**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

PNR NO.: ………………… Roll no:08

1. Name of the Student

**Rane Yash Santosh**

1. Title of the Project

**Online Artificial Intelligence Dietitian**

1. Name of the Guide

**PROF.** **Vandana Singh**

**PROF. Sumathi Rajkumar**

1. Teaching experience of the Guide
2. **Years**
3. Is this your first submission? **Yes No**

Signature of the Student Signature of the Guide

Date: …………………. Date: …………………

Signature of the Coordinator

ssDate: ……………………

**ABSTRACT**

This work proposes an intelligent agent, called the personal dietitian agent, based on the user’s characteristics and specification. This agent can create a meal plan according to a person’s lifestyle and particular health needs.

The expert recommend eating a wide variety of foods, including vegetables, whole grains, fruits, non-factor low-fat dairy products, beans, lean meats, poultry and fish. However, each person has a unique dietary pattern and have different health issues so a dietitian create a meal plan depending on each case.

The online artificial dietitian is an application with artificial intelligence about human diets. It act as a diet consultant similar to a real dietitian. This system acts in a similar way as that of a dietitian.

A person is order to know its diet plan needs to give some information to the dietitian such as its body type, weight, height and its working hour details.

The system asks all this data from the user and processes it ti provide the diet plan to the user. Thus the user does not need to visit any dietitian which also save time and the user can get required diet plan in just a click.

**ACKONOWLEDGEMENT**

I am pleased to present “**Online Artificial Intelligence Dietitian**” project and take this opportunity to express my profound gratitude to all those people who helped me in completion of this project.

I express my deepest gratitude towards **Prof. Sumathi Rajkumar and**

**Prof. Vandana Singh** our project guide for their valuable and timely advice during the various phase in my project. I would also like to thank them for providing us with all proper facilities and support, patience and faith in our capabilities and for giving us flexibility in term of working and reporting schedules.

I thank our collage for providing us with excellent facilities that helped me to complete and present this project. I would also like to thank the staff member and lab assistant for permitting us to use computer in the lab as and when required.

I would like to thank all my friends for their smiles and friendship making the collage life enjoyed and memorable and family member who always stood beside me and provided the utmost important moral support. Finally, I would like to thank everyone who has helped us directly or indirectly in our project.

**DECLARATION**

I here by declare that the project entitled, **“Online Artificial Intelligence Dietitian”** done at **Nirmala Memorial Foundation College of Commerce & Science,** has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

**Yash Santosh Rane**

**Chapter 1: Introduction**

**1.1 Background................................................................6**

**1.2 Objectives.................................................................. 6**

**1.3 Purpose, Scope and Applicability**

**1.3.1 Purpose ..........................................................7**

**1.3.2 Scope ..............................................................7**

**1.3.3 Applicability...................................................7**

**1.4Organisation of Report ................................................8-10**

**Chapter 2: Survey Of Technologies............................... 11-29**

**Chapter 3: Requirements And Analysis**

**3.1 Problem Definition.................................................. 37**

**3.1.1Problem statement…………………………30**

**3.1.2Modules…………………………………….30**

**3.1.3Explain each modules………………………31-32**

**3.2 Requirements Specifications....................................32**

**3.3 Planning and Scheduling......................................... 33**

**Chapter 4: System Design**

**4.1 Basic Modules...........................................................34-39**

**4.2 Data Design ............................................................... 40**

**4.3 Procedural Design**

**4.3.1 Logic Diagrams.............................................. 41-42**

**Chapter 5:**

**5.1 Implementation Approach………………………...50-52**

**5.2 Coding Details and Code Efficiency………………53**

**5.3 Testing Approach……………………………………….54**

**5.4 Modification and Improvements………………….54**

**5.5 Test Cases…………………………………………………..55-58**

**Chapter 6:**

**6.1 Test Report………………………………………....60-62**

**6.2 User Documentation……………………………………….63-77**

**Chapter 7:**

**7.1 Significance of the System…………………………79**

**7.2 Limitation of the System…………………………..79**

**7.3 Future Scope of Project…………………………....79**

**Reference……………………………………………….80**

**Chapter 1**

**INTRODUCTION**

* 1. **Background**

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

* 1. **Objective**

In this system of artificial intelligence dietitian, using the technique of artificial intelligence, we will get access to all the facilities via this web application, which is actually provided by a human dietician. The main advantage of using this web application is that the time required by the people to travel to the dietician will be reduced and also it reduces the cost of hiring dieticians for some particular purpose. Also, this application offers more than one diet plan also, for some particular kind of functionalities of human bodies.

**1.3. Purpose, Scope, and Applicability**

**1.3.1. Purpose**

The online artificial dietician is a bot with artificial intelligence about human diets.

It acts as a diet consultant similar to a real dietician. Dieticians are educated with nutrient value of foods. A dietician 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. It asks about how many hour the user works, his height, weight, age etc. The system stores and processes this data and then calculates the nutrient value needed to fill up users needs. The system then shows an appropriate diet to the users and asks if user is ok with it, else it shows other alternate diets to fill up users needs.

**1.3.2. Scope**

* No need of consulting doctor for diet plans.
* This system provides full details of the nutrient constitution in body and if required more or not along with the plan by just answering to some queries.
* Saves money and very effective and give accurate results as it is coded with keeping diet chart in mind.
* There are alternative diet chart provided by the system if the user don't like any.

**1.3.3 Applicability**

* Dietitians can use this system to make sure what they recommend patients.
* This system can be very well used in medical colleges for teaching and practicing purposes so that student can learn from it.
* This system can also be utilized in gym particularly for calculating the customers' calories and diet plans.
* Individual can also use this software especially for themselves in home

**1.4 ORGANISATION OF REPORT**

**Chapter 2**

**SURVEY OF TECHNOLOGIES**

In this chapter Survey of Technologies should demonstrate the students awareness and understanding of Available Technologies related to the topic of the project. The detail of all related technologies that are necessary to complete the project.

**Chapter 3**

**REQUIREMENTS AND ANALYSIS**

**Problem Definition:**

Defining the problem on which the we are working in the project. We Provide details of the overall problem and then divide the divide the problem in to sub-problems.

**Requirement Specification:**

We define the requirement of the system, independent of how these requirements will be accomplished.

**Planning and Scheduling:**

Planning and scheduling is a complicated part of software development. Planning, for our purpose, can be thought of as determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account, rule, know as constraints, which, control when certain tasks can or cannot happen. Scheduling can be thought of as determining whether adequate resource are available to carry out the plan. We show the Gantt chart and Program Evaluation Review Technique (PERT).

**Software and hardware Requirements:**

Define the details of all the software and hardware needed for the development and implementation of the project.

**Conceptual Models:**

Conceptual Model could consist of complete Data Flow Diagram, ER diagrams, Object oriented diagrams, System Flowchart etc.

**Chapter 4**

**SYSTEM DESIGN**

**Basic Modules:**

In this phase, we briefly describe all the modules and the functionality of these modules.

**Data Design:**

Data design will consist of how data is organized, managed and manipulated.

**Schema Design:**

We define the structure and explanation of schemas used in the project.

**Data Integrity and Constraints:**

We Data and explain all the validity checks and constraints provided to maintain data integrity.

**Procedural Design:**

Procedural Design is a systematic way for developing algorithms or procedurals.

**Logic Diagram:**

Define the systematic flow of procedure that improves its comprehension.

**Data Structure:**

Create and the define the data structure used in procedures.

**Algorithms Design:**

With proper explanation of input data, output data, logic of processes, design and explain the working of algorithms.

**User Interface Design:**

Defining user, task, environment analysis and how to map those requirements in order to develop a “User Interface”. We Describe the external and internal components and the architecture of user interface. We Show some rough pictorial view of the user interface and its components.

**Security Issues:**

Discuss Real-time Considerations and Security issues related to the project and explain how the student intends avoiding those security problem. What are the security policy plans and architecture?

**Test Case Design:**

Define test cases, which will provide easy detection of error and mistakes with in a minimum period of time and with the least effort. We explain different conditions in which the we wish to ensure the correct working of the project.

**CHAPTER 2**

**SURVEY OF TECHNOLOGIES**

**Front End Technology:**

**Microsoft .NET Framework**

* The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:
* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

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. We 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. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that we 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. Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the 15 runtime enables us to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage. The following illustration shows the relationship of the common language runtime and the class library to our 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.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network.

The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally featuring rich.

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 16 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, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of tn moday and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

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 us to use managed code to write our 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 our 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 we can use to develop our own collection classes. Our collection classes will blend seamlessly with the classes in the .NET Framework.

As we would expect from an object-oriented class library, the .NET Framework types enable us 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, we 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 we write an ASP.NET Web Form application, we 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. We can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases, the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

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. Our 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 our custom managed code to control the behavior of the server. This model provides us 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 20 perform standard operations while our 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.

If our have used earlier versions of ASP technology, we will immediately notice the improvements that ASP.NET and Web Forms offers. For example, we can develop Web Forms pages in any language that supports the .NET Framework. In addition, wer code no longer needs to share the same file with our HTTP text (although it can continue to do so if we prefer).

Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged 21 ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

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 solutions.

If we develop and publish our own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables us to focus on the logic of our service, without concerning our self with the communications infrastructure required by distributed software development.

Finally, like Web Forms pages in the managed environment, our XML Web service will run with the speed of native machine language using the scalable communication of IIS.

**Active Server Pages.NET**

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

**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 22 right out of the box. This amounts to dramatically better performance before we ever write a line of code.

**World-Class Tool Support**. The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

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

**Simplicity.** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows us to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

**Manageability.** ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to our 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. 23 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.

**Scalability and Availability.** ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep our application constantly available to handle requests.

**Customizability and Extensibility.** ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with our own custom-written component. Implementing custom authentication or state services has never been easier.

**Security.** With built in Windows authentication and per-application configuration, we can be assured that our applications are secure.

Language Support

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

**What is ASP.NET Web Forms?**

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework has been specifically designed to address a number of key deficiencies in the previous model. In particular, it provides:

The ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that a page developer has to write.

The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti code").

The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP code is opaque to tools).

ASP.NET Web Forms pages are text files with an .aspx file name extension. They can be deployed throughout an IIS virtual root directory tree. When a browser client requests .aspx resources, the ASP.NET runtime parses and compiles the target file into a .NET Framework class. This class can then be used to dynamically process incoming requests. (Note that the .aspx file is compiled only the first time it is accessed; the compiled type instance is then reused across multiple requests).

An ASP.NET page can be created simply by taking an existing HTML file and changing its file name extension to .aspx (no modification of code is required).

For example, the following sample demonstrates a simple HTML page that collects a user's name and category preference and then performs a form post back to the originating page when a button is clicked:

ASP.NET provides syntax compatibility with existing ASP pages. This includes support for code render blocks that can be intermixed with HTML content within an .aspx file. These code blocks execute in a top-down manner at page render time.

**Code-Behind Web Forms**

ASP.NET supports two methods of authoring dynamic pages. The first is the method shown in the preceding samples, where the page code is physically declared within the originating .aspx file.

An alternative approach--known as the code-behind method--enables the page code to be more cleanly separated from the HTML content into an entirely separate file.

**Introduction to ASP.NET Server Controls**

In addition to (or instead of) using code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages. Server controls are declared within an **.aspx** file using custom tags or intrinsic HTML tags that contain a **runat=**"server" attributes value. Intrinsic HTML tags are handled by one of the controls in the **System.Web.UI.HtmlControls** namespace. Any tag that doesn't explicitly map to one of the controls is assigned the type of **System.Web.UI.HtmlControls.HtmlGenericControl**.

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within an **<input type="hidden">** form field that is round-tripped between requests). Note also **that** no client-side script is required.

In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the **<asp:adrotator>**control can be used to dynamically display rotating ads on a page.

**1.** ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.

**2.** ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).

**3.** ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.

**4.** ASP.NET server controls provide an easy way to encapsulate common functionality.

**5.** ASP.NET ships with 45 built-in server controls. Developers can also use controls built by third parties.

**6.** ASP.NET server controls can automatically project both up level and down level HTML.

**7.** ASP.NET templates provide an easy way to customize the look and feel of list server controls.

**8.** ASP.NET validation controls provide an easy way to do declarative client or server data validation.

**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 Basic.NET, we can host reports on Web and Windows platforms and publish Crystal reports as Report Web Services on a Web server.

To present data to users, we could write code to loop through record sets and print them inside our Windows or Web application. However, any work beyond basic 27 formatting can be complicated: consolidations, multiple level totals, charting, and conditional formatting are difficult to program.

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

**Report Experts**

Using the Crystal Report Experts, we can quickly create reports based on our development needs:

* Choose from report layout options ranging from standard reports to form letters, or build our own report from scratch.
* Display charts that users can drill down on to view detailed report data.
* Calculate summaries, subtotals, and percentages on grouped data.
* Show TopN or BottomN results of data.
* Conditionally format text and rotate text objects.

**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.

A database typically has two components: the files holding the physical database and the database management system (DBMS) software that applications use to access data. The DBMS is responsible for enforcing the database structure, including:

* Maintaining the relationships between data in the database.
* Ensuring that data is stored correctly and that the rules defining data relationships are not violated.
* Recovering all data to a point of known consistency in case of system failures.

