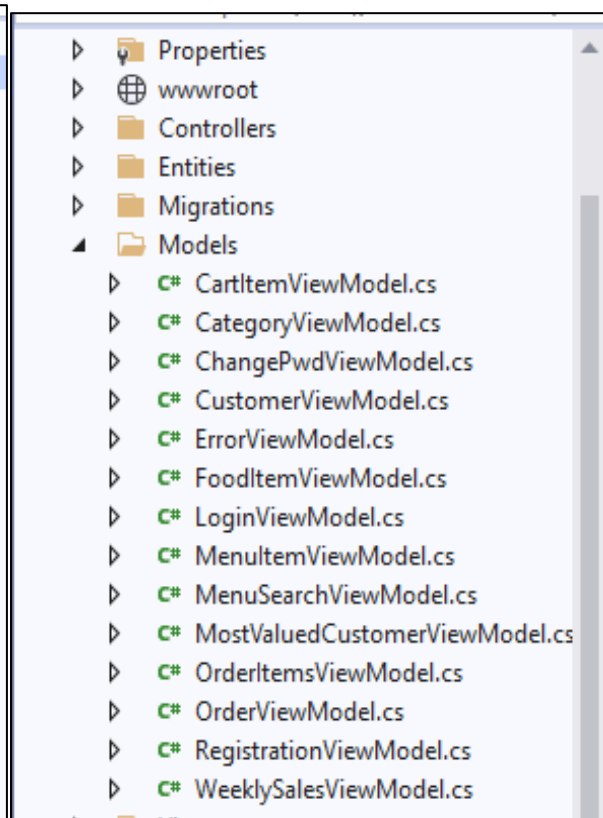


Figure 3: Model Structure in the Solution.

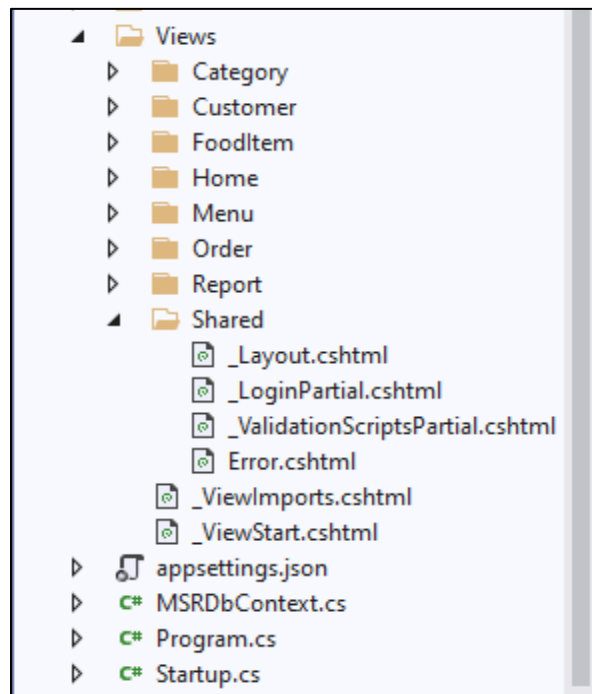


Figure 4: View Hierarchy in the Solution.

Entity Core Framework

For reading and managing data in databases, we use Entity Framework Core (EF Core), which is an object-relational mapping (ORM) framework that is lightweight and cross-platform [3]. EF Core supports both the code-first and data-first approaches. This project follows the Code First approach, where we will design our application domain classes and a special class that is derived from the entity framework DB context class. EF Core will create the database and the relevant tables.

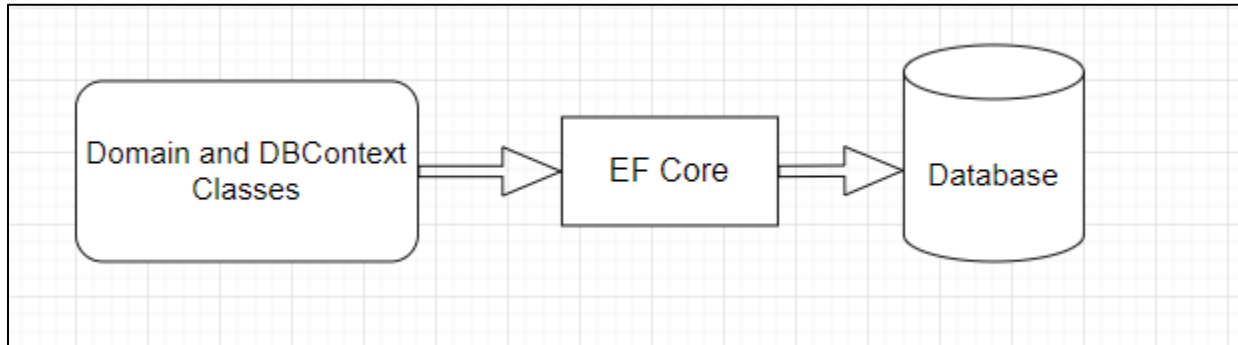


Figure 5: Entity Core Framework Design.

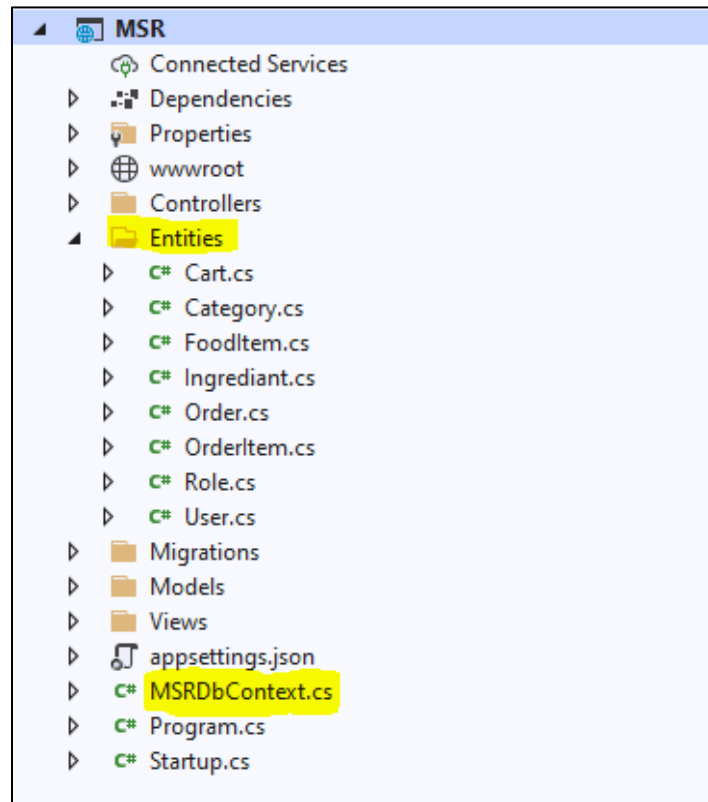


Figure 6: dB Context Class for Entity core Design.

