|  |  |  |
| --- | --- | --- |
| **Tech Saksham**  Final Project Report  **Full Stack Web Development** |  |  |

**Restaurant Management System**

**AP IIIT RGUKT RK VALLEY**

|  |  |
| --- | --- |
| **ROLL NO** | **NAME** |
| R170269 | Deepthi Sree Konduru |

**Poovaragavan**

**Master Trainer**

**ABSTRACT**

“Restaurant Management System” is a web-based portal for customers complaining about restaurant issues for restaurant. Where a customer can raise a ticket about an issue in restaurant and admin can assign a work for raised ticket to the employees. This portal is fully user controlled where a user can raise a complaint and user can see the status of the issue. The admin can be any registered user. He can control all the activities in the portal. He is responsible for assigning the complaints to employees that are raised by the customers. This portal is designed for the restaurant customers to complain about the issues related to restaurant. This portal is user friendly to the customers. We can conclude that our proposed model produced the highest accuracy while using .NET technology to build web-based portal to complain about issues in restaurant.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Table of Contents** | **Page No.** |
| 1 | Chapter 1: Introduction | 1 |
| 2 | Chapter 2: Services and Tools Required | 2-4 |
| 3 | Chapter 3: Project Architecture | 5 |
| 4 | Chapter 4: Architecture Blocks Detail Working | 6-8 |
| 5 | Chapter 5: Project Budget | 9 |
| 6 | Conclusion | 10 |
| 7 | References | 11 |
| 8 | Code | 12 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Overview**

The Overview of this document is to gather the requirements that are needed for implementing the Restaurant Management System. It also focuses on various key features, the product, product vision and scope, product overview. The main purpose of Restaurant Management System is to provide a platform to the Customers to complain about restaurant issues to the admin/manager

* 1. **Feature**

The product vision is to develop a Restaurant Management System, which is user friendly and easily accessible. The Restaurant Management System helps to complain about issues in restaurant to admin where admin can assign work to the employee.

* 1. **Advantages**
* A single user can raise multiple complaints and the admin can see all the complaints raised by different users so that he can assign work to that respected employee.
* Customer can see the status of the complaint in the customer module.
* The admin can see the issues raised by the customers.
* The admin can allot the work to the respected employee like plumber, electrician etc.
* The employee can check the work assigned to them individually.
* If the employee solves the problem, he can edit the progress of issues raised by the customers.
* The complaint will be solved in intime.
  1. **Scope**

The future implementation of this project is adding CSS (cascading style sheet) and deploying in the cloud to manage it. Based on this everyone can use this website by deploying in cloud. After adding CSS (cascading style sheet) the page will look attractive to use.

* 1. **Future Work**

The future implementation of this project is adding CSS (cascading style sheet) and deploying in the cloud to manage it. Based on this everyone can use this website by deploying in cloud. After adding CSS (cascading style sheet) the page will look attractive to use.

**CHAPTER 2**

**SERVICES AND TOOLS REQUIRED**

**2.1 Services Used**

**2.1.1 Liberty Profile**

A flexible server profile of IBM ‘s WebSphere Application Server (WAS) which enables the server to deploy only required custom features rather than deploying all available components.

**2.2 Tools and Software used**

**Tools:**

1.ASP.NET CORE MVC

2.HTML

3.SQL

**Software:**

1.Microsoft visual studio 2022

2.Sql Server Management-SQL 2019-SSEI-Expr 9

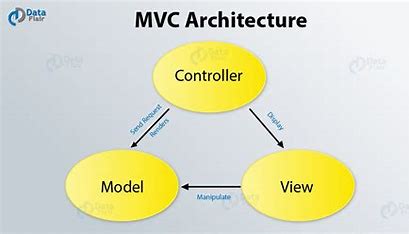
3.SSMS-SETUP-EMU

**2.2.1 ASP.NET**

**ASP.NET CORE MVC:** ASP.NET CORE MVC is a rich framework for building web apps and APIs using the Model-View-Controller design pattern.

**MVC Pattern**: The Model-View-Controller (MVC) architectural pattern separates an application into three main groups of components: Models, Views, and Controllers. This pattern helps to achieve separation of concerns. Using this pattern, user requests are routed to a Controller which is responsible for working with the Model to perform user actions and/or retrieve results of queries. The Controller chooses the View to display to the user, and provides it with any Model data it requires.

The following diagram shows the three main components and which one’s reference the others

****

**2.2.2 HTML5**

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

**2.2.3 Cloud Foundry**

**SQL:**

· SQL stands for Structured Query Language

· SQL lets you access and manipulate databases

· SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

**The SQL UPDATE Statement:** The UPDATE statement is used to modify the existing records in a table. UPDATE table name SET column1 = value1, column2 = value2, ... WHERE condition; **The SQL SELECT Statement:** The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set. SELECT column1, column2, FROM table name; **SQL CREATE DATABASE Statement:** The CREATE DATABASE statement is used to create a new SQL database. CREATE DATABASE name.

