

Development of AI Based Diet Consultant Application



Mini Project Member (s)

Sl. No.	Reg. No.	Student Name
1	19ETCS002083	Pasupuleti Rohithsaidatta
2	19ETCS002152	Rakesh Kumar R
3	19ETCS002081	Nithin Krishna M
4	19ETCS002066	Jayendra M
5	19ETCS002098	Rakshith G

Supervisors:1. Dr. Rinki Sharma

DEC– 2022

B. Tech. in Computer Science and Engineering

FACULTY OF ENGINEERING AND TECHNOLOGY
M. S. RAMAIAH UNIVERSITY OF APPLIED SCIENCES
Bengaluru -560058

FACULTY OF ENGINEERING AND TECHNOLOGY



Certificate

*This is to certify that the Mini Project titled “**Development of AI Based Diet Consultant application**” is a bonafide work carried out in the **Department of Computer Science and Engineering** by Mr Pasupuleti Rohithsaidatta, Rakesh kumar R, Nithin Krishna M, Jayendra M, Rakshith G bearing Reg. No.*

19ETCS002083,19ETCS002152,19ETCS002081,19ETCS002066,19ETCS002098 in partial fulfilment of requirements of the Course curriculum of 7th Sem Computer Science and Engineering of Ramaiah University of Applied Sciences.

Nov – 2022

Name of Mentor: Dr.Rinki Sharma

Designation: Assistant Professor- Dept. of CSE

Place: M. S. RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Declaration

Development of AI Based Diet Consultant Application

The project work is submitted in partial fulfilment of academic requirements for the award of **B. Tech.** Degree in the **Department of Computer Science and Engineering** of the Faculty of **Engineering and Technology** of Ramaiah University of Applied Sciences. The project report submitted herewith is a result of our own work and in conformance to the guidelines on plagiarism as laid out in the University Student Handbook. All sections of the text and results which have been obtained from other sources are fully referenced. We understand that cheating and plagiarism constitute a breach of university regulations, hence this project report has been passed through plagiarism check and the report has been submitted to the supervisor.

Sl. No.	Reg. No.	Student Name	Signature
1	19ETCS002083	PASUPULETI ROHITH SAI DATTA	<i>Rohithsaidatta p</i>
2	19ETCS002152	RAKESH KUMAR R	<i>Rakesh Kumar R</i>
3	19ETCS002081	NITHIN KRISHNA M	<i>Nithin Krishna M</i>
4	19ETCS002066	JAYENDRA M	<i>Jayendra M</i>
5	19ETCS002098	RAKSHITH G	<i>Rakshith g</i>

Date: DECEMBER 2022

Development of AI Based Diet Consultant application

Table of contents

Contents	pageno
• Introduction to Development of AI diet consultant application	1
• Modules and their Description	1
• Existing System & Proposed System	3
• Project Lifecycle Details	4
• Project Design	4
• PROJECT IMPLEMENTATION	9
• OVERVIEW OF TECHNOLOGIES USED	9
a. Frontend Technology	9
b. Backend Technology	15
c. Middle ware Technology	18
• Code	24
• Results	28
• Features	33
• Conclusions	33
• References	33

➤ Acknowledgement

- We are pleased to present **“Development of AI Based Diet Consultant application”** project and take this opportunity to express our profound gratitude to all those people who helped us in completion of this project.
- We thank our college for providing us with excellent facilities that helped us to complete and present this project.
- We express our deepest gratitude towards our project guide **Dr. Rinki Sharma** for her valuable and timely advice during the various phases in our project. We would like to thank her for support, patience and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules.



➤ 1. Introduction to Development of AI diet consultant application

1.1 Introduction

Just similar to a human dietician, this web system will also act like your dietician. When you go to a doctor of nutrition, then she will ask you your personal details related to body and health such as your age, your height, your weight and how much water do your consumer in a day and how much walk to do take regularly and how much work do you do regularly. Just similar to this doctor, this artificial intelligent dietitian also asks you similar questions in your device and you have to answer all those questions and then this AI consultant will also advice you about what should your intake in your diet and what should you ignore in order to keep yourself healthy via your diet.

➤ Modules and their Description

This website is having 3 major modules with its sub-modules as follows:

1. Admin Login

- **Add Dietitian:**
 - System allows admin to add a dietitian details into the system who can create the diet plan for the users.
- **View/Edit Dietitian:**
 - Can view and edit the dietitian details whenever required.
- **View User:**
 - Can view the list of registered users with their details.
- **View Feedback**
 - Can view feedback provided by the registered users.

2. User Login

- **Register:**
 - To continue with the diet plan details, user first need to fill up all the required details.
- **Calculate BMI:**
 - Based on details provided by the user, system automatically calculate the BMI of the user.
- **View Diet Plan:**
 - The diet plan for the user is generated by the system itself using artificial intelligence.
- **Request Diet Plan:**
 - If the user is unsatisfied with the diet plan provided by the system, the he/she can raise a request to generate the proper diet plan him/herself.
 - The diet plan request is forwarded to dietitian.
- **View Profile:**
 - User can view his/her own profile.
- **Write Feedback:**
 - Registered users can provide the feedback.

3. Dietitian Login

- **View Diet Plan Request:**
 - Here, dietitian can view the diet plan request from the users.
- **Create Diet Plan:**
 - Based on user's request, dietitian creates a diet plan for him/her.
- **Update Diet Plan:**
 - The user resends the request to dietitian about unsatisfied diet plan. So, as per the user's request, dietitian regenerates the diet plan and sends to the user.
- **View User:**
 - Can view user details as a when needed.

➤ Existing System & Proposed System

■ Problem with current scenario

In the existing system, you have to hire a dietician in order to get advice. Hiring a nutrition doctor will not only waste your time and efforts for calling them, going to them and so on but also cost you very high as their charges per month are very high. The moment will also arrive when they will not be available for you and you have to search for some other dietician urgently.

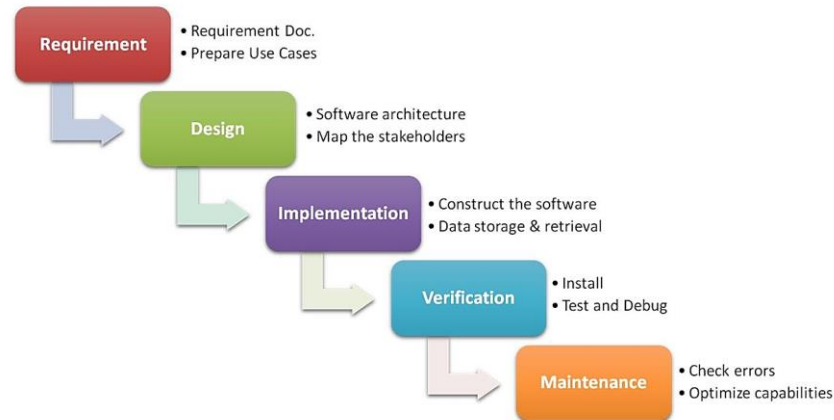
- I. Traditionally, one has to travel for consulting a Dietitian in his/her clinic which consumes time as well as it proves frenetic in transportation.
- II. Implementing such a system in the real world will reduce the efforts of travelling on busy roads and will save time.
- III. No such system has been developed for counselling.
- IV. Clinics are being overcrowded sometimes and the person is left dejected when the query is delayed to the next day just because of finished visiting hours.
- V. Pressure mounts on the human Dietitians when the number of waiting patients increases.