2.2 Application Information flows

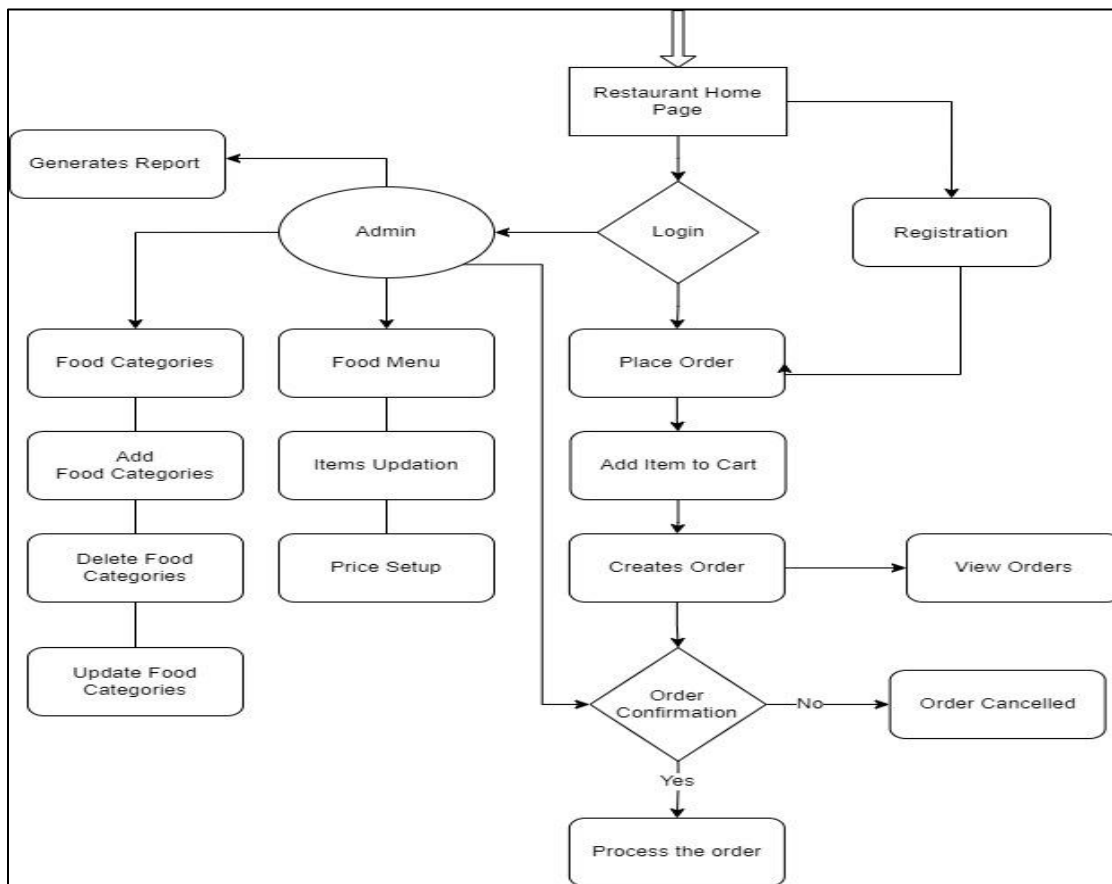


Figure 7: Application Design Flow.

2.3 Interactions with other Applications

This web application can easily be integrated with other online food ordering platforms like Uber Eats, Grub Hub, and Door Dash. Users can browse for various restaurants through these restaurant services. It also interacts with payment gateway services for successful banking transactions involving an order. A notification will be displayed to the customer regarding the transaction details.

2.4 Capabilities

The project must be able to fulfill all specifications for both users in terms of its capabilities. The development of this application used a variety of technologies, including SQL language for the database part and ASP.NET for server-side programming. Through the model classes, this application reads, edits, and deletes data from the database with SQL. The Controller classes are used to write business logic, like calculating the price of an order and setting the price of food items at the backend. This is done through the ASP.NET programming language. The View classes display data on web pages with HTML, CSS, and Bootstrap for users and administrators.

2.5 Risk Assessment and Management

If there are any risks, the development team will take steps to control them throughout the project by quickly identifying, analyzing, and resolving them. To minimize the damage, they will find out the issues and resolve them by following effective measures. To keep track of all the issues and their mitigation strategies and to enhance them for later use, the team will establish a risk analysis record. Risks are categorized as high, medium, or low based on their impact on the project, with the highest risks being mitigated first, followed by others.

3 Project Requirements

This project practices the Agile Methodology, which is one of the software development methodologies followed by many information technology industries to develop their projects. In this methodology, each functionality will be divided into user stories, and developers will pick these user stories and develop the application functionality described in the user stories.

3.1 Identification of Requirements

The core functionality and the business expectations that are needed for developing this application were gathered from the client application proposal document. And these requirements are analyzed and broken down into several functionalities.

<GSU_GS_CS8985_01-MSR ver1.0 Database-Setup “001”>

First, we need to identify the entities to set up the database design and tables. Installation and configuration of the database give a strong foundation for developing the application base. In this project, we are using an SQL Server database to store and retrieve the data.

<GSU_GS_CS8985_01-MSR ver1.0 – Login Functionality - “001”>

Create login functionality for users and admins. Users can authenticate the website by entering their username and password. If the user is unable to authenticate, the application generates a warning message for an invalid login attempt.

<GSU_GS_CS8985_01-MSR ver1.0 User-Registration - “001”>

Create a user registration feature for the website. The registration fields should consist of user information like name, email address, mobile number, and password. Users were only allowed to register successfully once all fields were validated.

<GSU_GS_CS8985_01-MSR ver1.0 Admin-functionality - “001”>

Define a job role which is categorized into admin and user. Create admin login credentials and assign privileges where admin only has the authorization to manage the stock, food items, categories, and orders.

<GSU_GS_CS8985_01-MSR ver1.0 Front-End-UI-Design - “001”>

Application design and user interaction for the application functionality were developed to meet this requirement. UI components, interactive buttons, customizing the interface, themes for each web page, error messages, and graphic content were developed in this functionality with the bootstrap and HTML.

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

This project involved numerous administrative and user-performed tasks. A user who can access the website and place an order. This process consists of various operations performed by the user. First, the user logs into the website with their username and password. Next, users can search for the food items they are looking for and filter the items by categories. Users can add food items to the cart, delete food items, and place an order.

Admin also performs operations like adding food items and categories to the restaurant menu. And admin can control the flow of orders made by the customers by giving order updates. Application maintenance was done at regular intervals by giving security updates, analyzing the code quality, updating the application software, and analyzing the incidents created by the user on the website support portal.

3.3 Security and Fraud Prevention

For security purposes, we provide role-based authorization as well as claim-based authorization [4]. In general, authorization is the procedure of deciding what the user who is currently logged in is allowed and cannot do. Based on the application authorization requirements, in ASP.NET core we can use claim-based authorization, role-based authorization or a combination of both. For this application, we provide two types of authorizations, Admin and Customer, so that specific functionalities can be accessed by individuals who have been granted authorization. Claim based authorization stores the trusted user’s identity and allows access to the users based on the values defined in that identity.

3.4 Release and Transition Plan

After the final development of the application, the testing team will test the entire application's functionality in the development and test environments with a quality analyst team. After being tested by the QA team, the application will be deployed in a non-production environment where product owners and business owners will test it. Once the application passes all the functional test cases, the developers will deploy the application on a production server, and the website will go live.

4 Project Design Description

There are two roles in this application's user interface: admin and customer. When the application runs, the landing page appears in the browser. On this landing page, there are some functionalities present that are mentioned below.

- I. Place order
- II. Registration
- III. Login

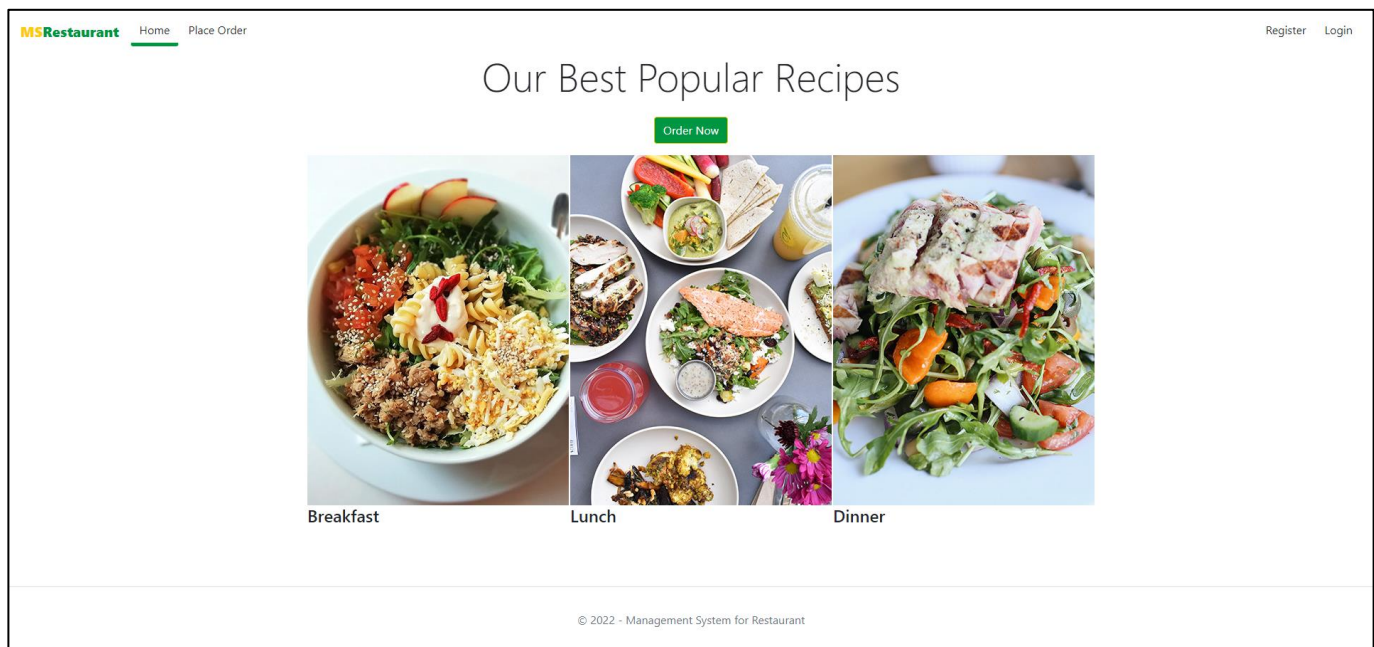


Figure 8: Landing Page of the Application.