**Relational Database**

There are different ways to organize data in a database but relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data. In a relational database, data is collected into tables (called relations in relational theory).

When organizing data into tables, we can usually find many different ways to define tables. Relational database theory defines a process, normalization, which ensures that the set of tables we define will organize our data effectively.

**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, we 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).

**SQL Server Features**

Microsoft SQL Server supports a set of features that result in the following benefits:

**Ease of installation, deployment, and use**

SQL Server includes a set of administrative and development tools that improve our ability to install, deploy, manage, and use SQL Server across several sites.

**Scalability**

The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows® 95/98 to large, multiprocessorservers running Microsoft Windows NT®, Enterprise Edition.

**Data warehousing**

SQL Server includes tools for extracting and analyzing summary data for online analytical processing (OLAP). SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

**System integration with other server software**

SQL Server integrates with e-mail, the Internet, and Windows.

**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 we 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.

**Normalization Theory:**

Relations are to be normalized to avoid anomalies. In insert, update and delete operations. Normalization theory is built around the concept of normal forms. A relation is said to be in a particular form if it satisfies a certain specified set if constraints. To decide a suitable logical structure for given database design the concept of normalization, which are briefly described below.

1. **1 st Normal Form (1 N.F):** A relation is said to be in 1 NF is and only if all unaligned domains contain values only. That is the fields of an n-set should have no group items and no repeating groups.
2. **2 nd Normal Form (2 N.F):** A relation is said to be in 2 NF is and only if it is in 1 NF and every non key attribute is fully dependent on primary key. This normal takes care of functional dependencies on non-key attributes.
3. **3 rd Normal Form (3 N.F):** A relation is said to be in 3 NF is and only if it is in 2 NF and every non key attribute is non transitively dependent on the primary key. This normal form avoids the transitive dependencies on the primary key.
4. **Boyce code Normal Form (BCNF):** This is a stronger definition than that of NF. A relation is said to be in BCNF if and only if every determinant is a Candidate key.
5. **4 th Normal Form (4 NF):** A relation is said to be in 4 NF if and only if whenever there exists a multi valued dependency in a relation say A->->B then all of the relation are also functionally dependent on A (i.e. A->X for all attributes x of the relation.).
6. **5 th Normal Form (5 NF) OR Projection Join Normal Form (PJNF):** A relation R is in 5 NF. if and only if every join dependency in R is implied by the candidate key on R. A relation can’t be non-loss split into two tables but can be split into three tables. This is called Join Dependency.

**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. We 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.

A Data Adapter is the object that connects to the database to fill the Dataset. Then, it connects back to the database to update the data there, based on operations performed while the Dataset held the data. In the past, data processing has been primarily connection-based. Now, in an effort to make multi-tiered apps more efficient, data processing is turning to a message-based approach that revolves around chunks of information. At the center of this approach is the Data Adapter, which provides a bridge to retrieve and save data between a Dataset and its source data store. It accomplishes this by means of requests to the appropriate SQL commands made against the data store.

The XML-based Dataset object provides a consistent programming model that works with all models of data storage: flat, relational, and hierarchical. It does this by having no 'knowledge' of the source of its data, and by representing the data that it holds as collections and data types. No matter what the source of the data within the Dataset is, it is manipulated through the same set of standard APIs exposed through the Dataset and its subordinate objects.

While the Dataset has no knowledge of the source of its data, the managed provider has \*-3-\*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 f\*our basic objects: the Command, Con--nection, 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 us 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.

When dealing with connections to a database, there are two different options: SQL Server .NET Data Provider (System.Data.SqlClient) and OLE DB .NET Data Provider (System.Data.OleDb). In these samples we will use the SQL Server .NET Data Provider. These are written to talk \*directly to Microsoft SQL Server. The OLE DB .NET Data Provider is used to talk to any OLE DB provider (as it uses OLE DB underneath)

-

1. ADO.NET is the next evolution of ADO for the .Net Framework.
2. ADO.NET was created with n-Tier, statelessness and XML in the forefront. Two new objects, the Dataset and Data Adapter, are provided for these scenarios. ADO.NET can be used to get data from a stream, or to store data in a cache for updates.
3. There is a lot more information about ADO.NET in the documentation.
4. Remember, we can execute a command directly against the database in orderto do inserts, updates, and deletes. We don't need to first put data into a Dataset in order to insert, update, or delete it.
5. Also, we can use a Dataset to bind to the data, move through the data, and navigate data relationships.

**CHAPTER 3**

**REQIREMENTS AND ANALYSIS**

**3.1 Problem definition**

**3.1.1 Problem statement**

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

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

**3.1.2 Modules**

1. Admin login
2. User login
3. Dietitian login

**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, systemautomatically 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.
* **Chat:** User can chat with the dietitian and ask or solve their queries online.
* **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. Therefore, 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.
* **Chat:** A Dietitian can chat with the dietitian and ask or solve user’s queries online.

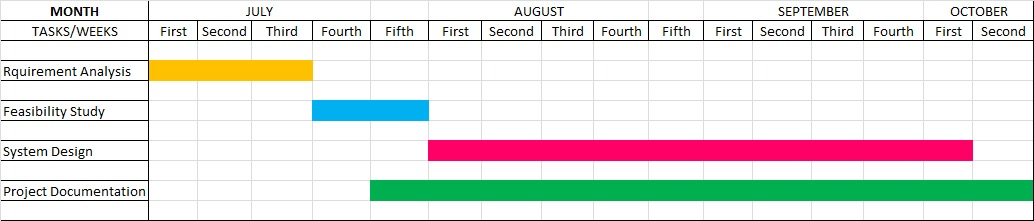
**3.2 Requirement specification**

* **Hardware Requirement:**
* i3 OR i5 Processor Based Computer
* 2GB-Ram
* 5 GB Hard Disk
* Internet Connection
* **Software Requirement:**
* Windows 7 or higher
* Visual Studio 2010
* SQL Server 2008
* Google Chrome Browser

**-**

**3.3 Planning and scheduling**

**Gantt Char**

****

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 Basic Modules**

**User case diagram**

**1. User Case Diagram for User**

User

**2. User Case Diagram for Admin**

**Admin**

**3.User Case Diagram for Dietitian**

**Dietitian**

**Data Flow Diagrams**

User

Database

Query

**DATABASE DETAIL**

Query

User

Check for user Requirement

Database

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ User need \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Feedback For

User

LEVEL 1 DFD

User

Give info about DB

Give

Respect

To

user

Via Artificial Intelligence Diet DB

LEVEL 2 DFD: PREDICTION

**4.2 Data Design**

**ER-Diagram**

About

Diet Plan

View

View

Dietitian site

Access

User

**4.3 Procedural Design**

**4.3.1 Logic Diagram**

**Activity Diagram**

Query

Enter User-id and Password

Validation

No

User Administrator

**Sequence Diagram**

Artificial Intelligence Dietitian

Main Page

User

Enters User\_id & Password

Validates User

Invalid Login

----------------------------------------------

New Registration

If Valid, then Displays Main page

Online Consulting ability

Diet Data Processing

Retrieve’s Diet Data Processing

User nutrition counseling

Online Chat

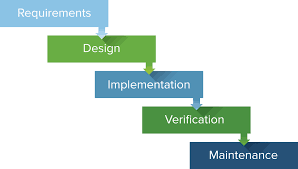
**CHAPTER 5**

**IMPLEMENTATION AND TASTING**

**5.1 Implementation Approach:**

**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.

****

The sequential phases in Waterfall model are:

* Requirement Gathering and analysis

Requirements for the project are gathered and fixed in this phase.

* System Design

System architecture and design are developed based on requirements gathered.

* Implementation

The design and source code are implemented in developers environment.

* Integration and Testing

The components of the system are integrated and tested together. Individual components are also tested.

* Deployment of system

The system is deployed in the user’s environment for use in real life.

* Maintenance

The system is maintained time to time meeting new trends in the market.

In this project the waterfall model is used because:

* All the requirements were clear and fixed.
* Requirements are not ambiguous.
* The design and architecture were developed.
* Ample resources with required expertise are available freely.
* The project is short.
* It is good to use this model when the technology is well understood.
  1. **Coding Details and Code Efficiency** 
     1. **Code Efficiency**

Code efficiency is a broad term to depict the reliability, speed and programming methodology used in developing code for an application. Code efficiency is directly linked with algorithmic efficiency and the speed of runtime execution for software. Its is the key element is ensuring high performance. The goal of code efficiency is to reduce resource consumption and completion time as much as possible with minimum risk to the business or operating environment. The software product can be accessed and evaluated with the help of the efficiency of the code used.

Code efficient plays a signification role in applications in a high-execution-speed environment where performance and scalability are paramount.

One of the recommended best practices in coding is to ensure good code efficiency. Well-developed programming code should be to handle complex algorithms.

* 1. **Testing Approach**

**5.3.1Unit Testing**

Unit testing focuses verification effort on the smallest unit of the software design, the module. This is also known as “Module Testing”. The module 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.

* + 1. **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 test to uncover error 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 error uncovered are corrected for the text testing step.

**5.3.3 Beta 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.

**5.4 Modification and Improvements**

If user in enter the alphabet instead of number then it will reset the weight and height text box but the user is not aware of what happened so we are modify the code if the user in enter the alphabet then website will give a Warning that user enter the Invalid input so user can re-enter the valid data.

**5.5 Test Cases**

**User Test case**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  Case Id | Test  Scenario | Test Steps | Test Data | Expected  Result | Actual  Result | Pass/Fail |
| TU01 | Check User  Login with  Valid Data | 1.Go to site  2.Enter User Id  3.Enter  Password  4.Click Login | User Id=user99  Password=1234 | User Should  Login into  Application | User go to the page | Pass |
| TU02 | Check User  Login  With Invalid  Data | 1.Go to Site  2.Enter User Id  3.Enter  Password  4.Click Login | User Id=user99  Password=5689 | User Should  Not Login  into  Application | User cannot go to the page | Pass |
| TU03 | For new user enter registration details with Valid data | 1.Go to site  2.Enter name  3.Enter DOB  4.Enter Gender  5.Enter address  6.Enter mobile no  7.Enter mail ID  8.Enter User ID  9.Enter Password | Name: Kalpana  Dob:24/4/1998  Gender: Female  Address=Mumbai  Mobile No=65845986  Email Id= [Emailid@gmail.com](mailto:Emailid@gmail.com)  User Id=user99  Password-1234 | User Should Register for an Application | User register Successfully | Pass |
| TU04 | For New  User  Enter  Registration  Details with  Invalid Data | 1.Go to Site  2.Enter Name  3.Enter DOB  4.Enter Gender  5.Enter Address  6.Enter Mobile  No.  7.Enter Email Id  8.Enter User Id  9.Enter  Password | Name: kalapan78  Dob:24/4/1998  Gender: Female  Address=Mumbai  Mobile No=Blank  Emailid=  Emailid@com  User Id=Blank  Password-1234 | User Should  Not Register  Until User  Will  Fill the all the  correct Data  in  The  Registration  Form | User will get Error message “please enter Valid data” | Pass |
| TU05 | Calculate  BMI  With Valid  Data | 1.Enter weight  2.Enter Height  3.Click on Calculate Button | Weight=50  Height=5.4 | User Should  Get Diet Plan  And View  Diet Plan | User will get the Diet Plan as per his/her Weight and Height | Pass |
| TU06 | Calculate  BMI  With Invalid  Data | 1.Enter weight  2.Enter Height  3.Click on Diet  Plan Button | Weight=45  Height=Blank  (Not Enter Height) | User Should  Enter a  Height  And Not Get  the Diet Plan | User get error Message “Please Enter Height” |  |

**Dietitian Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  Case  Id | Test Scenario | Test Steps | Test Data | Expected  Result | Actual  Result | Pass/Fail |
| TD01 | Check Dietitian Login with Valid Data | 1.Go to Site  2.Enter Dietitian Id  3.Enter Password  4.Login | Dietitian  Id=DietianGeeta1  Password=Diet12 | Dietitian  Should Login  Into  Application | Dietitian Page will be display | Pass |
| TD02 | Check  Dietitian  Login with  Invalid Data | 1.Go to Site  2.Enter Dietitian Id  3.Enter Password  4.Login | Dietitian  Id=DietianGeeta1  Password=1234 | Dietitian  Should Not  Login Into  Application | Dietitian should get message “Enter valid Email and Password” | Pass |
| TD03 | Check the  Diet Plan  Request | 1.Login  2.Click on View  Diet Request | List of all the  User Request | Dietitian  Should View all  the Request | Dietitian can View all User Request | Pass |
| TD04 | Create the  Diet Plan for  the every  Users that  request for  Diet Plan. | 1.View Diet Plans  2.Click on Create  Diet Plan for  Particular User  Request | Create the Diet  Plan for  Particular User  according to  Users Physical  Health | Dietitian should  create a Diet  Plan for that  user who  Request for  Diet Plan | Dietitian create the Diet Plan and send to the user | Pass |

**Admin Test Case**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case Id | Test Scenario | Test Step | Test Data | Expected Result | Actual Result | Remark |
| TA01 | Admin login with Valid Data | 1.Go to site  2.Enter Admin User-Id  3.Enter Password  4.Click on Login | User-Id=admin  Password=admin | Admin should login into the Application | Admin Page will be display | Pass |
| TA02 | Admin login with Invalid Data | 1.Go to site  2.Enter Admin User-Id  3.Enter Password  4.Click on Login | User-Id=admin  Password=admin | Admin should login into the Application | Admin will get Error Message “Enter Invalid User Id and Password” | Pass |
| TA03 | Admin can Add dietitian | 1.Click on the Tab  2.Add details  3.Click on Submit button | Enter User Id, Name,  Address, Email Id, Mobile No, Disease, Password | Add Dietitian should be Successful | Add Dietitian should be Successfully | Pass |
| TA04 | Admin can View Customer | 1.Click on the tab  2.View Customer | View User List | View Updated User List | Admin View all User | Pass |

**CHAPTER 6**

**RESULTS AND DISCUSSION**

* 1. **Test Report**

Test Report is needed to reflect testing results in a formal way, which gives an opportunity to estimate testing results quickly. It is a document that records data obtained from an evaluation experiment in an organized manner, describes the environmental or operating conditions, and shows the comparison of test results with test objectives.