■ Proposed System

- I. Considering the anomalies in the existing system, computerization of the whole activity is being suggested after initial analysis.
- II. **“Artificial Intelligence Dietitian”** is a BOT with artificial intelligence about human diets.
- III. It acts as a diet consultant similar to a real Dietitian.
- IV. User may login and view various diet information.
- V. A Dietitian consults a person based on his schedule, body type, height and weight. The system too asks all this data from the user and processes it.
- VI. It asks about how many hours the user works, his height, weight, age etc.
- VII. It stores and processes the above data and then calculates the nutrient value needed to fill up user's needs.
- VIII. It then shows an appropriate diet to the users and asks if the user is ok with it, else it shows other alternate diets to fill up user's needs.

➤ Project Lifecycle Details

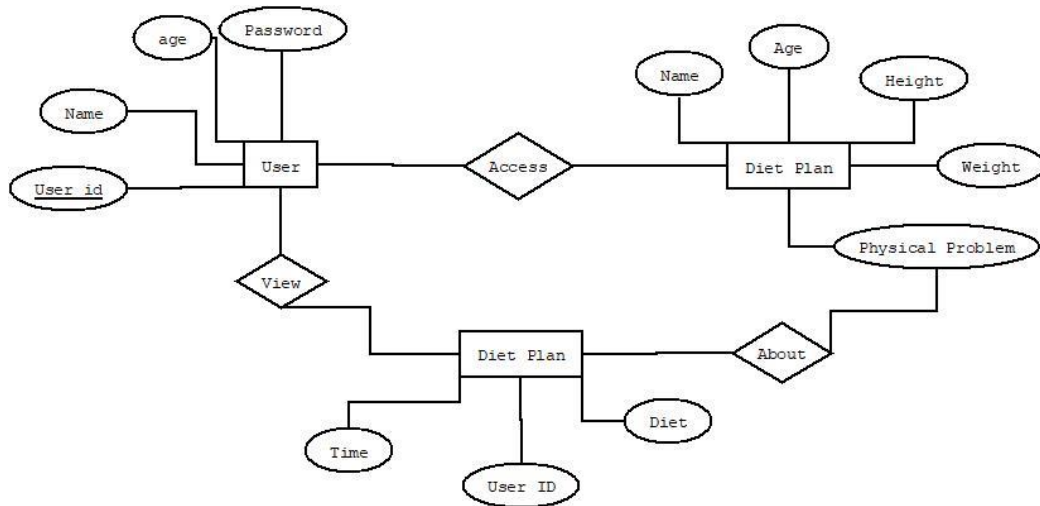
● Waterfall Model



The waterfall Model is a linear sequential flow. In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. The waterfall approach is the earliest approach that was used for software development.

➤ Project Design

● E-R Diagram



● Use Case Diagram

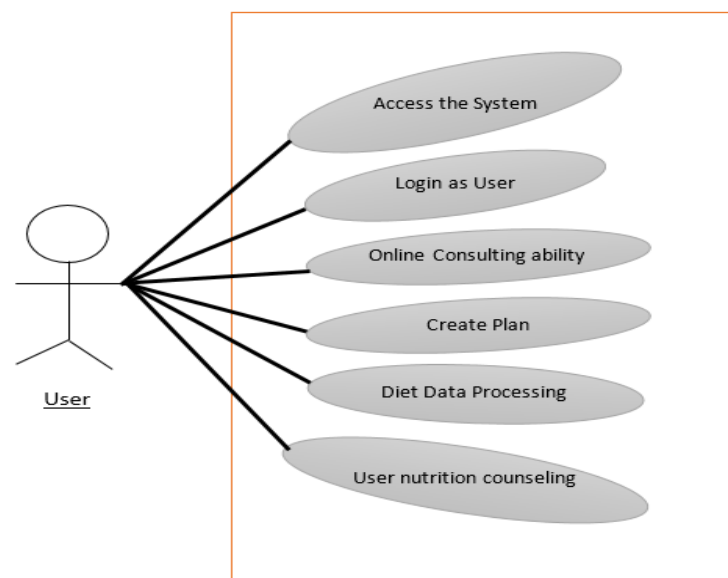


Fig. User Case Diagram for User

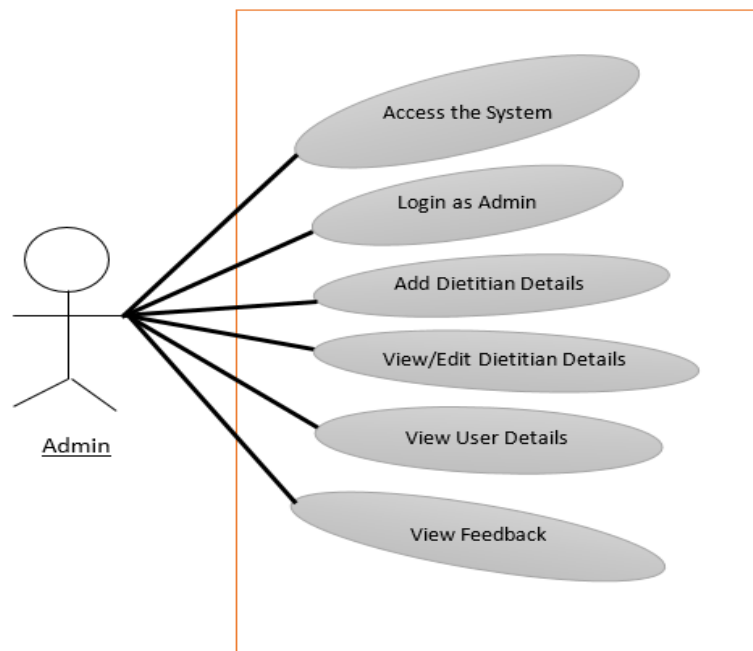


Fig. User Case Diagram for Admin

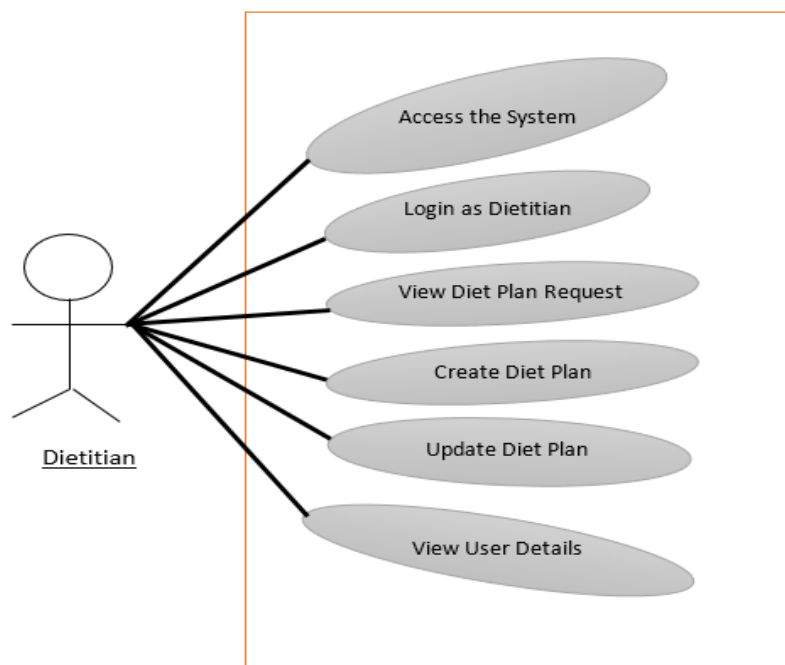


Fig. User Case Diagram for Dietitian

➤ **Data Flow Diagram**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. . The development of DFD's is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