On this page, when we navigate to the Order Now link, we will be able to see the restaurant menu. Whenever a user tries to add a food item to the cart, the application checks the user's authentication. If the user is logged in, the application allows them to add food items to their cart; otherwise, they are directed to the login page for authentication.

In the customer role, the application page is redirected to the customer user interface, which consists of many functionalities, including:

- I. Place Order
- II. My Cart
- III. My Orders
- IV. Change Password
- V. Logout.

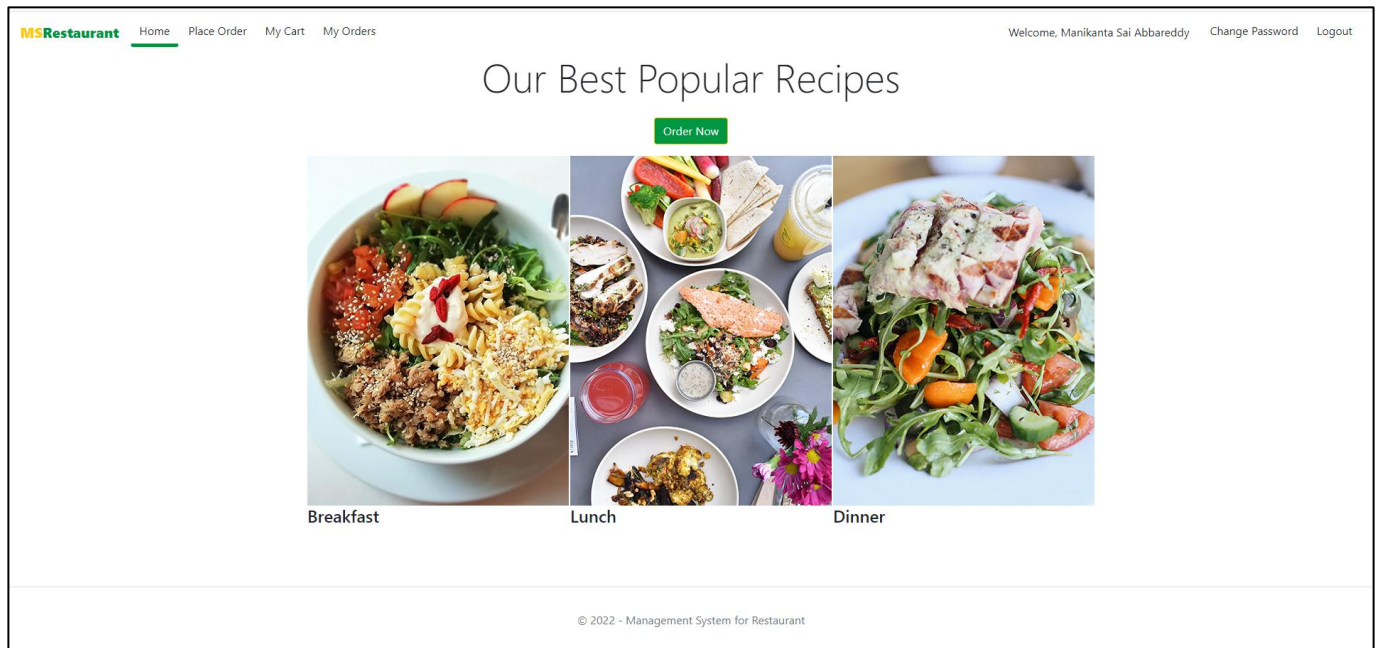


Figure 9: Customer Role Page.

When the customer clicks the place order button, they will navigate to the restaurant menu. Customers can search for the food item and filter it by categories. Once the customer adds the items to the cart, the items are shown on the My Cart page. On this page, customers can manage the quantity of a food item and can delete the item from the cart using the delete item button. Customers will create the order by clicking the create order button. When the customer creates the order, then it will pop up on the My Orders page with Order View Model attributes. The My Orders page of the Customer screen will update when the admin modifies the order's status in the admin user interface. Every user can change the password using the change password button. The customer can return to the application's home page by using logout.

In the admin role, the application user interface consists of some functionalities, which are listed below:

- I. Orders
- II. Food categories
- III. Food Items
- IV. Customers
- V. Reports
- VI. Change Password
- VII. Logout.

Order Id	Order Date	Customer Name	# Items	Total Price	Status	
10	11/23/2022 4:26:27 PM	Abbareddy	3	\$54.00	Order Placed	View/Update Order
9	11/10/2022 6:41:09 PM	Abbareddy	1	\$10.00	Order Placed	View/Update Order
8	11/10/2022 6:29:40 PM	Abbareddy	2	\$35.00	Order Placed	View/Update Order
7	11/10/2022 6:23:46 PM	Abbareddy	1	\$10.00	Order Placed	View/Update Order
6	11/10/2022 6:02:21 PM	Abbareddy	1	\$10.00	Order Placed	View/Update Order
5	11/10/2022 6:01:41 PM	Abbareddy	1	\$10.00	Order Placed	View/Update Order
4	11/10/2022 11:35:37 AM	Abbareddy	13	\$218.00	Completed	View/Update Order
3	10/27/2022 2:18:21 PM	Abbareddy	0	\$90.00	Order Confirmed	View/Update Order
2	10/27/2022 12:26:48 PM	Abbareddy	0	\$60.00	Completed	View/Update Order
1	10/27/2022 12:24:16 PM	Abbareddy	0	\$30.00	Cancelled	View/Update Order

Figure 10: Admin Role Page.

We initially fixed the credentials of the administrator in the code itself. Once the user enters the valid inputs for the admin, the application enters the admin role orders page. On that page, the admin can see all the orders made by all customers and update the status of the orders. Admin has the capability to add, edit, and delete the food categories and food items in the respective web pages. Admin has access to the application's registered customers via the Customer button. Admin can see the most valued customer, most ordered item, and weekly sales by using the Reports page.

5 *Internal/external Interface Impacts and Specification*

The core architecture of this project was developed based on standard web application specifications. A web application is in general, consists of three components Front-End, Back-End and Application logic (middleware). In each level we are using a different set of tools and software which integrate and work together. For the front-end (client-side), we are using HTML and Bootstrap Web technologies; for the back end, Microsoft SQL Server, and middleware, we are using the ASP.NET Core Framework.

To build dynamic websites and web apps, ASP.NET, a server-side technology, is used. It is the latest version of the active server pages. A server-side scripting language and engine used to create web applications is called active server pages. For programmers, it is essentially a web-based platform that aids in the creation of dynamic web pages.

On the Internet, ASP.NET creates dynamic, interactive web applications. Web applications communicate through HTTP protocols and use HTTP commands to set communications between the server and the browser. Different programming languages, including C#, Visual Studio, and JavaScript, can be used to write the code. Microsoft provides ASP.NET, which was first released in 2002. It has the extension of.aspx. Language, library, and common language runtime are the three main building blocks of the ASP.NET architecture.

Whenever we build a web application, the first task is to communicate with the application using a programming language. We're utilizing C# as our programming language for this project. A library is a pre-written class or coded template a developer can use for developing an application. The most common library that ASP.NET uses is the WEB Library. Web Library possesses all the properties that help to build web applications. The Common Language Runtime is a platform that allows the execution of programs. ASP.NET provides state management, which helps store the state of an application.

Microsoft SQL Server Architecture

When a client sends a request to the database server, SQL Server accepts, processes, and responds to the request with the processed data. Three main parts make up the SQL Server architecture: Protocol Layer, Relational Engine, and Storage Engine.