There are various types of test reports that are widely being used in the software industry. As you know software development process goes through different stages of development such as planning, coding, testing, and maintenance etc. At every stage it generates an artifacts which acts as an input to its subsequent stage.

* Our Website has gone through both ALPHA and BETA testing. Testers were:

For Alpha testing:

1. Omkar Rane
2. Santosh Rane

For Beta testing:

1. Sakshi Mestry
2. Uma

The Test Report was based on the TEST CASES performed.

The following Reports are of Beta Testing:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case id | Test case objective | Test case specification | Result | Decision |
| 1 | Check whether user is able to login with wrong details | Enter wrong login details and click on login | To be directed on login page with proper error | OK |
| 2 | Check interface between login page and home page | Enter login details and click on Login | To be directed to home page | OK |
| 3 | Check whether user is able to login without credential | Do not enter login details and click on login | To be directed on login page with proper error | OK |
| 4 | Verify that the login link is redirected to home/sign in page | Click on logout button/link | To be directed on home/sign in page | OK |
| 5 | Verify that diet plan page is displayed | Enter the details the user wants to add. | Status should be displayed | OK |
| 6 | Verify that proper validation error occur at login/registration page | Enter wrong credentials | Validation/verification error should be shown | OK |
| 7 | Verify that user is able to see its personalized diet plan . | Check the diet plan | To show the Personalized diet plan | OK |

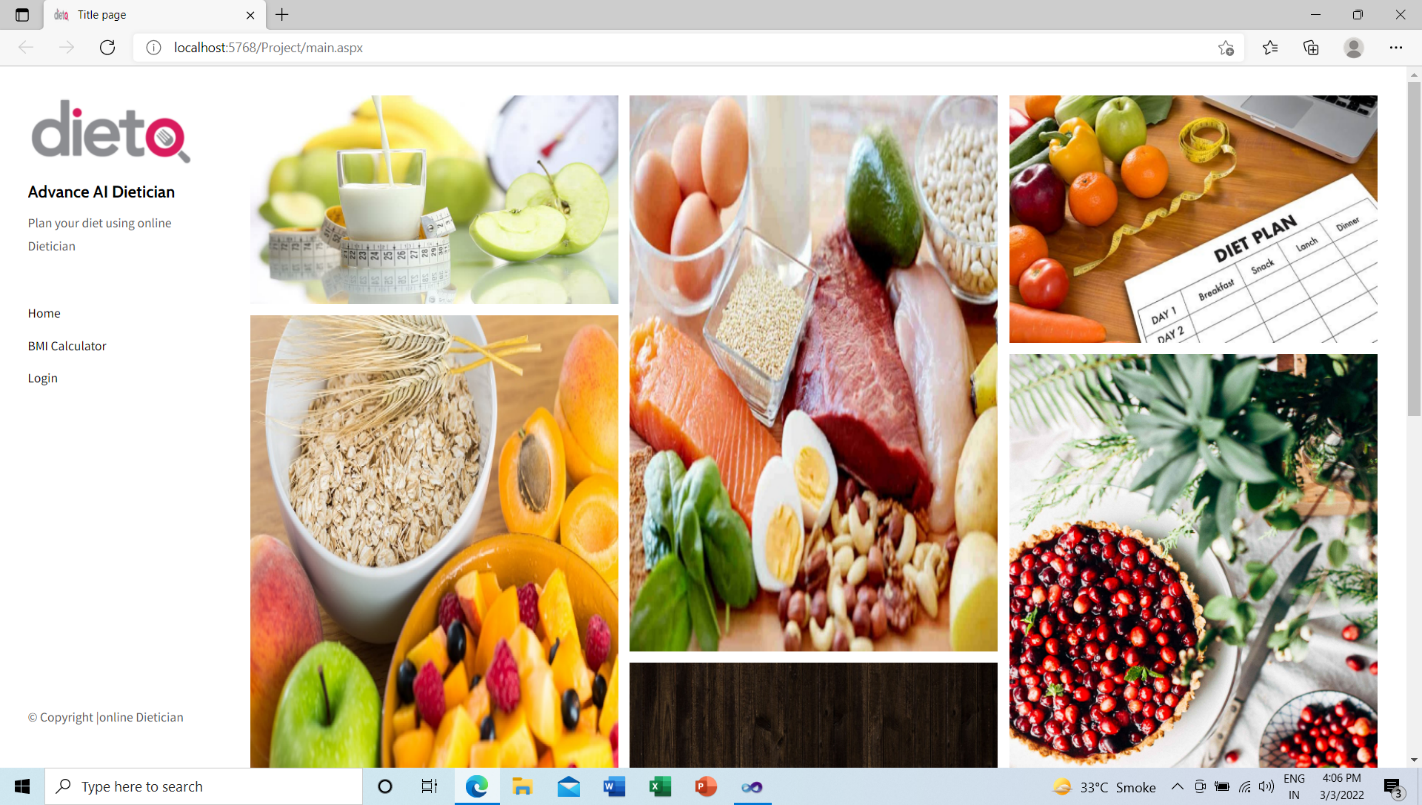
The following reports are again verified and checked by the Alpha Testers according to the requirements given the Beta Testers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case id | Test case objective | Test case specification | Result | Decision |
| 1 | Check whether user is able to login with wrong details | Enter wrong login details and click on login | To be directed on login page with proper error | OK |
| 2 | Check interface between login page and home page | Enter login details and click on Login | To be directed to home page | OK |
| 3 | Check whether user is able to login without credential | Do not enter login details and click on login | To be directed on login page with proper error | OK |
| 4 | Verify that the login link is redirected to home/sign in page | Click on logout button/link | To be directed on home/sign in page | OK |
| 5 | Verify that diet plan page is displayed | Enter the details the user wants to add. | Status should be displayed | OK |
| 6 | Verify that proper validation error occur at login/registration page | Enter wrong credentials | Validation/verification error should be shown | OK |
| 7 | Verify that user is able to see it personalized diet plan | Check the diet plan | To show the personalized diet plan | OK |

* 1. **User Documentation**

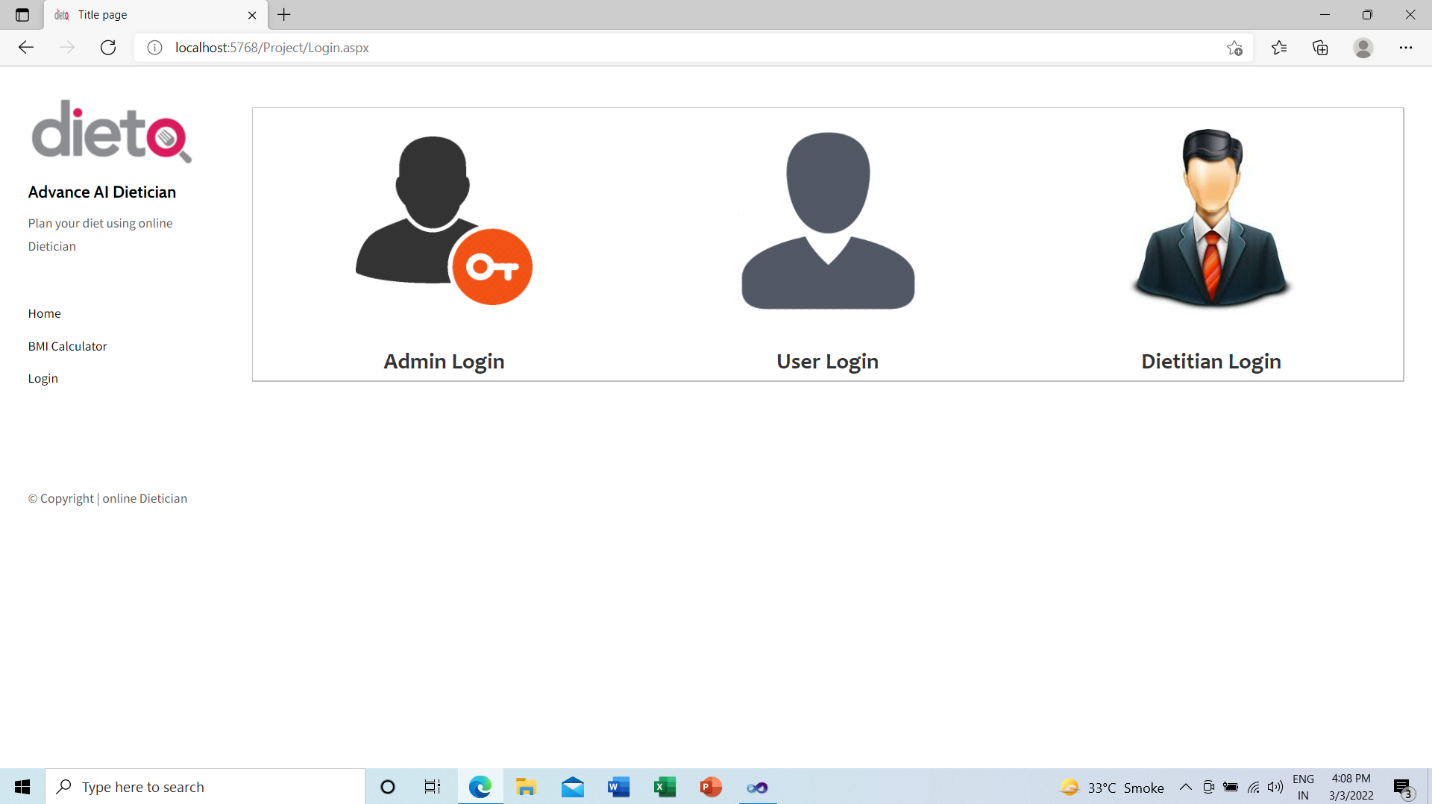
1. **Home Page**

In this Page can Calculate the BHI .User can Get the Diet Plan but user need to login .Login tab has 3 Login Admin Login, User Login, Dietitian Login.