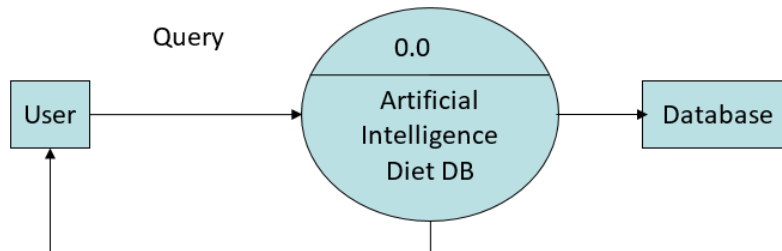
➤ **CONSTRUCTING A DFD**

Several rules of thumb are used in drawing DFD's:

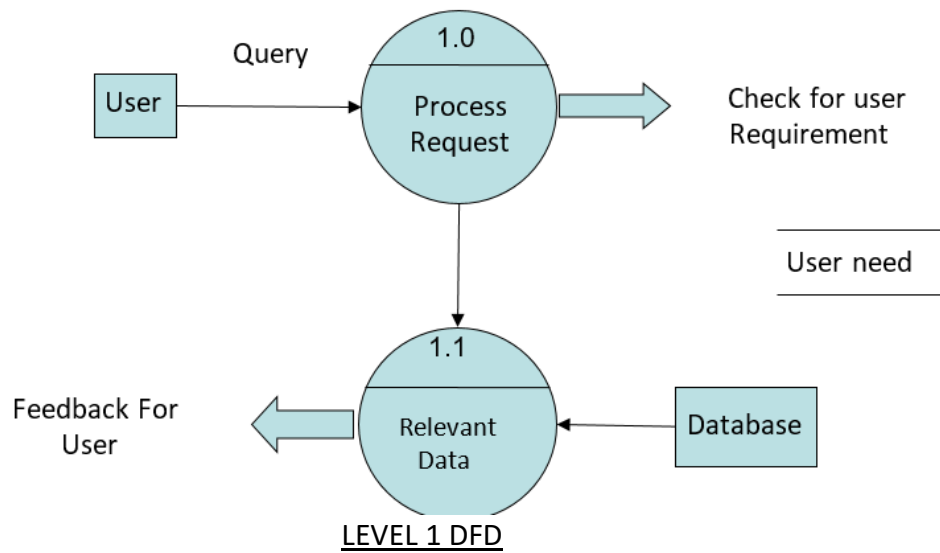
1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized

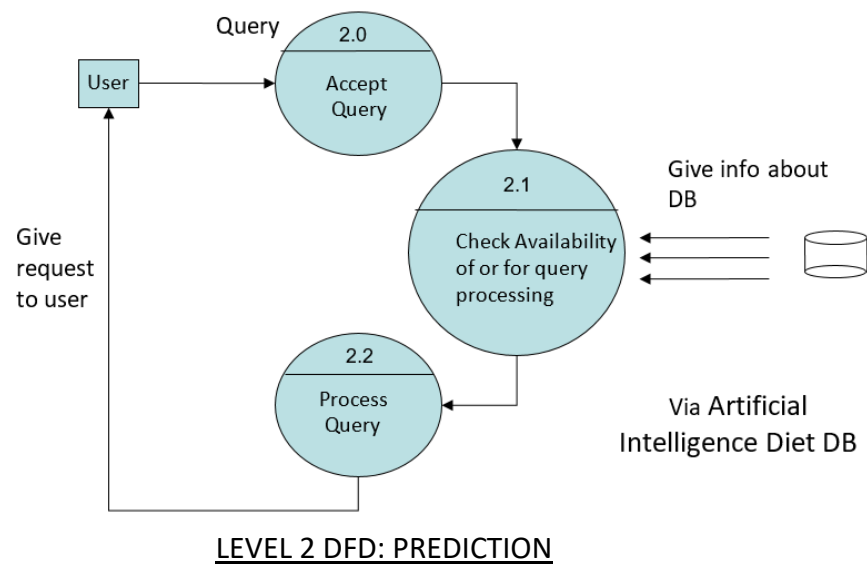
A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

➤ Data Flow Diagrams

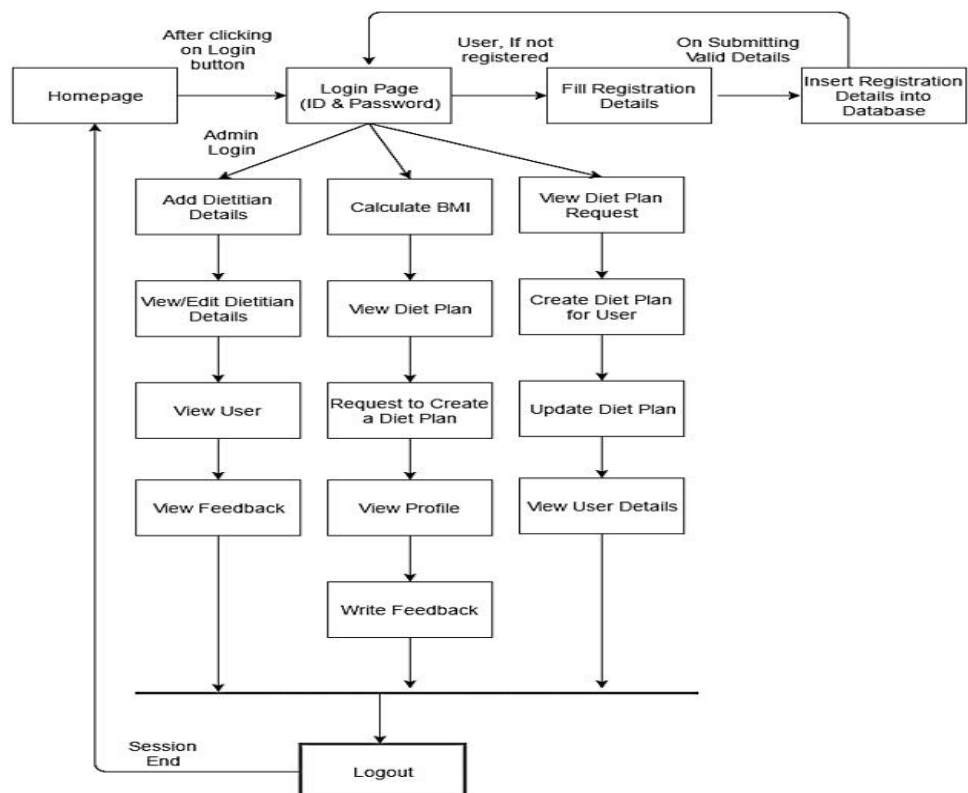


➤ DATABASE DETAIL





➤ SYSTEM ARCHITECTURE



➤ **PROJECT IMPLEMENTATION**

- Project Implementation Technology

The Project is loaded in Visual Studio 2010. We used Visual Studio for Design and coding of project. Created and maintained all databases into SQL Server 2008, in that we create tables, write query for store data or record of project.

- **Hardware Requirement:**

- I. i3 Processor Based Computer
- II. 1GB-Ram
- III. 5 GB Hard Disk
- IV. Internet Connection

- **Software Requirement:**

- I. Windows 7 or higher
- II. Visual Studio 2010
- III. SQL Server 2008

➤ **OVERVIEW OF TECHNOLOGIES USED**

Front End Technology

Microsoft .Net Framework

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remote, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. The class library, the other main component of the .NET Framework, is a

comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.

➤ **Features of the Common Language Runtime**

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and

third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language,

Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

➤ **.NET Framework Class Library**

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access.

In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

- **Console applications.**
- **Scripted or hosted applications.**
- **Windows GUI applications (Windows Forms).**
- **ASP.NET applications.**
- **XML Web services.**

- **Windows services.**

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

➤ **Client Application Development**

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®.

The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

Unlike ActiveX controls, Windows Forms controls have semi-trusted access to a user's computer. This means that binary or natively executing code can access some of the resources on the user's system (such as GUI elements and limited file access) without being able to access or compromise other resources.