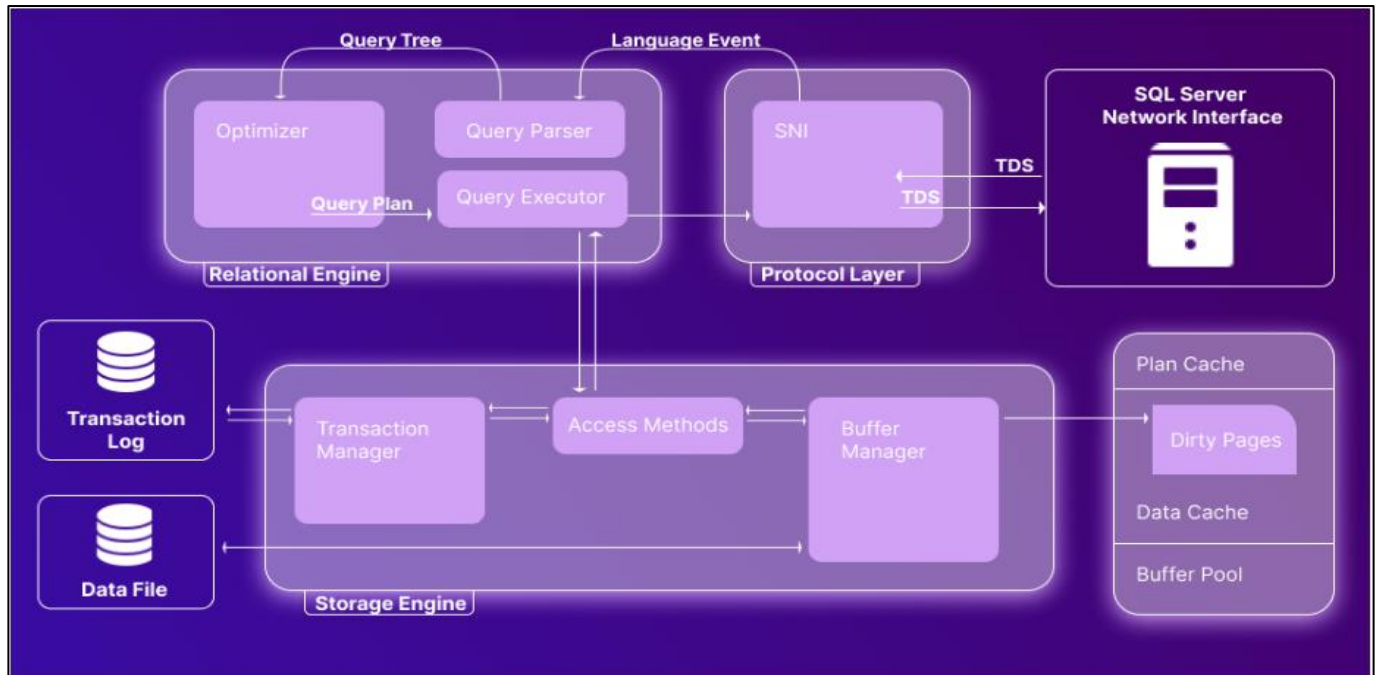


Figure 11: Microsoft SQL Server Architecture [5].

In our application we will provide the necessary connection strings to connect with the MS SQL Server Database and uses the entity core framework to store, retrieve, search, and delete the data.

```
appsettings.json  X
Schema: <No Schema Selected>
1  {
2    "ConnectionStrings": {
3      "MSRDbConnection": "Server=(localdb)\\mssqllocaldb;Database=MSR;Trusted_Connection=True;MultipleActiveResultSets=true"
4    },
5    "Logging": {
6      "LogLevel": {
7        "Default": "Information",
8        "Microsoft": "Warning",
9        "Microsoft.Hosting.Lifetime": "Information"
10     }
11   },
12   "AllowedHosts": "*"
13 }
14
```

Figure 12: Creating the connection in appsettings.json file.


```

Startup.cs  Startup(IConfiguration configuration)
17      }
18
19
20      2 references
21      public IConfiguration Configuration { get; }
22
23      // This method gets called by the runtime. Use this method to add services to the container.
24      0 references
25      public void ConfigureServices(IServiceCollection services)
26      {
27          services.AddDbContext<MSRDbContext>(options =>
28              options.UseSqlServer(
                Configuration.GetConnectionString("MSRDbConnection")));

```

Figure 13: Configuring the connection file in Startup class.

In this project we use Language-Integrated Query (LINQ) in the source code for retrieving and modifying data in Web pages [6]. To work with various data sources using a comparable coding style without prior knowledge of the data source's syntax, LINQ gives programmers the ability to query any form of data store (in this project it's MS SQL Server). The fact that LINQ offers intelligence and compile-time error checking, which is one of its key benefits.

Between the LINQ query and the real data source, there is a component called the LINQ provider, which transforms all LINQ queries into a format that is appropriate for the underlying data source. In our project, it converts a LINQ query to Transact-SQL, which MS SQL Server can understand.

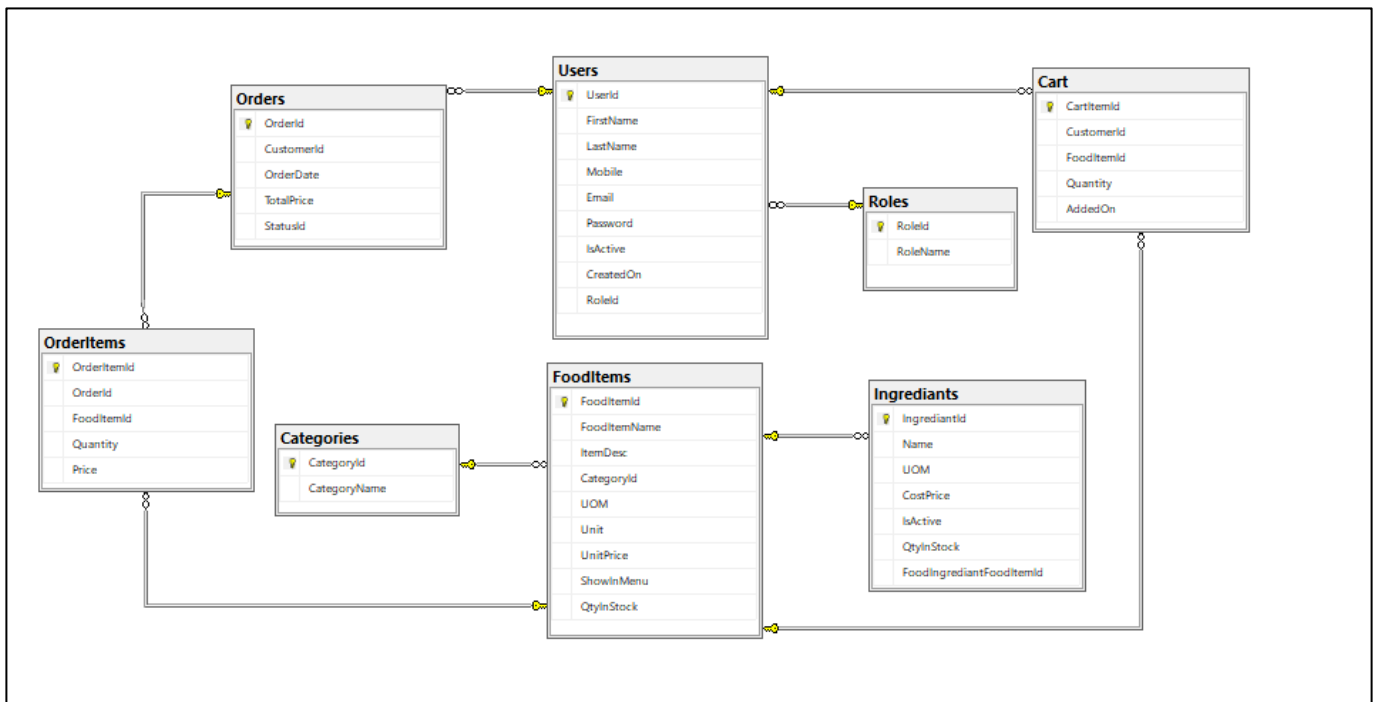


Figure 14: Database Design Diagram.

Front-End

We have used Bootstrap and HTML technologies for designing the user interface [7],[8]. UI is where end user will be first communicating with the application. For building responsive web apps, we used Bootstrap, which is a free, open-source HTML and CSS framework. One of the main advantages of using Bootstrap is that it supports responsive design. Using bootstrap, we can save a lot of development time by using predefined blocks of code suitable for the application's development.