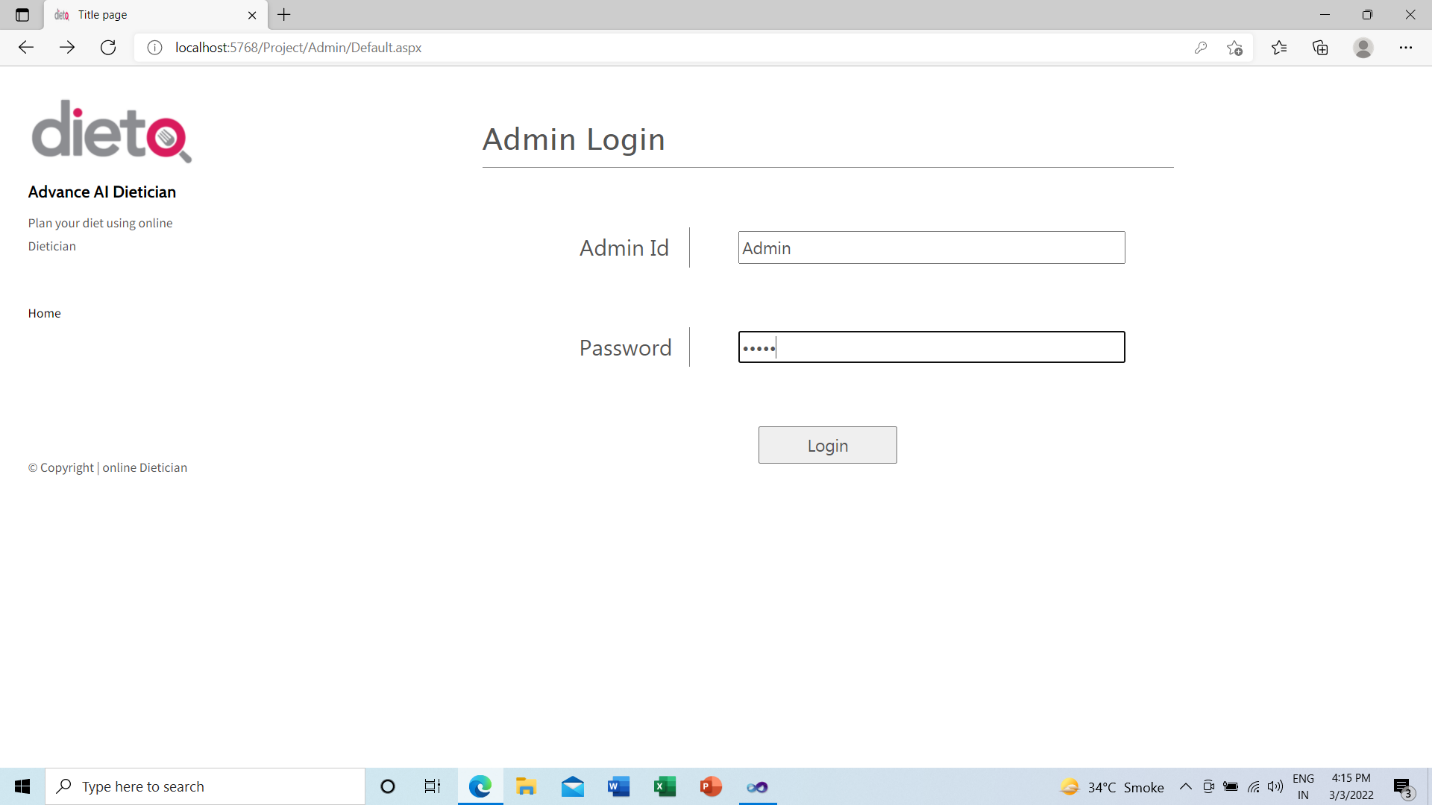
1. **Login**

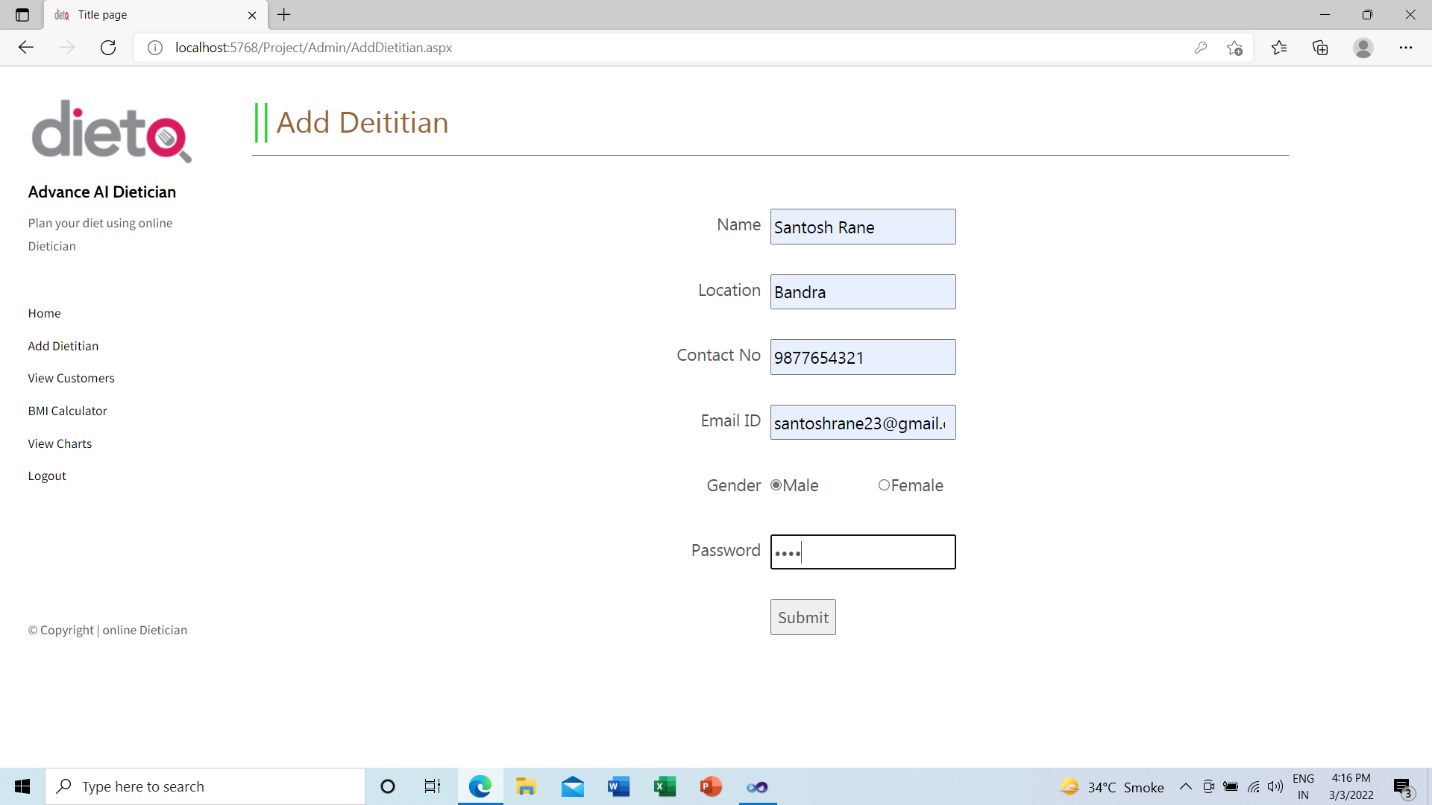
In this Page User and dietitian can Login with their respective Username and Password.