Because of code access security, many applications that once needed to be installed on a user's system can now be safely deployed through the Web. Your applications can implement the features of a local application while being deployed like a Web page.

➤ **Server Application Development**

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

➤ **Server-side managed code**

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites.

However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft

Enhanced Performance: ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.

Power and Flexibility: Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web.

ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

Manageability: ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools.

This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

➤ **Crystal Reports**

Crystal Reports for Visual Basic .NET is the standard reporting tool for Visual Basic.NET; it brings the ability to create interactive, presentation-quality content — which has been the strength of Crystal Reports for years — to the .NET platform.

With Crystal Reports for Visual Studio .NET, you can quickly create complex and professional-looking reports. Instead of coding, you use the Crystal Report Designer interface to create and format the report you need. The powerful Report Engine processes the formatting, grouping, and charting criteria you specify.

➤ **BACK END TECHNOLOGY**

About Microsoft SQL Server

Microsoft SQL Server is a Structured Query Language (SQL) based, client/server relational database. Each of these terms describes a fundamental part of the architecture of SQL Server.

Database

A database is similar to a data file in that it is a storage place for data. Like a data file, a database does not present information directly to a user; the user runs an application that accesses data from the database and presents it to the user in an understandable format.

Client/Server: -

In a client/server system, the server is a relatively large computer in a central location that manages a resource used by many people. When individuals need to use the resource, they connect over the network from their computers, or clients, to the server.

Examples of servers are: In a client/server database architecture, the database files and DBMS software reside on a server. A communications component is provided so applications can run on separate clients and communicate to the database server over a network. The SQL Server communication component also

allows communication between an application running on the server and SQL Server.

Server applications are usually capable of working with several clients at the same time. SQL Server can work with thousands of client applications simultaneously. The server has features to prevent the logical problems that occur if a user tries to read or modify data currently being used by others.

While SQL Server is designed to work as a server in a client/server network, it is also capable of working as a stand-alone database directly on the client. The scalability and ease-of-use features of SQL Server allow it to work efficiently on a client without consuming too many resources.

Structured Query Language (SQL)

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational databases; the most common is SQL.

Both the American National Standards Institute (ANSI) and the International Standards Organization (ISO) have defined standards for SQL. Most modern DBMS products support the Entry Level of SQL-92, the latest SQL standard (published in 1992).

Databases

A database in Microsoft SQL Server consists of a collection of tables that contain data, and other objects, such as views, indexes, stored procedures, and triggers, defined to support activities performed with the data.

The data stored in a database is usually related to a particular subject or process, such as inventory information for a manufacturing warehouse.

SQL Server can support many databases, and each database can store either interrelated data or data unrelated to that in the other databases. For example, a server can have one database that stores personnel data and another

that stores product-related data. Alternatively, one database can store current customer order data, and another; related database can store historical customer orders that are used for yearly reporting. Before you create a database, it is important to understand the parts of a database and how to design these parts to ensure that the database performs well after it is implemented.

➤ **Middleware Technology**

Active Data Objects.Net Overview

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind.

ADO.NET uses some ADO objects, such as the Connection and Command objects, and also introduces new objects. Key new ADO.NET objects include the Dataset, Data Reader, and Data Adapter.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object -- the Dataset -- that is separate and distinct from any data stores. Because of that, the Dataset functions as a standalone entity. You can think of the Dataset as an always disconnected record set that knows nothing about the source or destination of the data it contains. Inside a Dataset, much like in a database, there are tables, columns, relationships, constraints, views, and so forth.

While the Dataset has no knowledge of the source of its data, the managed provider has detailed and specific information. The role of the managed provider is to connect, fill, and persist the Dataset to and from data

stores. The OLE DB and SQL Server .NET Data Providers (System.Data.OleDb and System.Data.SqlClient) that are part of the .Net Framework provide four basic objects: the Command, Connection, Data Reader and Data Adapter. In the remaining sections of this document, we'll walk through each part of the Dataset and the OLE DB/SQL Server .NET Data Providers explaining what they are, and how to program against them.

The following sections will introduce you to some objects that have evolved, and some that are new. These objects are:

- **Connections.** For connection to and managing transactions against a database.
- **Commands.** For issuing SQL commands against a database.
- **Data Readers.** For reading a forward-only stream of data records from a SQL Server data source.
- **Datasets.** For storing, removing and programming against flat data, XML data and relational data.
- **Data Adapters.** For pushing data into a Dataset, and reconciling data against a database.

➤ TESTING

BLACK BOX TESTING

In clearing house across various modules this testing was performed to check the following.

1. Establishing communication with the database for handling request and response.
2. Verification of OLE-DB providers(ADO) in functionality
3. Parameters passing and report generation used from the application with crystal report.

WHITE BOX TESTING

All the statements included in the code across various modules were tested to find none of the statements were overlooked or skipped from execution. This enabled isolating of errors that would have otherwise occurred and would have resulted in abnormal terminal or exceptions thrown. The test was deeply tested in patient and responsibility, insured party, ailments, procedures and applied payment modules.

STRING TESTING

The applications was tested for inputs pertaining to patient data, responsible party, insured party for strings such as name, relation, employ information, policy details, insurance company details, claim center information and attorney data physician, reference physician information were tested for the following

- i. null data
- ii. string length
- iii. data format
- iv. alpha numeric characters

In addition, numeric inputs were tested for invalid characters, invalid data format, size of the input data and the data type being handled.

As the project is on bit large scale, we always need testing to make it successful. If each components work properly in all respect and gives desired output for all kind of inputs then project is said to be successful. So the conclusion is-to make the project successful, it needs to be tested.

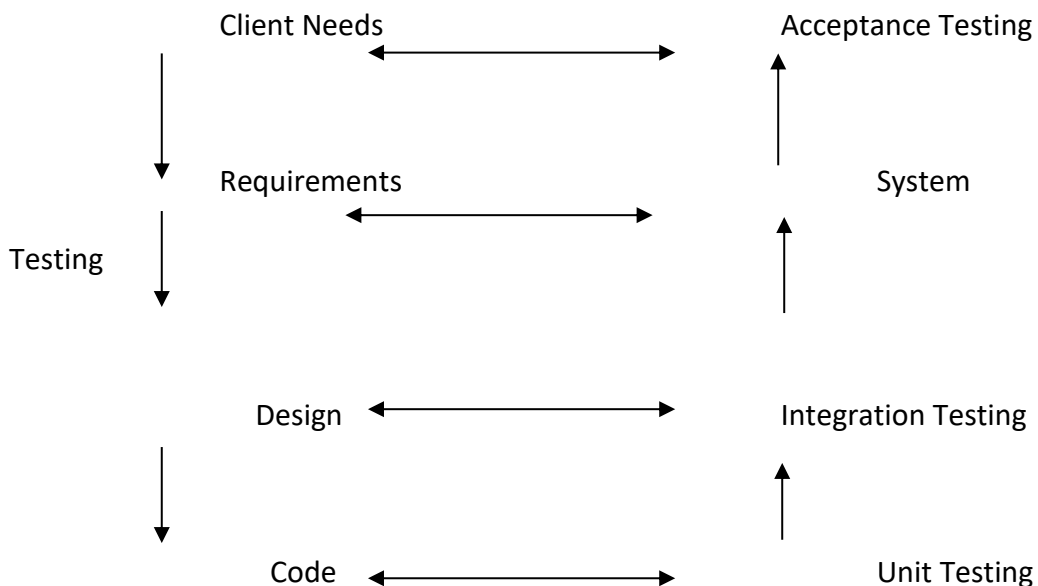
The testing done here was System Testing checking whether the user requirements were satisfied. The code for the new system has been written completely using ASP .NET with C# as the coding language, C# as the interface for front-end designing. The new system has been tested well with the help of

the users and all the applications have been verified from every nook and corner of the user.