**CHAPTER 3**

**PROJECT ARCHITECTURE**

**3.1 Architecture**

**USER FRONTEND BACKEND**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **HTML 5** | **ASP.NET** |  |
|  |  | **SQL** |  |
|  |  |  |  |

**CHAPTER 4**

**ARCHITECTURE BLOCKS DETAIL WORKING**

**4.1 Blocks**

**4.1.1 User:**

* The user will register using the registration form.
* After register user will login to the customer module.
* After login user can compliant about the issue using the create new request link.
* User can see the list of raised by them.
* User can edit or delete the particular request.
* User can see the request and assigned employee

**4.1.2 Frontend:**

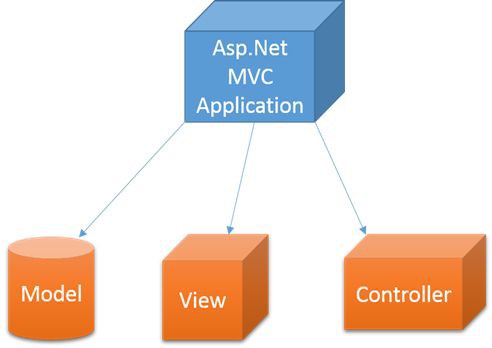
**HTML5**

**HTML5** is a [markup language](https://en.wikipedia.org/wiki/Markup_language) used for structuring and presenting content on the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). It is the fifth and final[[3]](https://en.wikipedia.org/wiki/HTML5#cite_note-W3C_transfer_ZDNet-3) major [HTML](https://en.wikipedia.org/wiki/HTML) version that is a [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C) recommendation. The current specification is known as the [HTML Living Standard](https://en.wikipedia.org/wiki/HTML_Living_Standard). It is maintained by the [Web Hypertext Application Technology Working Group](https://en.wikipedia.org/wiki/Web_Hypertext_Application_Technology_Working_Group) (WHATWG), a consortium of the major browser vendors ([Apple](https://en.wikipedia.org/wiki/Apple_Inc.), [Google](https://en.wikipedia.org/wiki/Google), [Mozilla](https://en.wikipedia.org/wiki/Mozilla), and [Microsoft](https://en.wikipedia.org/wiki/Microsoft)).

The primary thing to keep in mind, the super magic key, is that HTML is used for **meaning** and CSS is used for **presentation**. HTML is nothing more than fancy structured content and the visual formatting of that content will come later when we tackle CSS.

**4.1.3 Backend:**

**ASP.NET**



**Model Responsibilities:** The Model in an MVC application represents the state of the application and any business logic or operations that should be performed by it. Business logic should be encapsulated in the model, along with any implementation logic for persisting the state of the application. Strongly-typed views typically use View Model types designed to contain the data to display on that view. The controller creates and populates these View Model instances from the model.

**View Responsibilities:** Views are responsible for presenting content through the user interface. They use the Razor view engine to embed .NET code in HTML markup. There should be minimal logic within 10 views, and any logic in them should relate to presenting content. If you find the need to perform a great deal of logic in view files in order to display data from a complex model, consider using a View Component, View Model, or view template to simplify the view.

**Controller Responsibilities**: Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. In the MVC pattern, the controller is the initial entry point, and is responsible for selecting which model types to work with and which view to render (hence its name - it controls how the app responds to a given request).

**SQL:**

· SQL stands for Structured Query Language

· SQL lets you access and manipulate databases

· SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

**The SQL UPDATE Statement:** The UPDATE statement is used to modify the existing records in a table. UPDATE table name SET column1 = value1, column2 = value2, ... WHERE condition;

**The SQL SELECT Statement:** The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set. SELECT column1, column2, ...

FROM table name;

**SQL CREATE DATABASE Statement:** The CREATE DATABASE statement is used to create a new SQL database. CREATE DATABASE name

**CHAPTER 5**

**PROJECT BUDGET**

We collaboratively developed this project without any cost.

**CONCLUSION**

The Restaurant Management System helps your organization improve workforce productivity and boost overall well-being by assigning work to the particular employee and user can easily raise an issue to solve. The Restaurant Management System is developed in order to computerize the activities which take more time, if done manually. This study takes similar route, but with an improved and novel method and by using this portal the user can save their productive time. This website provides more accuracy to solve the problems within the time. The Restaurant Management System is used to the customer raise the problems by using the website and admin can assign or delete the work raised by customer by using employee id and employee can see the work assigned by admin and edit the status of employee.

**REFERENCES**

[HTML Tutorials | HTML Dog](https://htmldog.com/guides/html/)

https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0#modelbinding https://youtu.be/4IgC2Q5-yDE -Ku Venkat channel <https://www.w3schools.com/sql/>

<https://geekforgeeks.com>

<https://javatpoint.com>

**CODE**

[JNANA-PRASANNA-PURINI/Majorproject (github.com)](https://github.com/JNANA-PRASANNA-PURINI/Majorproject)