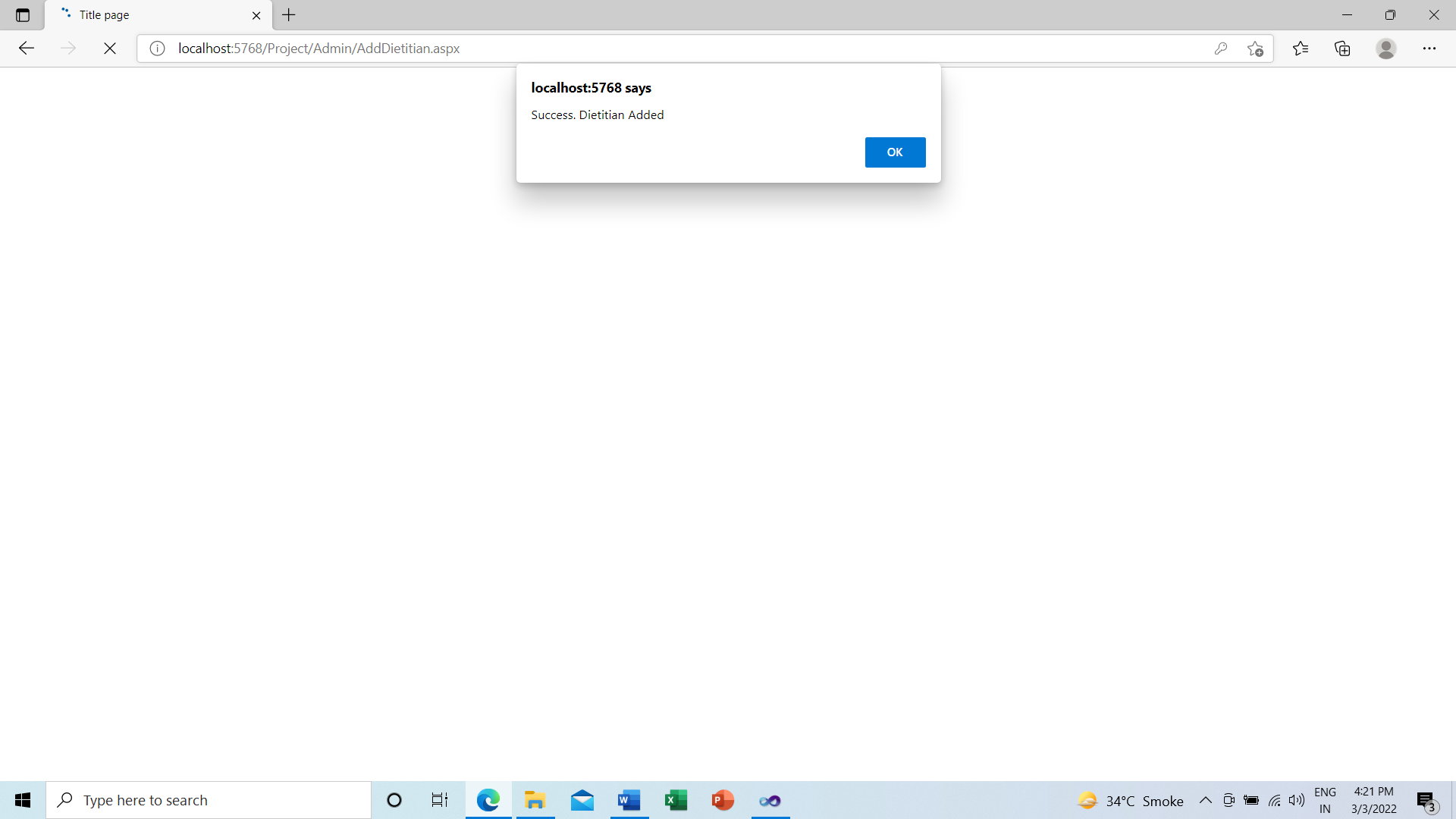


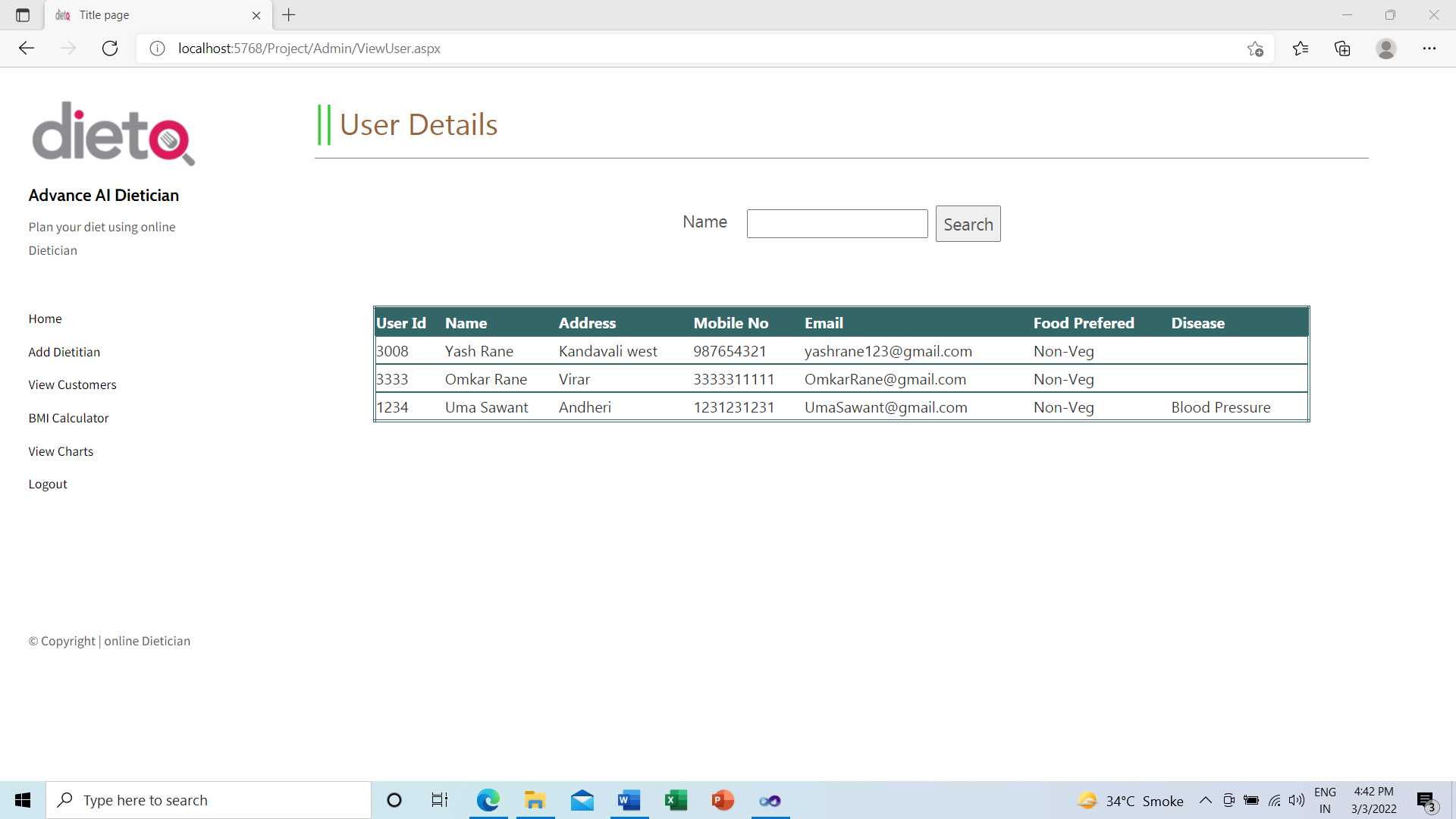
1. **Admin Login**

Admin can Add Dietitian, Admin can View Customers



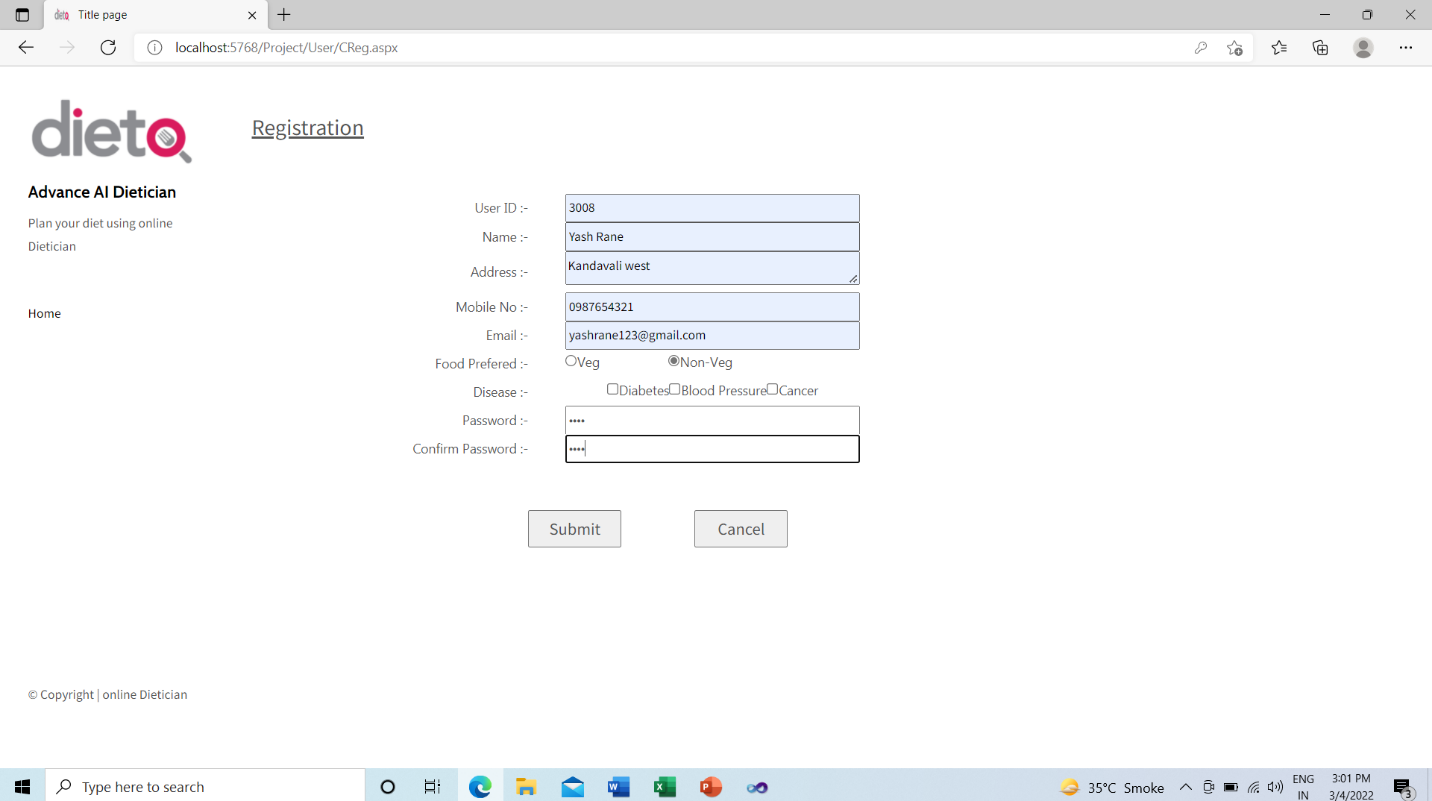


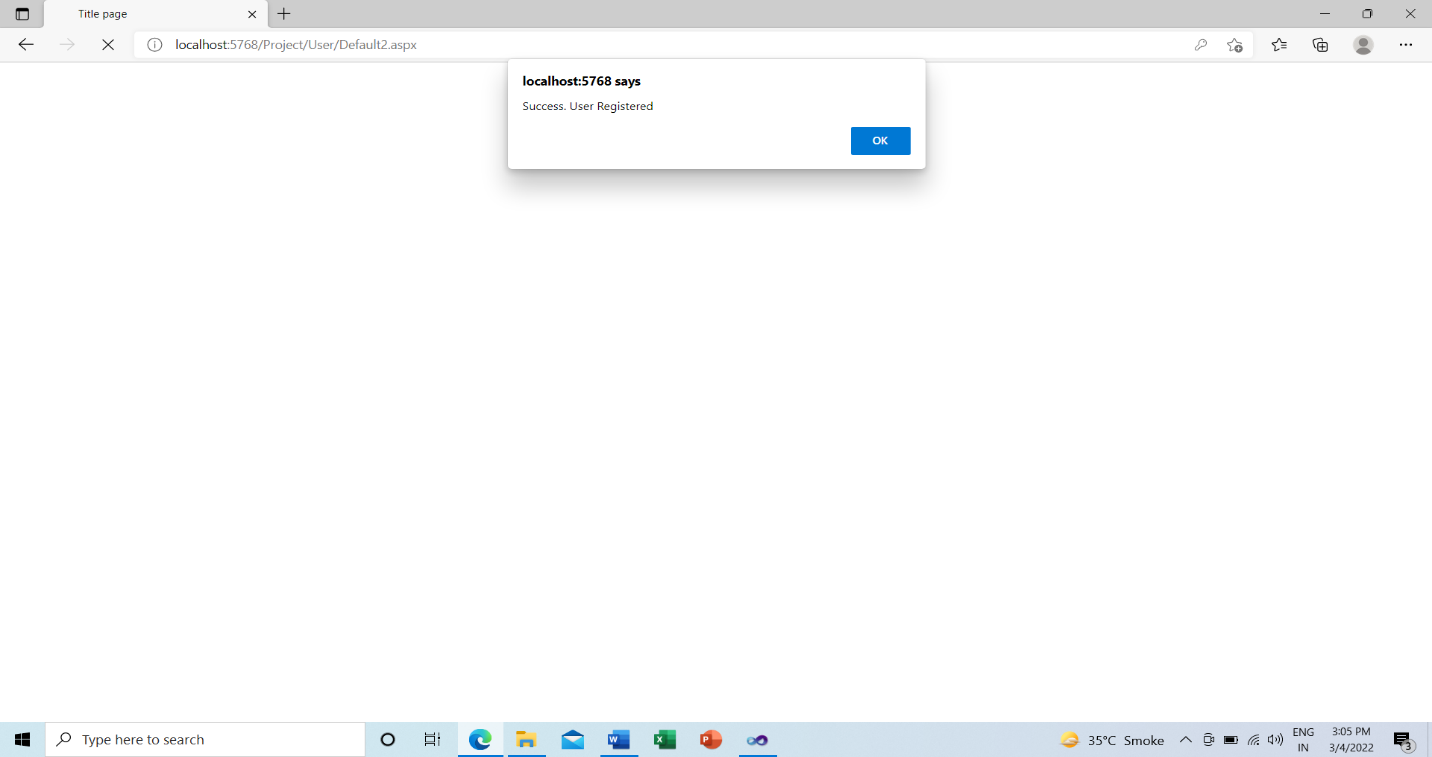




1. **User Registration Form**

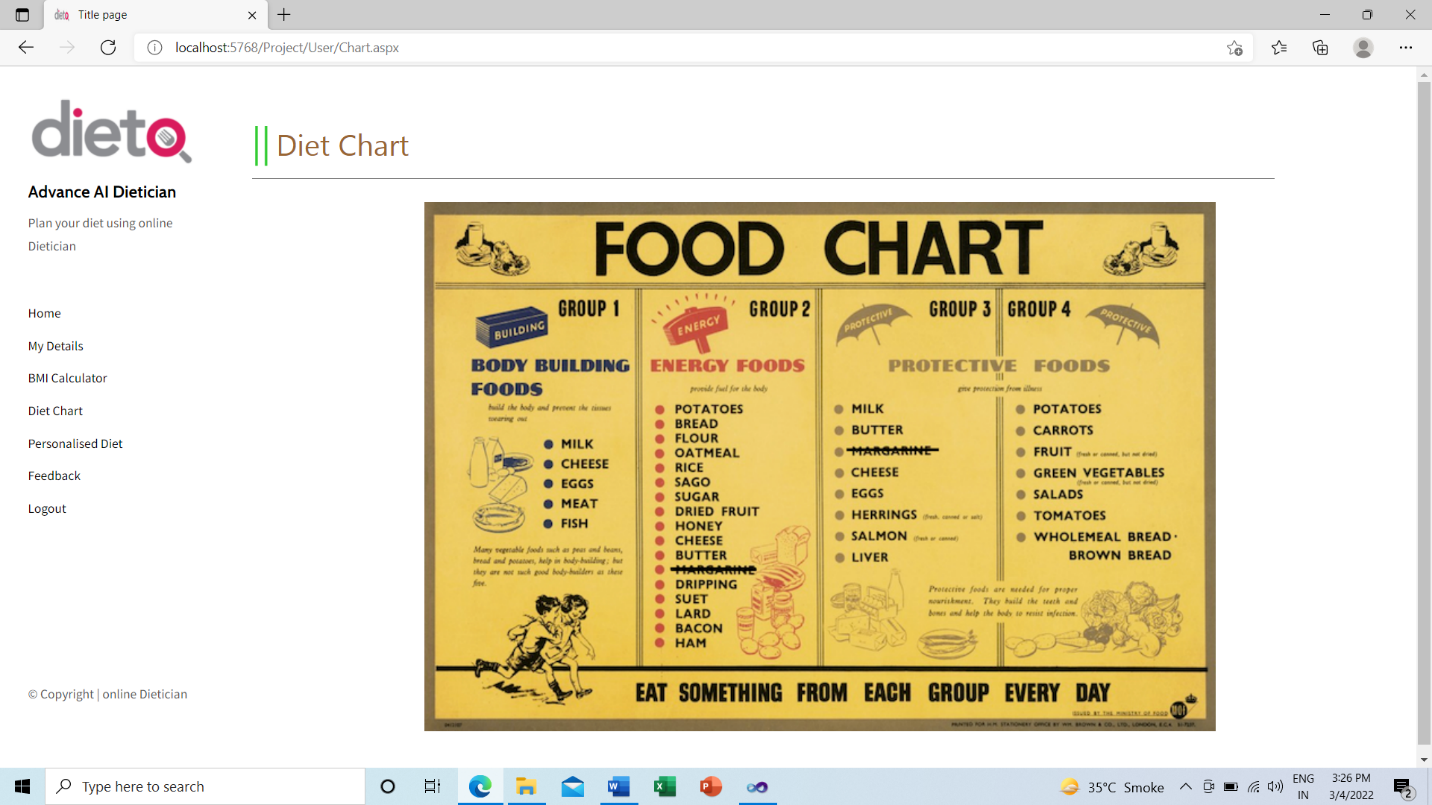
User need to enter Name, Address, Email Id, Mobile No, User need to add his/her food preference like Veg or Non-Veg. User need to add the his/her Disease. User need to add User Id and Password. User need to click on Submit Button for Registration.



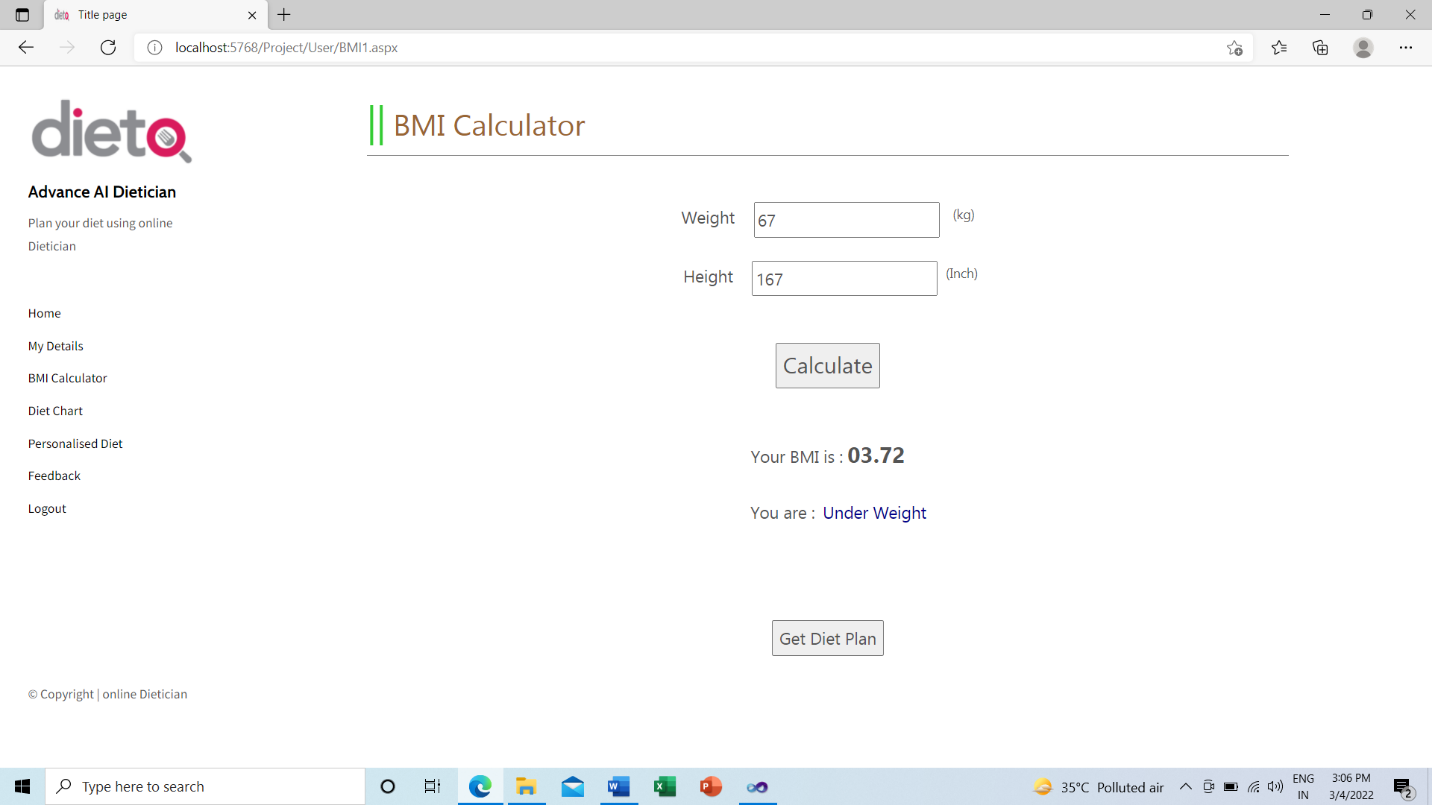


1. **User Login**

User need to Add the correct User Id and Password and click on login Button. Then User can use the this Application. User can view the Food Chart also.