Although some applications were found to be erroneous these applications have been corrected before being implemented. The flow of the forms has been found to be very much in accordance with the actual flow of data.

Levels of Testing

In order to uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are:



A series of testing is done for the proposed system before the system is ready for the user acceptance testing.

The steps involved in Testing are:

✓ **Unit Testing**

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as “Module Testing”. The modules are tested separately. This testing carried out during programming

stage itself. In this testing each module is found to be working satisfactorily as regards to the expected output from the module.

✓ **Integration Testing**

Data can be grossed across an interface; one module can have adverse efforts on another. Integration testing is systematic testing for construction the program structure while at the same time conducting tests to uncover errors associated with in the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the isolation of cause is complicate by the vast expense of the entire program. Thus in the integration testing stop, all the errors uncovered are corrected for the text testing steps.

✓ **System testing**

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then goal will be successfully achieved.

✓ **Validation Testing**

At the conclusion of integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins, validation test begins. Validation test can be defined in many ways. But the simple definition is that validation succeeds when the software function in a manner that can reasonably expected by the customer. After validation test has been conducted one of two possible conditions exists.

One is the function or performance characteristics confirm to specifications and are accepted and the other is deviation from specification is

uncovered and a deficiency list is created. Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

✓ **Output Testing**

After performing validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated by the system under consideration. Here the output format is considered in two ways, one is on the screen and other is the printed format. The output format on the screen is found to be correct as the format was designed in the system designed phase according to the user needs.

For the hard copy also the output comes as the specified requirements by the users. Hence output testing does not result any corrections in the system.

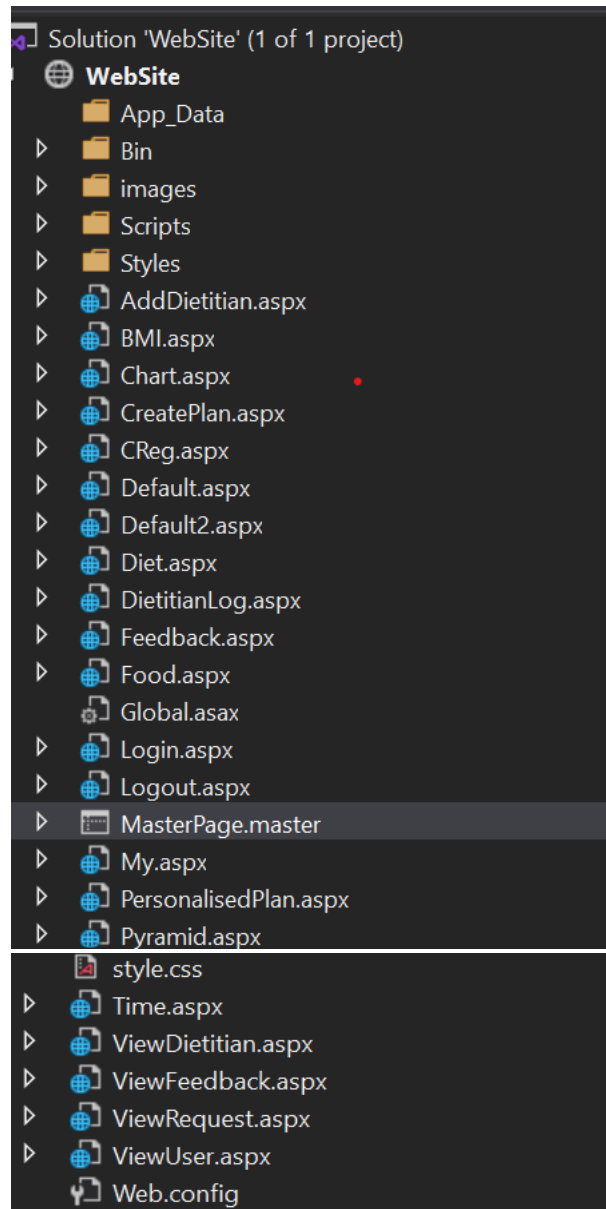
✓ User Acceptance Testing

User acceptance of a system is the key factor of the success of any system. The system under study is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

Different Test Cases

<u>Action</u>	<u>Result</u>
1. Admin Login → Intelligence dietitian	Here, the admin of the Artificial website put his credentials in the Login page.
2. Login → and password it show main application on window.	Here when User enter user name
3. New Registration →	Here, the user put his personal details to get registered on this website.
4. View Diet → specification	Here, user can view appropriate diet as per
5. Cost Calculation & Credit Card Payment →	Here, the website calculates the cost and accordingly user pays the bill using credit card. Payment receipt is then sent to the user via mail.
Feedback Form →	Here, the user fills in feedback form which is then submitted to the administrator.

Code:



These are the libraries and webforms that are linked to master page

```

MasterPage.master  BML.aspx  Login.aspx
1  Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs" Inherits="MasterPage"
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
3
4  <html xmlns="http://www.w3.org/1999/xhtml">
5  <head runat="server">
6      <title></title>
7      <asp:ContentPlaceHolder id="head" runat="server">
8      </asp:ContentPlaceHolder>
9
10     <link rel="stylesheet" href="style.css" type="text/css" /><style type="text/css">_css3m{display:none}</style>
11     <script src="includes/jquery-1.10.2.min.js" type="text/javascript"></script>
12     <style>
13         .tabular
14         {
15             -webkit-border-radius: 10px;
16             -moz-border-radius: 10px;
17
18         }
19
20         .ATMtabular
21         {
22             -webkit-border-radius: 10px;
23             -moz-border-radius: 10px;
24             background-image:url(images/ATM.jpg);
25             height:300px;
26
27         }
28         .intabular
29
30
31
32
33
34
35     {
36         -webkit-border-radius: 10px;
37         -moz-border-radius: 10px;
38         background-color:White;
39     }
40     .auto-style1 {
41         font-family: gadugi;
42     }
43 </style>
44 </head>
45 <body style="background-image:url(images/Diet_Back.png); background-repeat: no-repeat; background-size: cover">
46
47     <form id="form1" runat="server">
48
49         <div>
50
51             <center>
52                 <table class="tabular" width="80%">
53
54                     <tr>
55                         <td colspan="2">
56                             
57                         </td>
58                     </tr>
59
60                     <tr>
61                         <td colspan="2">
62                             <asp:Panel ID="Panel1" runat="server">
63                                 <ul id="css3menu1" class="topmenu">
64                                     <input type="checkbox" id="css3menu1switcher" class="switchbox"><label onclick="" class="switch" for="cs