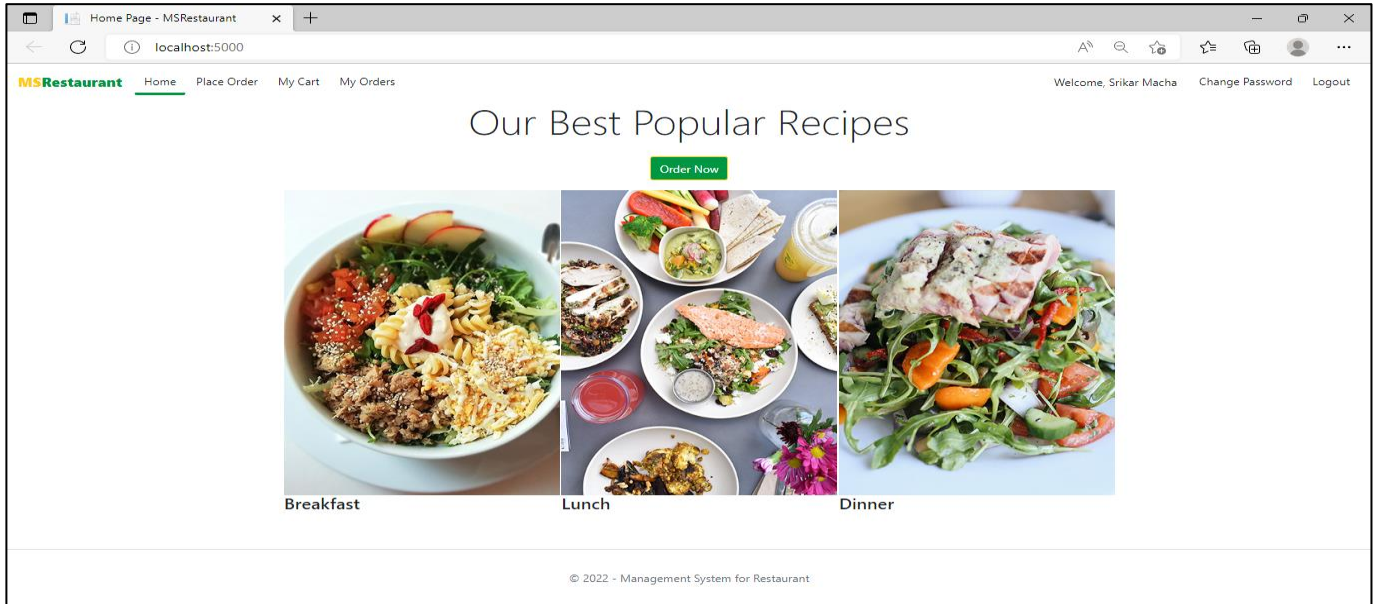


Figure 15: Application landing page with full screen window.

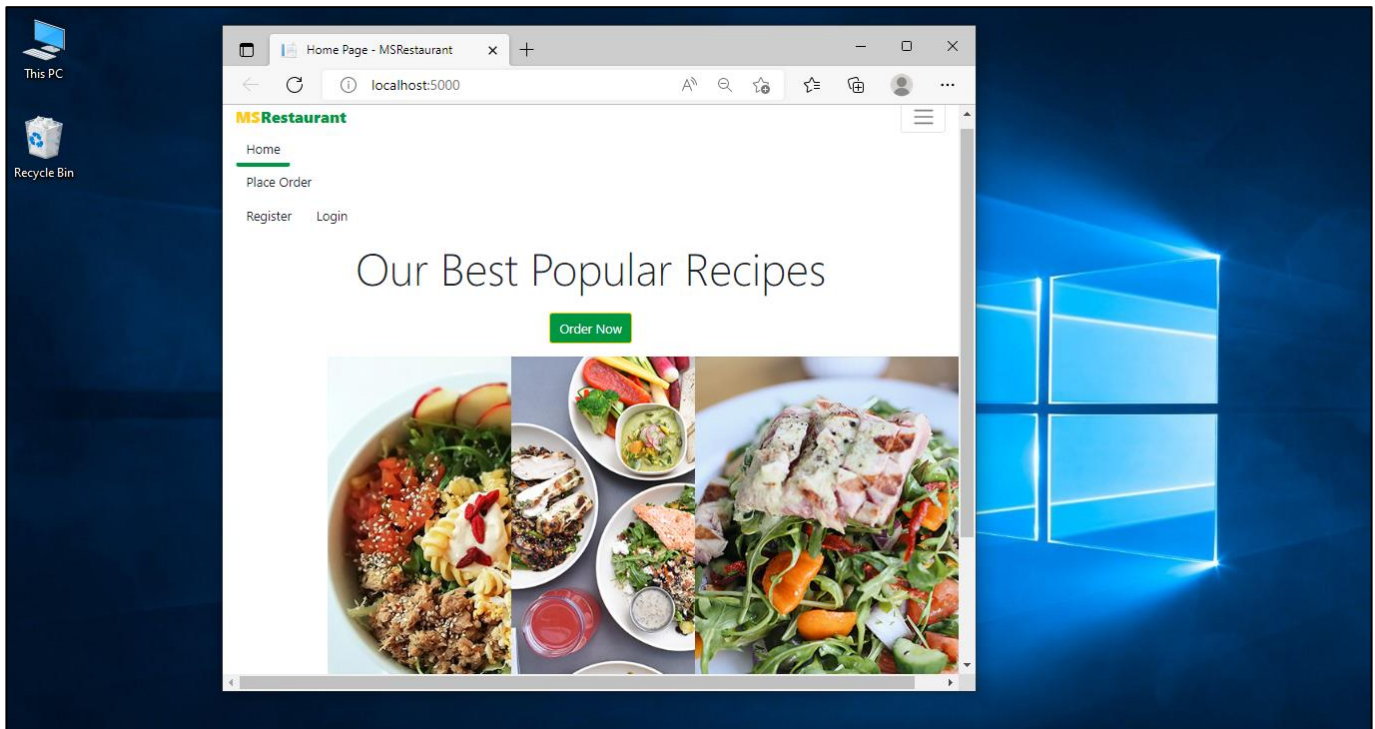


Figure 16: Application landing page with minimized window.

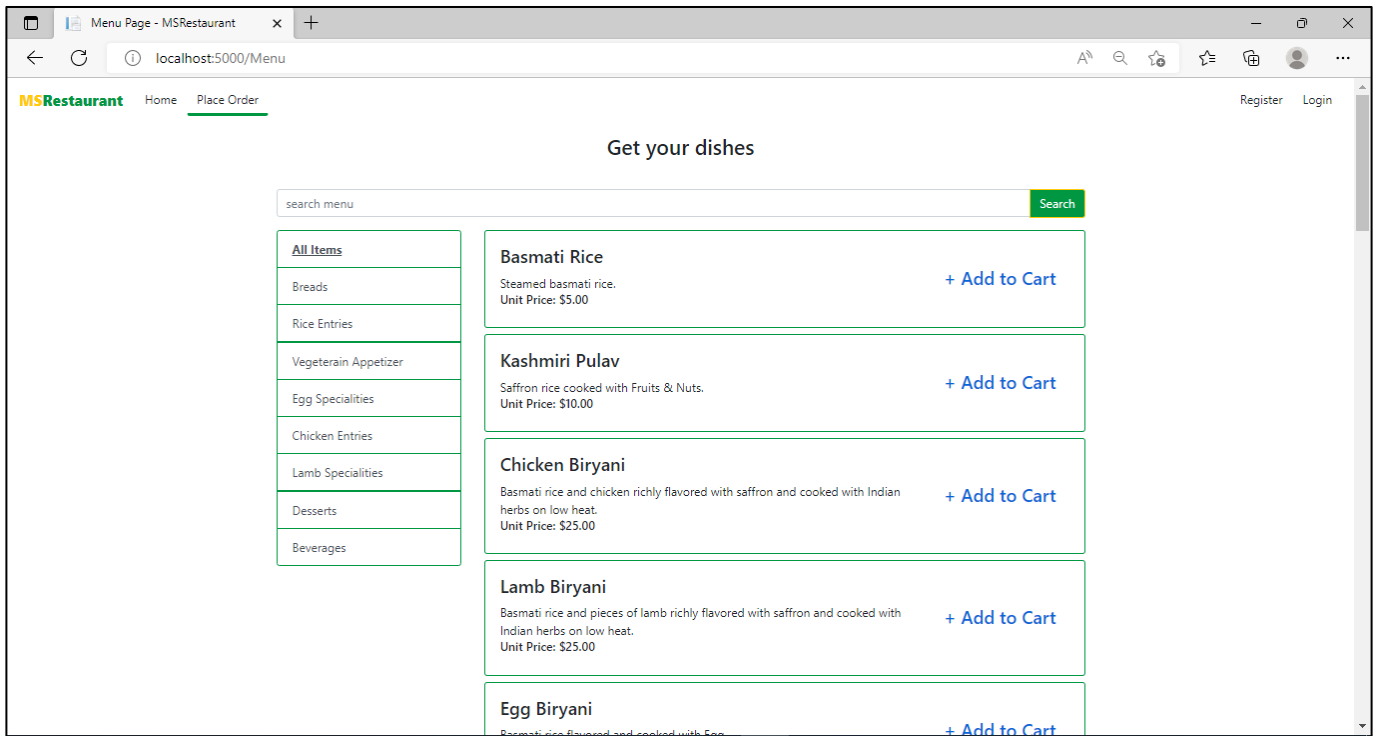


Figure 17: Order page of the Restaurant.