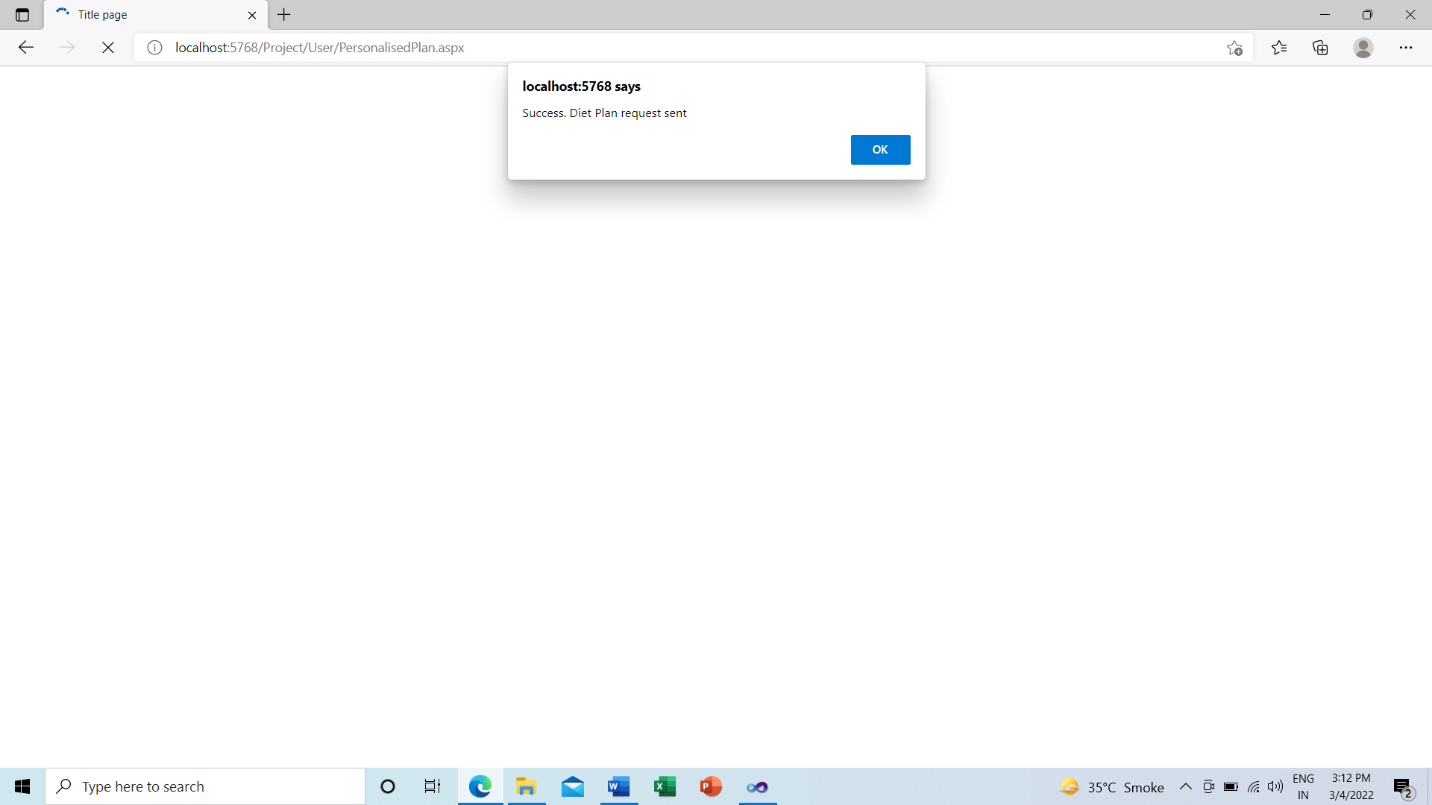
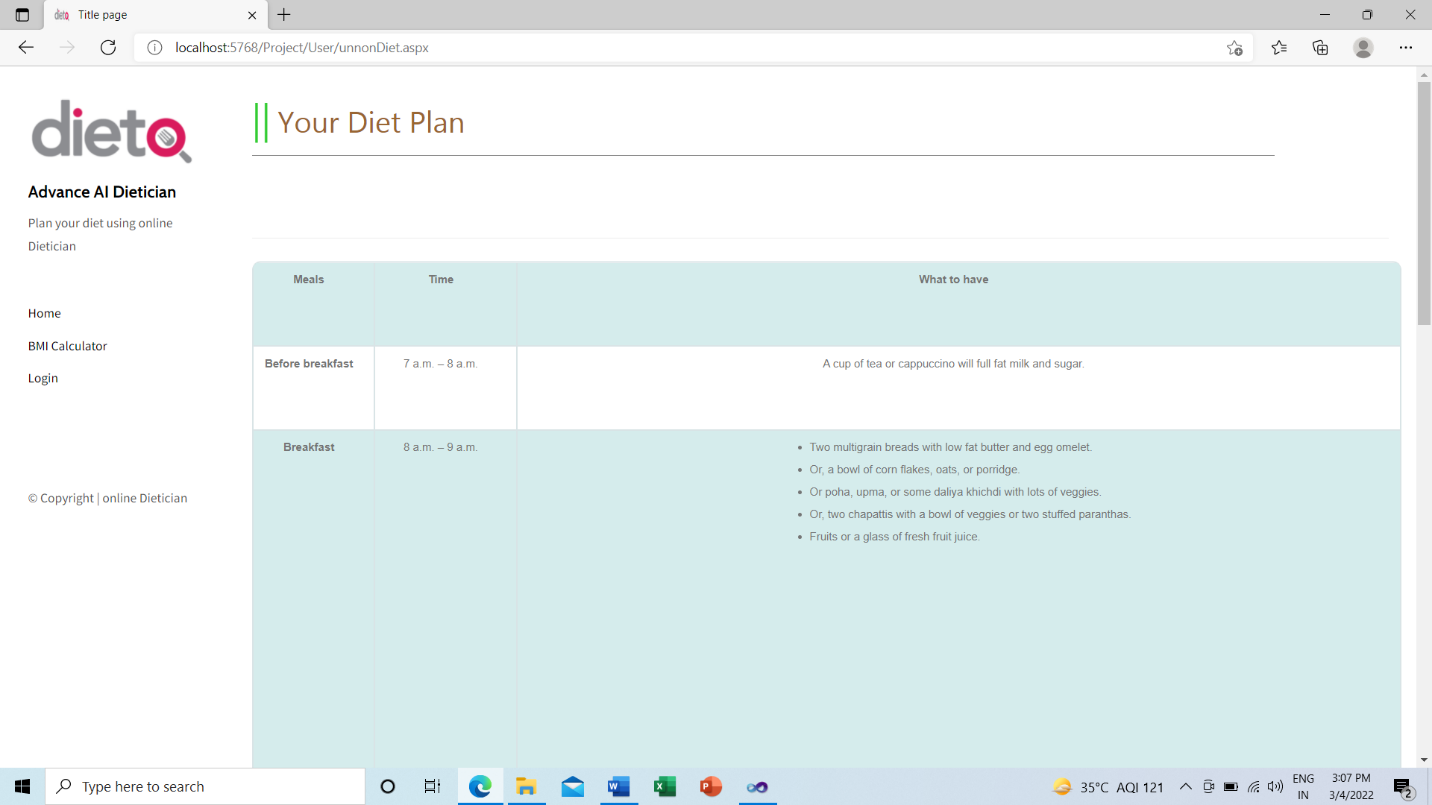
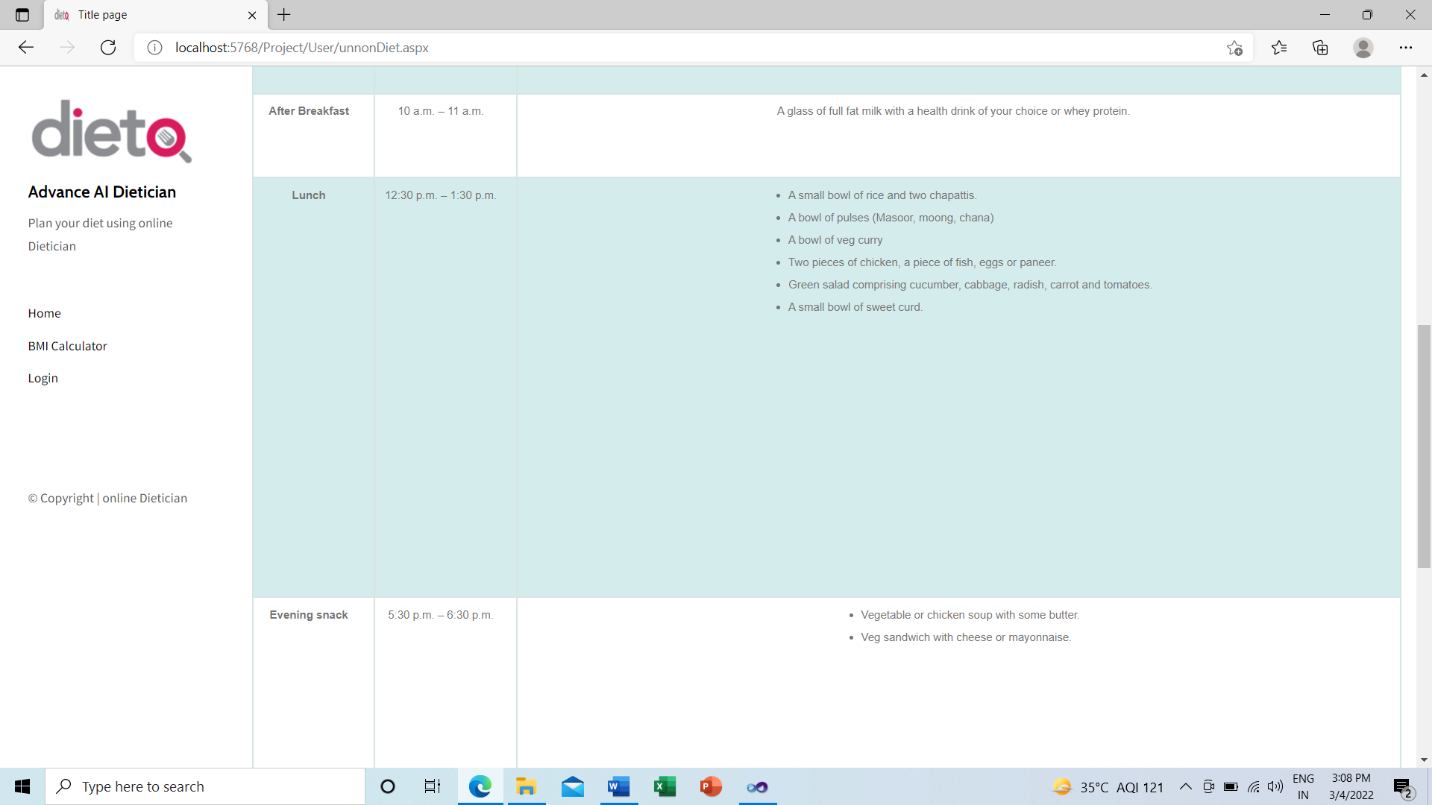
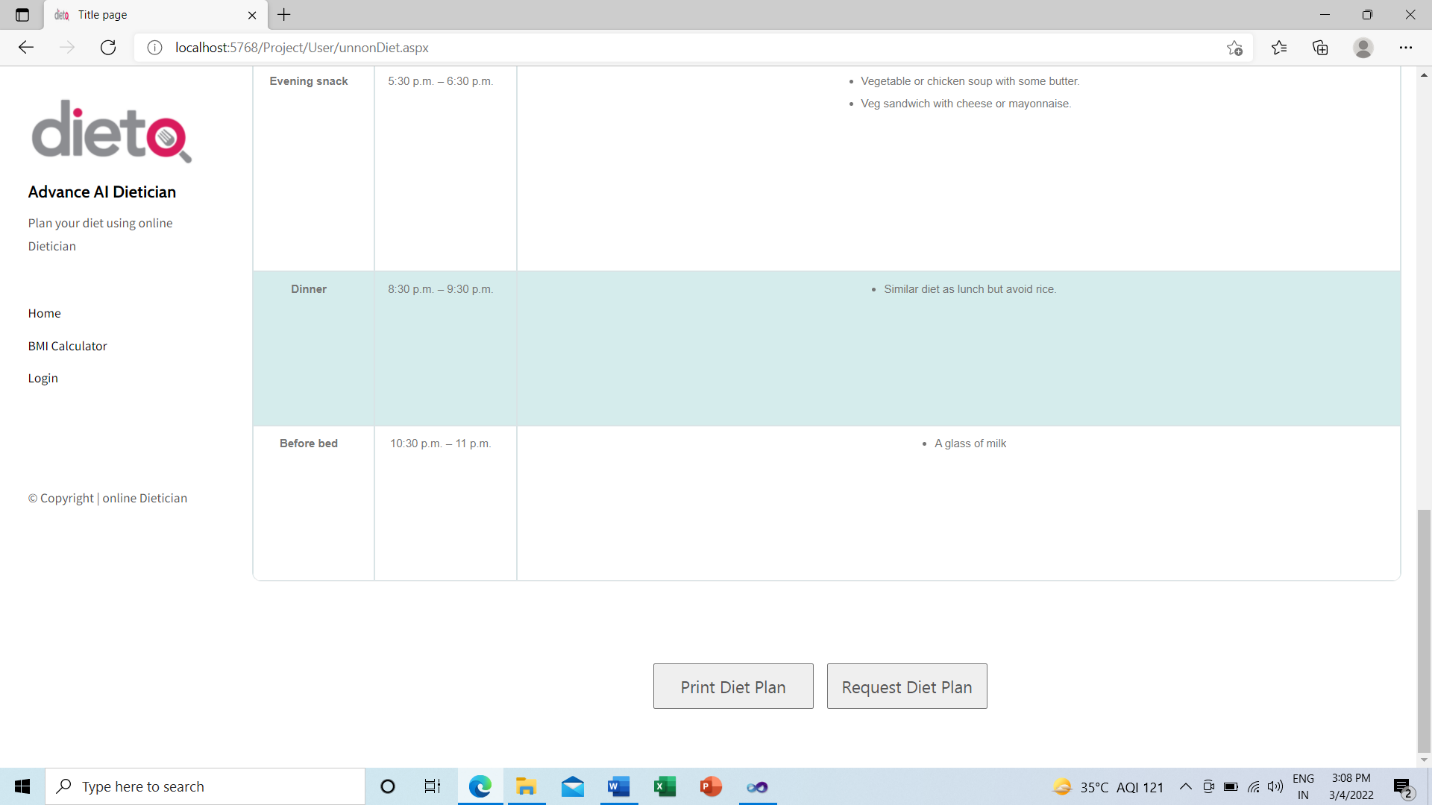
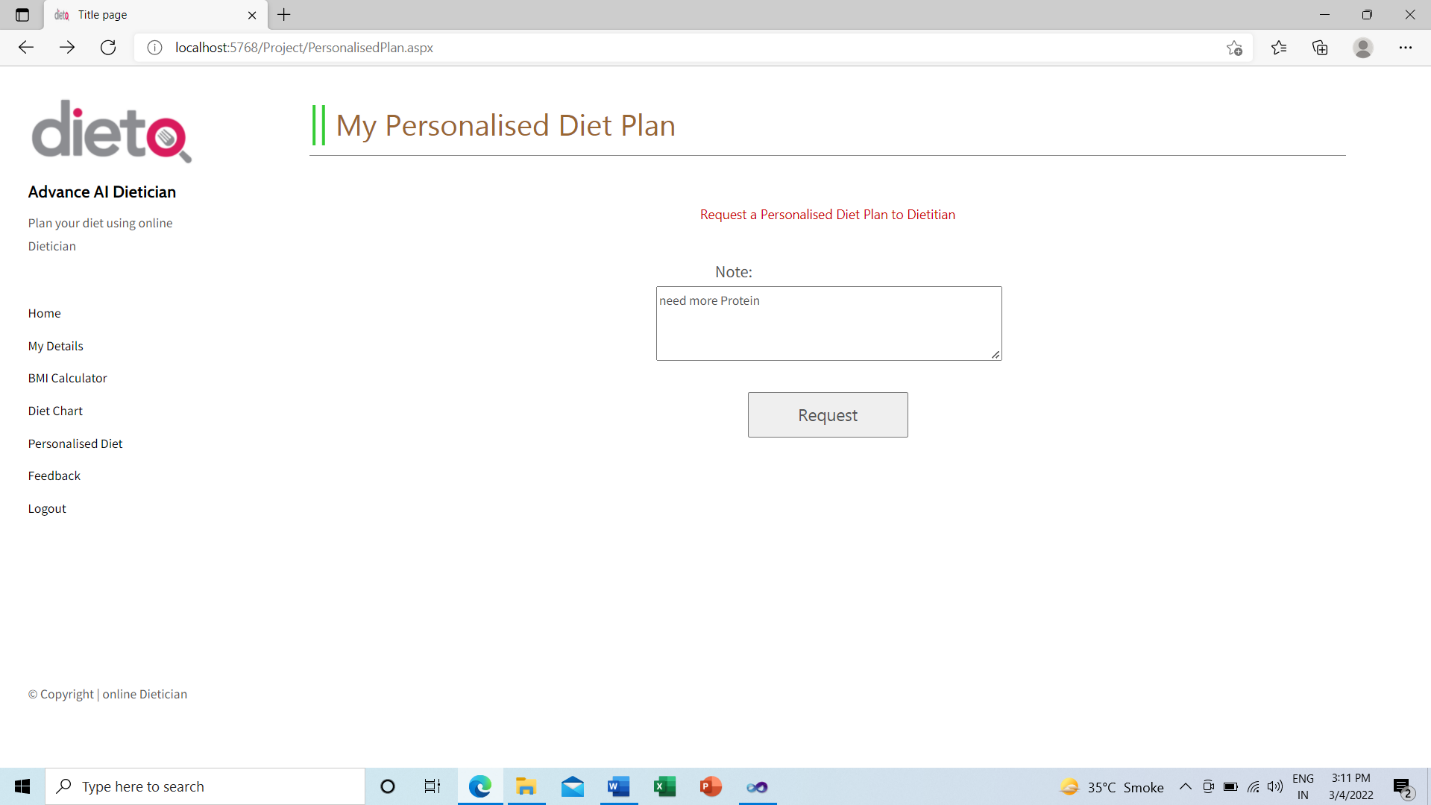