```

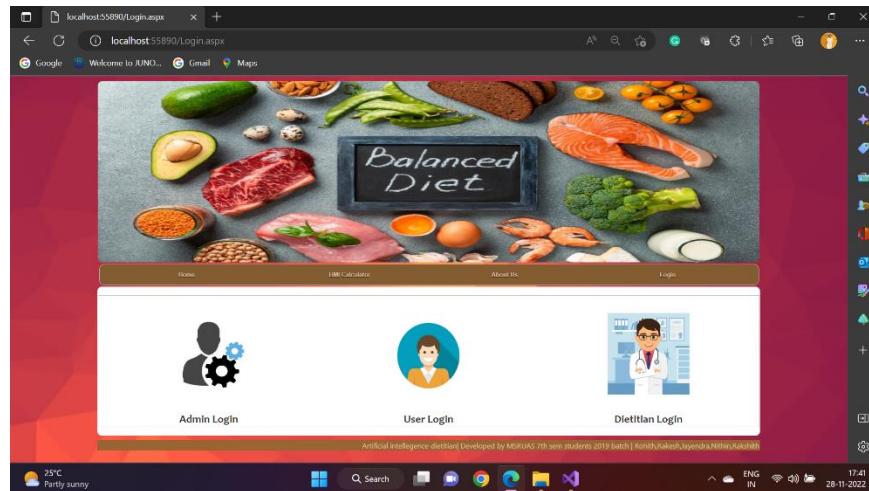
```

1      <td colspan="2">
2      <asp:Panel ID="Panel1" runat="server">
3          <ul id="css3menu1" class="topmenu">
4              <input type="checkbox" id="css3menu-switcher" class="switchbox"><label onclick="" class="switch" for="cs
5              <li class="topmenu"><a href="#" style="height:16px;line-height:16px;">Home</a></li>
6              <li class="topmenu"><a href="BMI.aspx" style="height:16px;line-height:16px;">BMI Calculator</a></li>
7              <li class="topmenu"><a href="#" style="height:16px;line-height:16px;">About Us</a></li>
8              <li class="toplast"><a href="Login.aspx" style="height:16px;line-height:16px;">Login</a></li>
9          </ul>
10     </asp:Panel>
11
12     <asp:Panel ID="Panel2" runat="server">
13         <ul id="css3menu1" class="topmenu">
14             <input type="checkbox" id="Checkbox1" class="switchbox"><label onclick="" class="switch" for="css3menu-s
15             <li class="topmenu"><a href="AddDietitian.aspx" style="height:16px;line-height:16px;">Add Dietitian</a><
16             <li class="topmenu"><a href="ViewUser.aspx" style="height:16px;line-height:16px;">View Customers</a></li>
17             <li class="topmenu"><a href="BMI.aspx" style="height:16px;line-height:16px;">BMI Calculator</a></li>
18             <li class="topmenu"><a href="Food.aspx" style="height:16px;line-height:16px;">View Charts</a></li>
19             <li class="toplast"><a href="Logout.aspx" style="height:16px;line-height:16px;">Logout</a></li>
20         </ul>
21     </asp:Panel>
22
23     <asp:Panel ID="Panel3" runat="server">
24         <ul id="css3menu1" class="topmenu">
25             <input type="checkbox" id="Checkbox2" class="switchbox"><label onclick="" class="switch" for="css3menu-s
26             <li class="topmenu"><a href="My.aspx" style="height:16px;line-height:16px;">My Details</a></li>
27             <li class="topmenu"><a href="BMI.aspx" style="height:16px;line-height:16px;">BMI Calculator</a></li>
28             <li class="topmenu"><a href="Chart.aspx" style="height:16px;line-height:16px;">Diet Chart</a></li>
29             <li class="topmenu"><a href="Pyramid.aspx" style="height:16px;line-height:16px;">Food Pyramid</a></li>
30             <li class="toplast"><a href="PersonalizedPlan.aspx" style="height:16px;line-height:16px;">Diet Plan</a></li>
31         </ul>
32     </asp:Panel>
33
34 </div>
35
36 <div>
37     <table>
38         <tr>
39             <td colspan="2">
40                 <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
41                     </asp:ContentPlaceHolder>
42                 </td>
43             </tr>
44             <tr>
45                 <td colspan="2">
46                     </td>
47                 </tr>
48             <tr>
49                 <td colspan="2">
50                     </td>
51                 </tr>
52             <tr>
53                 <td colspan="2">
54                     </td>
55                 </tr>
56             <tr>
57                 <td colspan="2" style="background-color: #996633; color: #e4c4ab; align="right">
58                     <asp:Label ID="Label1" runat="server" CssClass="auto-style1" Text="Artificial intelligence dietitian| Develo
59                 </td>
60             </tr>
61         </table>
62     </div>
63
64 </form>
65
66 </body>
67 </html>

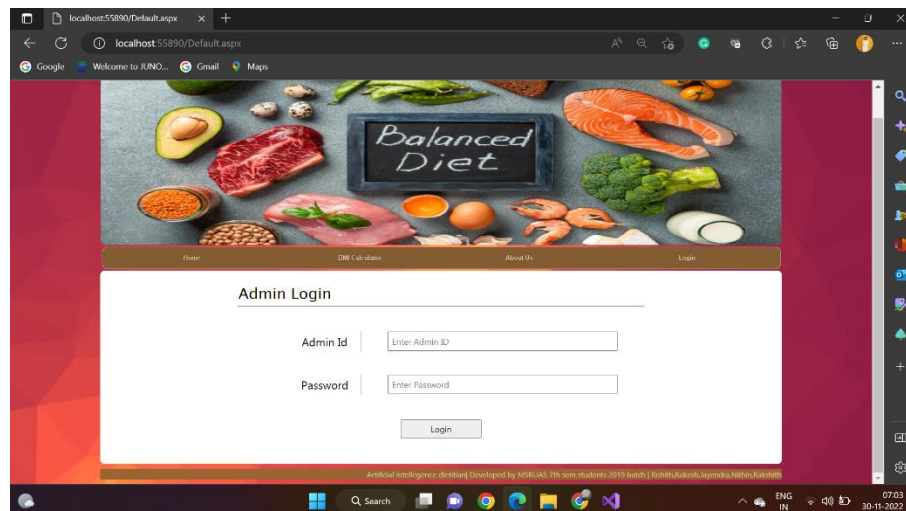
```

Master page Code of website

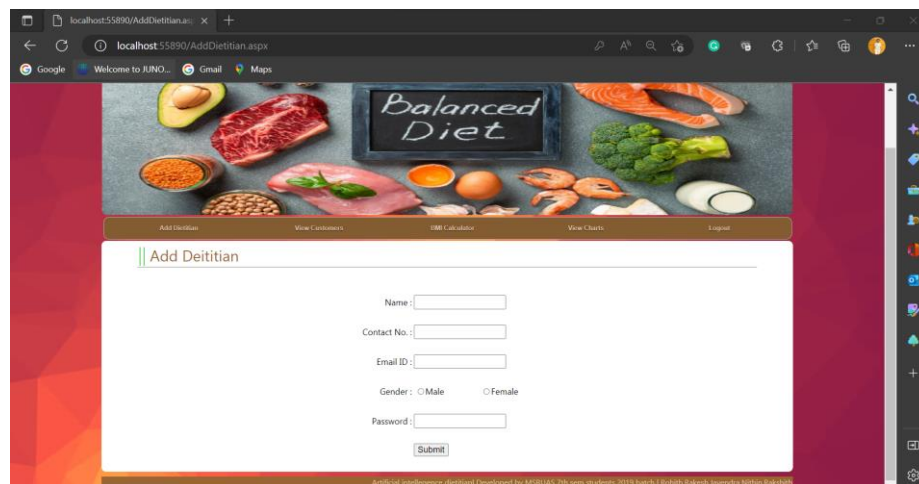
Results



We have 3 modules at the website Admin login user login dietitian login



Admin log in page where admin is required to enter admin id and password to add the dietitian to the system



localhost:55890/AddDietitian.aspx

Google Welcome to JUNO... Gmail Maps

Balanced Diet

Add Dietitian View Customers BMI Calculator View Charts Logout

Add Dietitian

Name :

Contact No. :