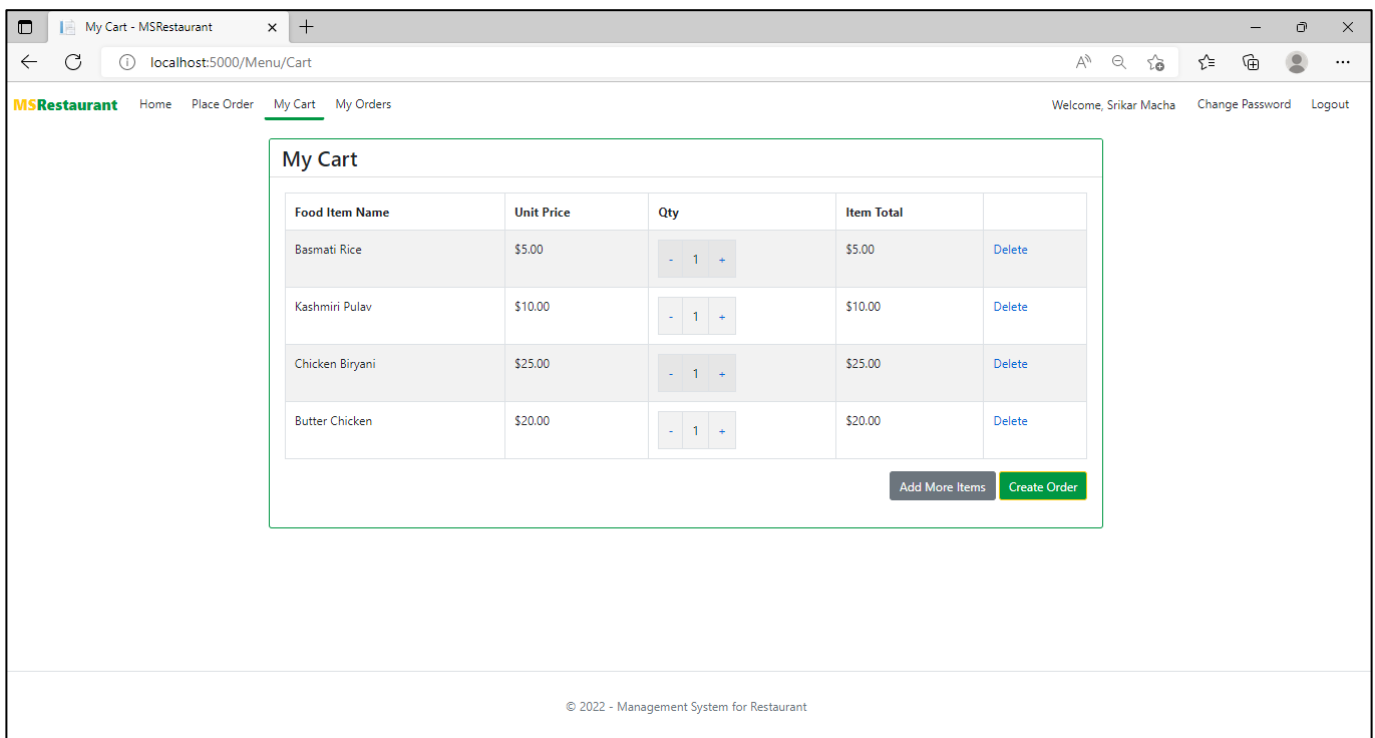


Figure 18: My Cart page of a customer.

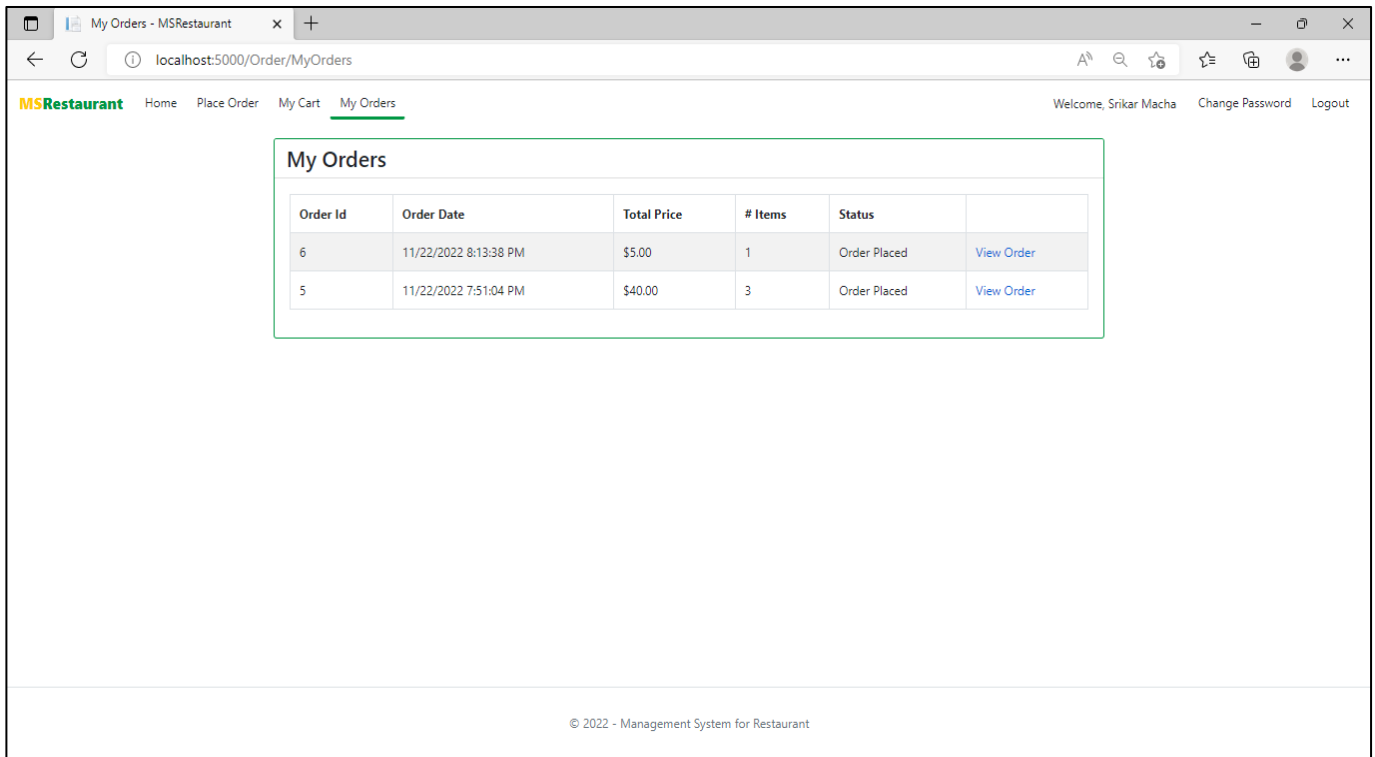


Figure 19: My Orders page of the admin.