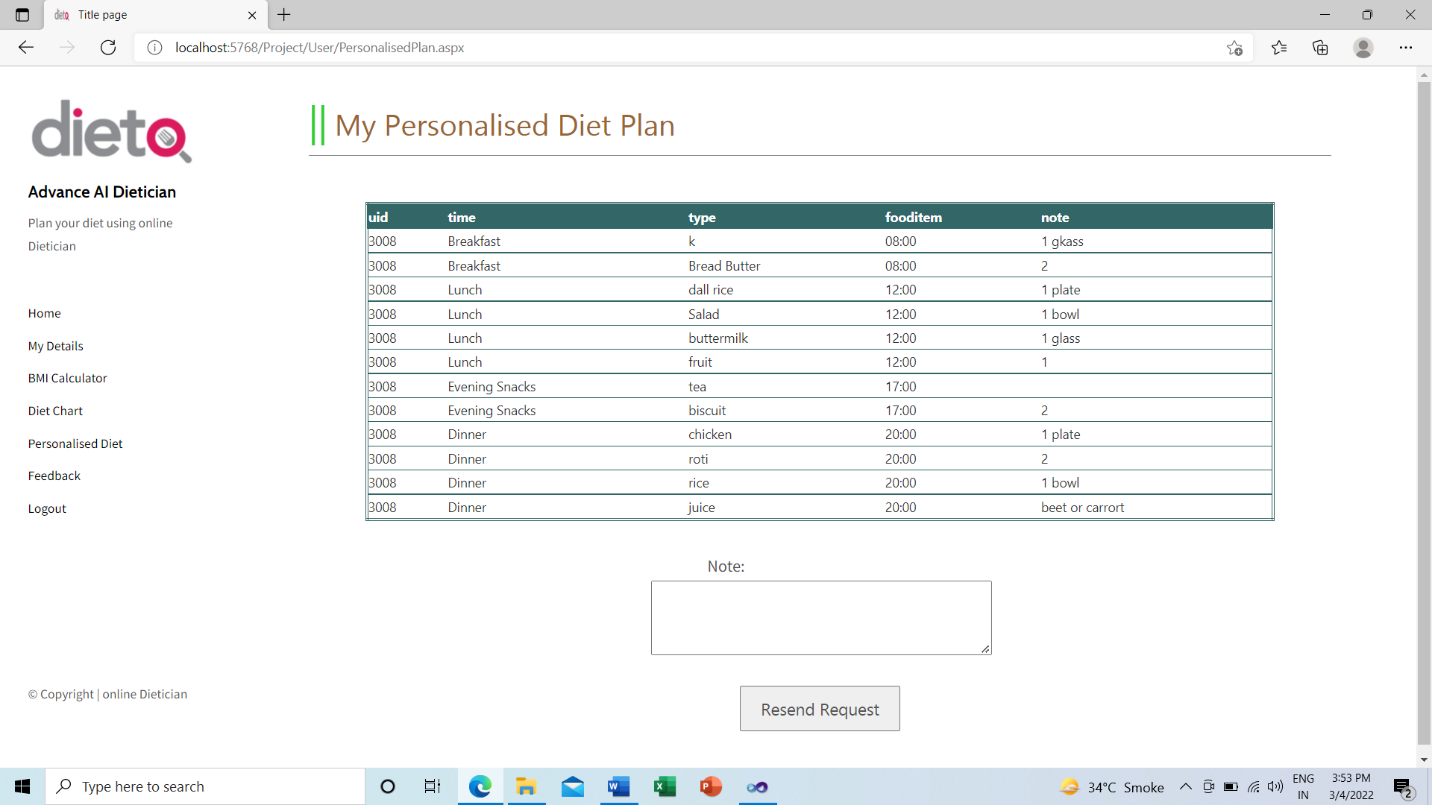
1. **BMI Calculator**

User need ton enter Correct Height and wright. Then Click on Calculation Button. Then it will show your BMI. Then User need to lick on get Diet Plan w.r.t to his/her height and weight. Then User will get the Personalized Diet plan. 

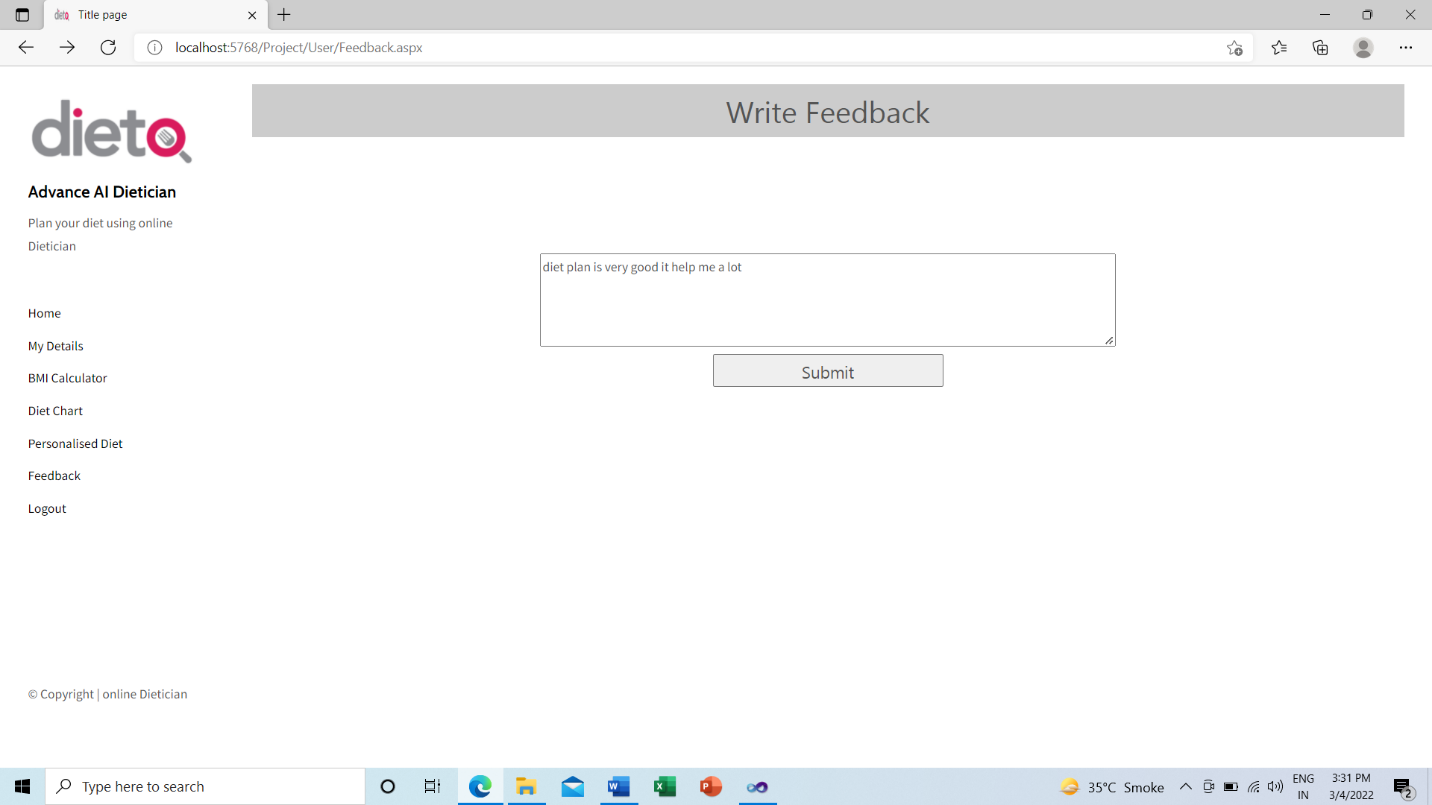
1. **Requested Diet Plan**

If User is not satisfied with given diet plan than user can request for the new Diet Plan on Clicking with the Request Diet Plan Button. User need to write his/her preference in the Note and then submit the Request to the dietitian and get the new Update Diet Plan.