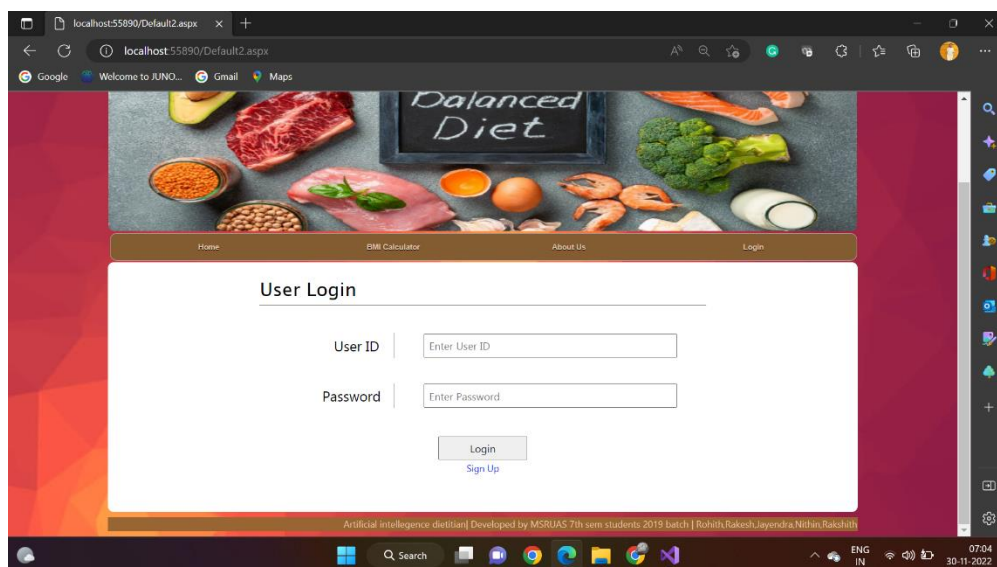
Email ID :

Gender : ☐ Male ☐ Female

Password :

Artificial intelligence dietitian Developed by MSRUAAS 7th sem students 2019 batch | Rohith, Rakesh, Jayendra, Nithin, Rakshit

Admin can add dietitian to the system and admin can view the customers of the system



localhost:55890/Default2.aspx

Google Welcome to JUNO... Gmail Maps

Balanced Diet

Home BMI Calculator About Us Login

User Login

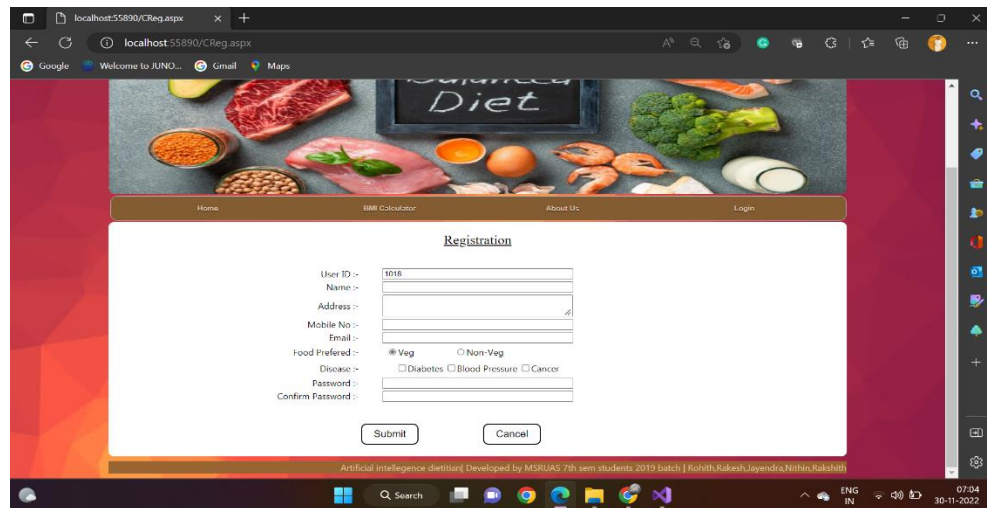
User ID

Password

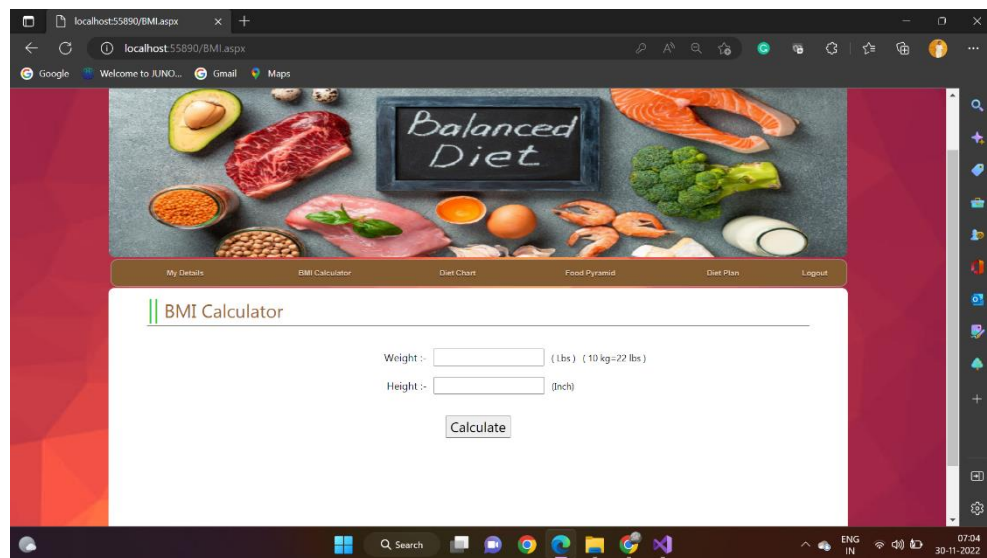
[Sign Up](#)

Artificial intelligence dietitian Developed by MSRUAAS 7th sem students 2019 batch | Rohith, Rakesh, Jayendra, Nithin, Rakshit

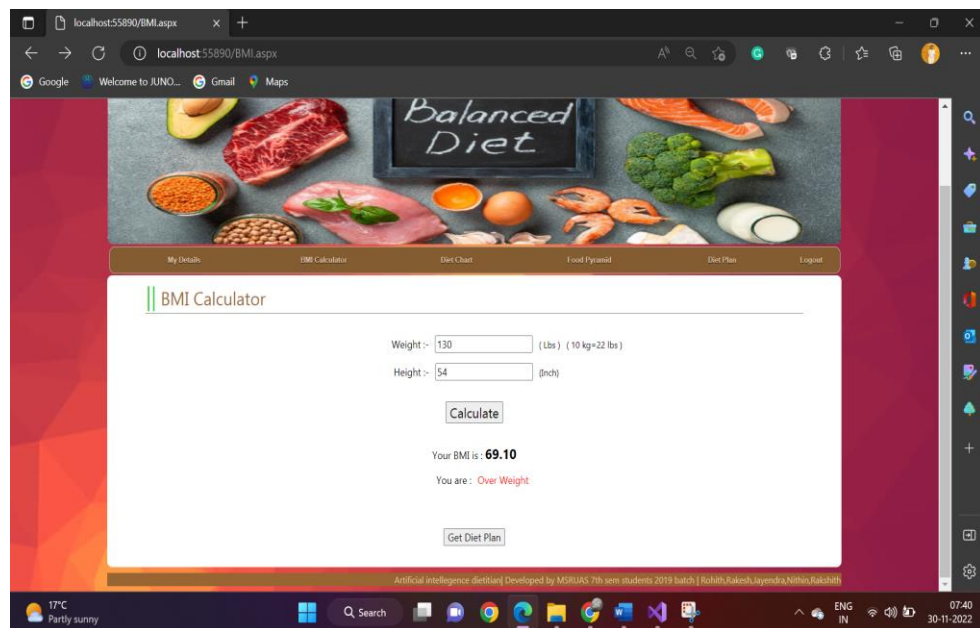
In user module where the user can login to his/her account or else he/she can singnup