Admin

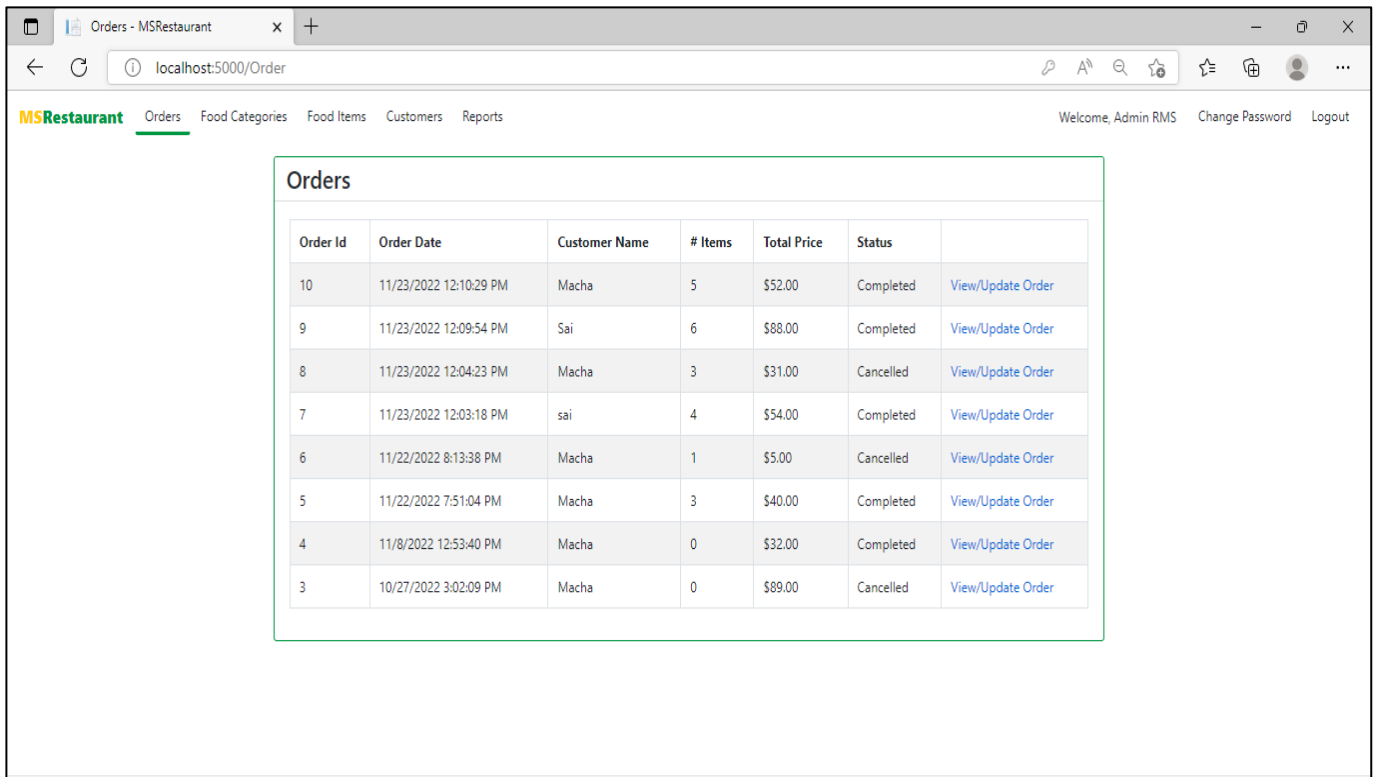


Figure 20: Orders User Interface of Admin's.

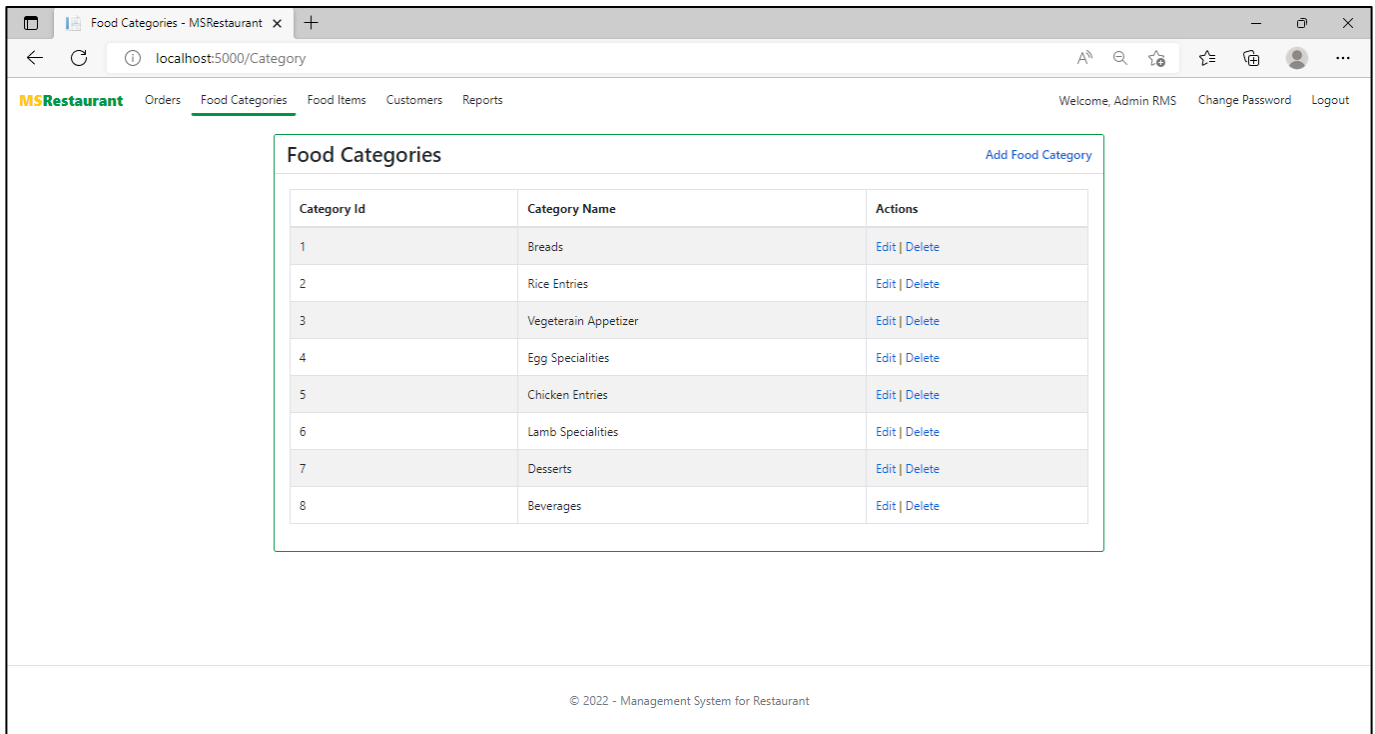


Figure 21: Food Categories User Interface.

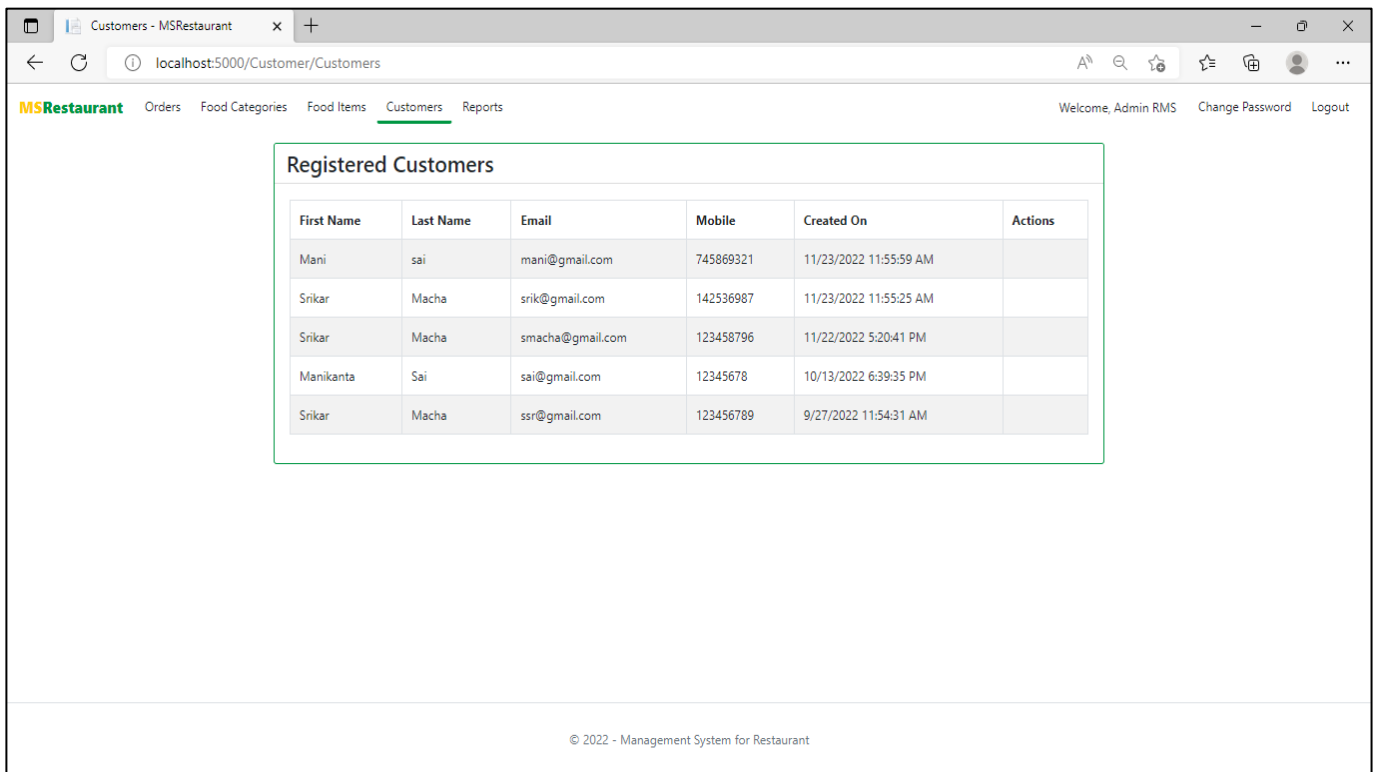


Figure 22: List of customers who had registered for the restaurant application.