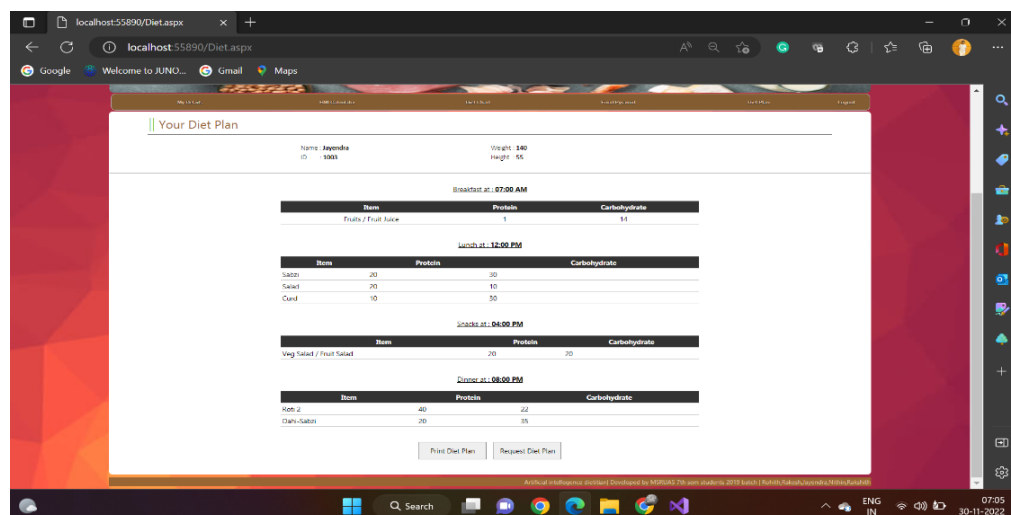
User signup page where its containing user personal details along food type and with pre medical history of the user



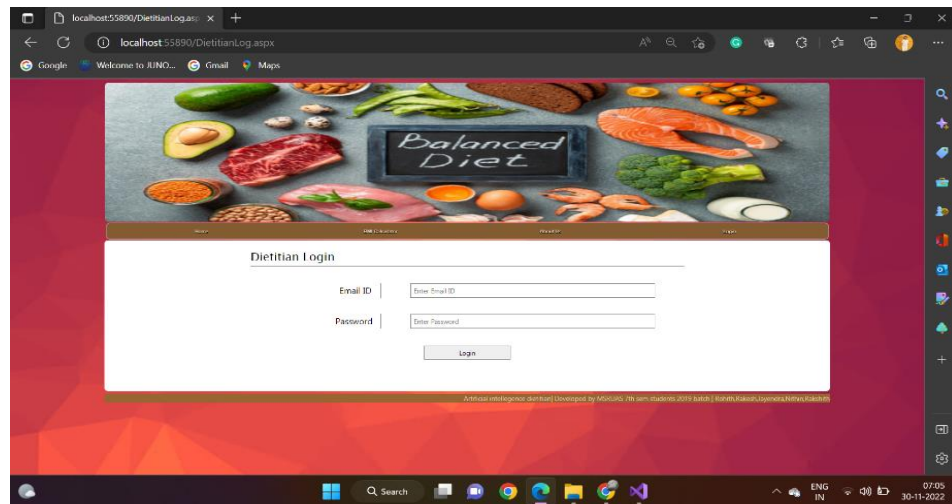
User can enter his weight in lbs and height in inches



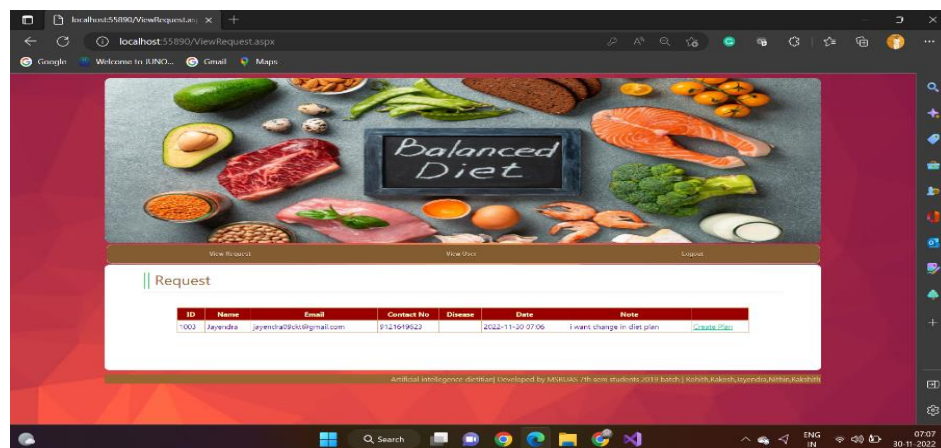
After user enters his weight and height bmi shows whether user is normal or under weight or over weight



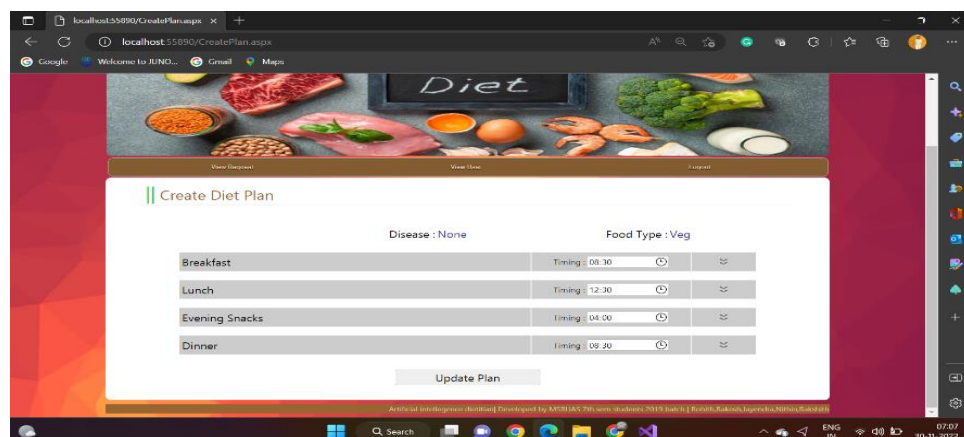
After user gets the bmi diet plan is generated automatically by using pre defined datasets and the user can print the diet or else if user is not willing to take this diet he can request to dietitian for change of diet



In dietitian module we will have dietitian log in



Dietitian can view all the requests from the user for change in diet plan



Dietitian will change the new diet preferences based on user requirements

Features1) **Load Balancing:**

Since the system will be available only the admin logs in the amount of load on server will be limited to time period of admin access.

2) **Easy Accessibility:**

Records can be easily accessed and store and other information respectively.

3) **User Friendly:**

The Website will be giving a very user friendly approach for all user

4) **Efficient and reliable:**

Maintaining the all secured and database on the server which will be accessible according the user requirement without any maintenance cost will be a very efficient as compared to storing all the customer data on the spreadsheet or in physically in the record books.

5) **Easy maintenance:**

Artificial Intelligence Dietitian website is design as easy way. So maintenance is also easy.

➤ CONCLUSION

Artificial Intelligence and the technology are one side of the life that always interest and surprise us with the new ideas, topics, innovations, products ...etc. AI is still not implemented as the films representing it (i.e. intelligent robots), however there are many important tries to reach the level and to compete in market, like sometimes the robots that they show in TV. Nevertheless, the hidden projects and the development in industrial companies.

At the end, we've been in this research through the AI definitions, brief history, and applications of AI in public, applications of AI in military, ethics of AI, and the three rules of robotics. This is not the end of AI, there is more to come from it, who knows what the AI can do for us in the future, maybe it will be a whole society of robots.

➤ References**■ Websites**

- <http://www.wikipedia.com>
- en.wikipedia.org
- <http://projectsgeek.com/2016/08/ai-diet-consultant-project.html>

■ Books

- Visual Studio 2010 - Rob Thayer