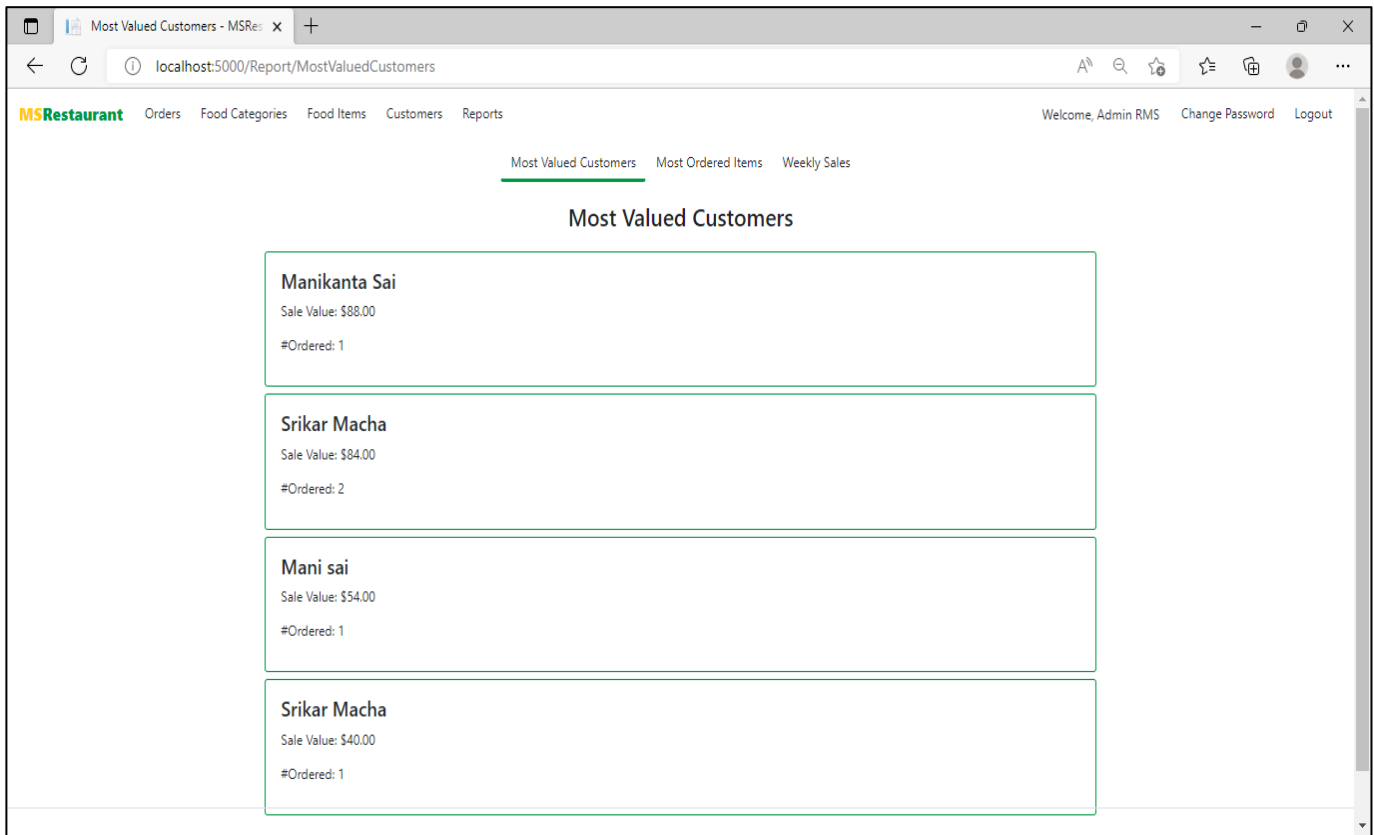


Figure 23: Reports page of admin.

6 Design Units Impacts

Design units are used for developing web applications are key areas which we need to focus on. If one of the units breaks, entire application flow will be affected, and the website functionality will not work as expected. Below are main design units implemented on the website:

- The admin will manage the food items and update the stock.
- Users search the food items and creates the order online.
- Admin will accept the food orders.

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

For developing a web application, we are focused on four areas: the front-end, the back end, the database, and the server. All elements of the user interface are part of the front end. Every single website that we see on the internet uses the same language to build the application. In this project, our web application requires a dynamic response from the server, so we need back-end development, where this part can't be seen by the users, and we use programming languages to help write functionalities that handle the business logic, like login in, browsing food items, etc. And to store all the website data, we need a relational database. Finally, we have a place to host our web application.

- *Front-End:* HTML5, CSS3, Bootstrap.
- *Back End:* C#
- *Database:* Microsoft SQL Server
- *Operating System:* Windows 10
- *Browser:* Microsoft Edge
- *Framework:* ASP.NET Core MVC, Entity Core Framework.
- *Integrated Development Environment (IDE):* Microsoft Visual Studio Community 2019
- *Tools:* SQL Server Management Studio (SSMS)

6.1.2 Impacts

All the functional requirements mentioned in the above section are used for the successful flow of business functionalities, and they do not impact the business logic or alter any scenarios. Software updates, upgrading the servers, managing the web traffic and transition the database from legacy to modern will be done by data engineers and application deployment team without interfering the source code and website live hosting.

7 Open Issues

All users embrace all forms of risk-free applications in the internet era. The online environment can be risky in numerous ways, including social media issues, online application fraud, cyberattacks, etc. Online applications of the management system for restaurants contain so many risks, like money transactions, privacy violations, and data handling. User personal data is gathered via applications created by a third party. When making a bill payment online, the consumers' sensitive information was stolen by these third-party websites. To reduce these risks' effects, the firms should have good and practical risk management strategies. This strategy ought to cover risk acceptance, risk minimization, avoidance, and liability.

8 Acknowledgements

We want to express our sincere gratitude to my department for giving me the good opportunity to work as a team and to work on the selected platforms for the completion of project. We are grateful to our professor, Dr. Yunchuan Liu for helping us choose and carry out the project. We completed our project successfully under our professor's guidance and the valuable weekly feedback is given to us. This project plays a vital role in our academic career.

9 References

- [1] ardalis. (n.d.). Overview of ASP.NET Core MVC. Learn.microsoft.com.
<https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-3.1>
- [2] ASP.NET Core MVC Tutorial for Beginners | .NET Core Tutorial. (n.d.). Pragim Tech. Retrieved December 5, 2022, from <https://www.pragimtech.com/courses/asp-net-core-mvc-tutorial-for-beginners/>
- [3] ajcvickers. (n.d.). Overview of Entity Framework Core - EF Core. Learn.microsoft.com.
<https://learn.microsoft.com/en-us/ef/core/>
- [4] Rick-Anderson. (n.d.). *Use cookie authentication without ASP.NET Core Identity*. Learn.microsoft.com. Retrieved December 5, 2022, from <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/cookie?view=aspnetcore-3.1>
- [5] (2022). Devart.com. <https://blog.devart.com/wp-content/uploads/2022/04/sql-server-architecture-6.png>
- [6] SQL server, .net and c# video tutorial: LINQ Tutorial. (n.d.). SQL Server, .net and C# Video Tutorial. Retrieved December 5, 2022, from <https://csharp-video-tutorials.blogspot.com/2014/07/linq-tutorial.html>
- [7] Otto, M. (2019). *Introduction*. Getbootstrap.com. <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
- [8] Bootstrap Tutorial - Tutorials point. (n.d.). [Www.tutorialspoint.com](https://www.tutorialspoint.com/).
<https://www.tutorialspoint.com/bootstrap/index.htm>