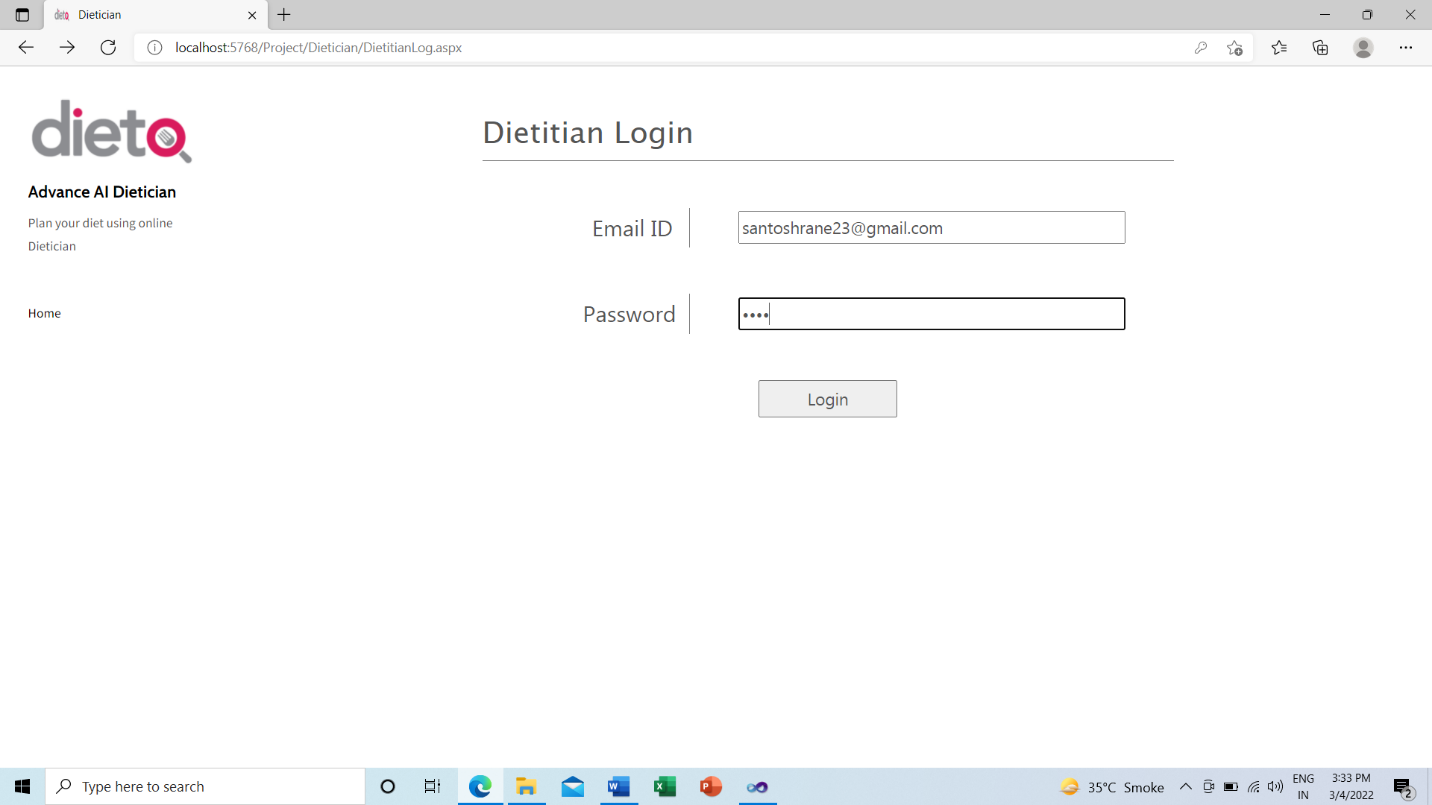




1. **Feedback Form**

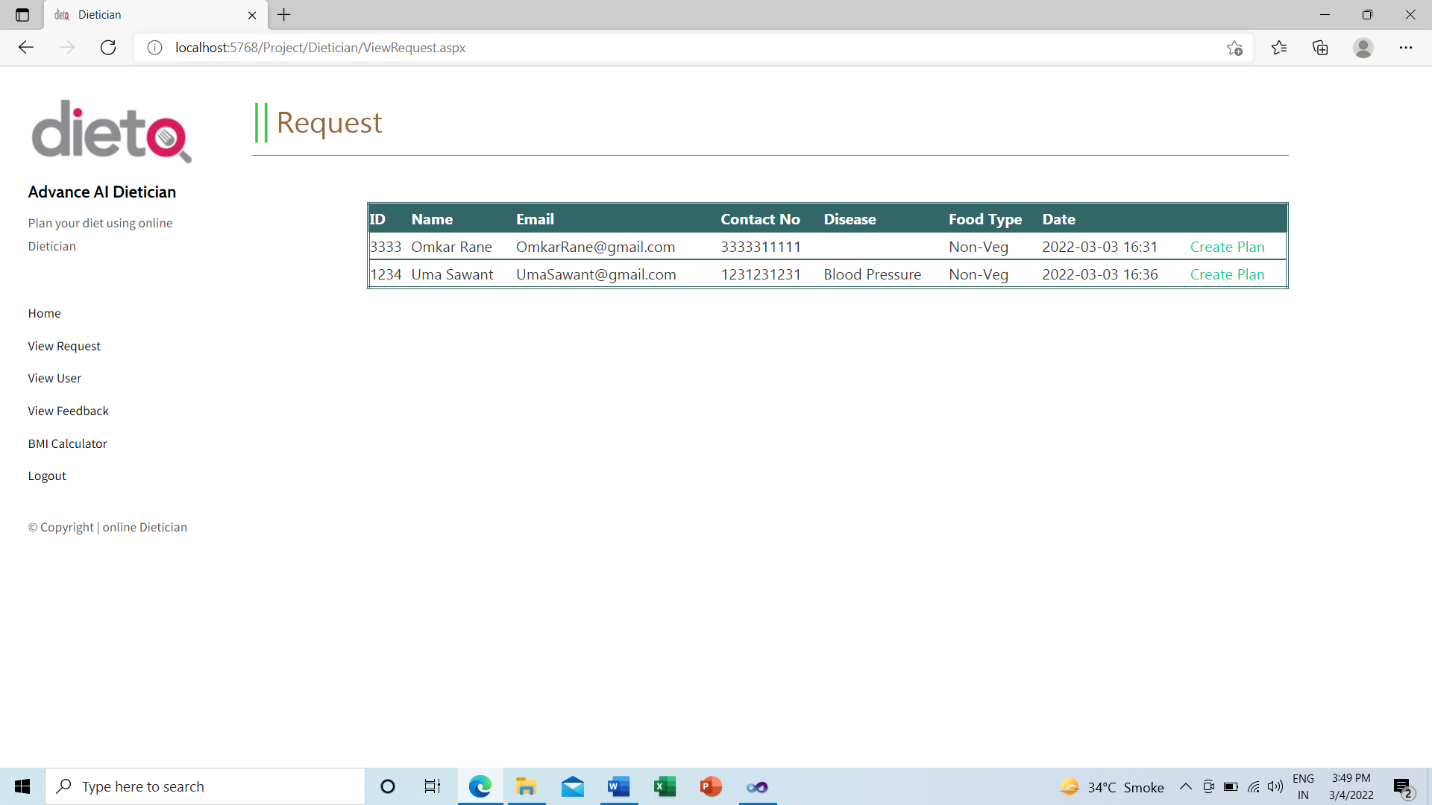
User can also write Feedback to the dietitian and submit the feedback. 

1. **Dietitian Login**

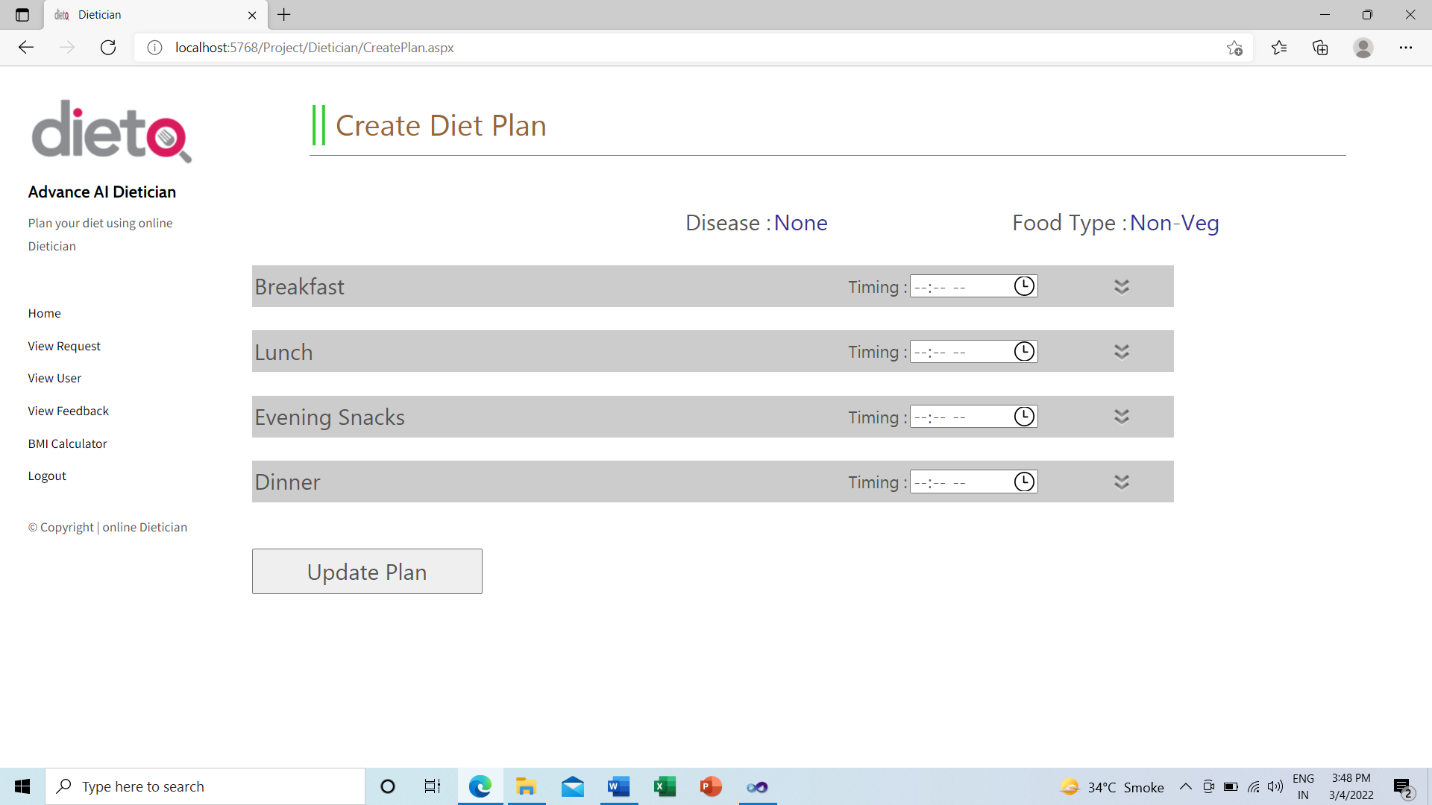
Dietitian need to enter his /her correct Email Id and Password. Then Dietitian can view the Customers /Users. View the Request from the Users. 

1. **View Requests**

Dietitian can view the Request given by the User then Dietitian need to create the new diet plan w.r.t his/her requirement and preferences.

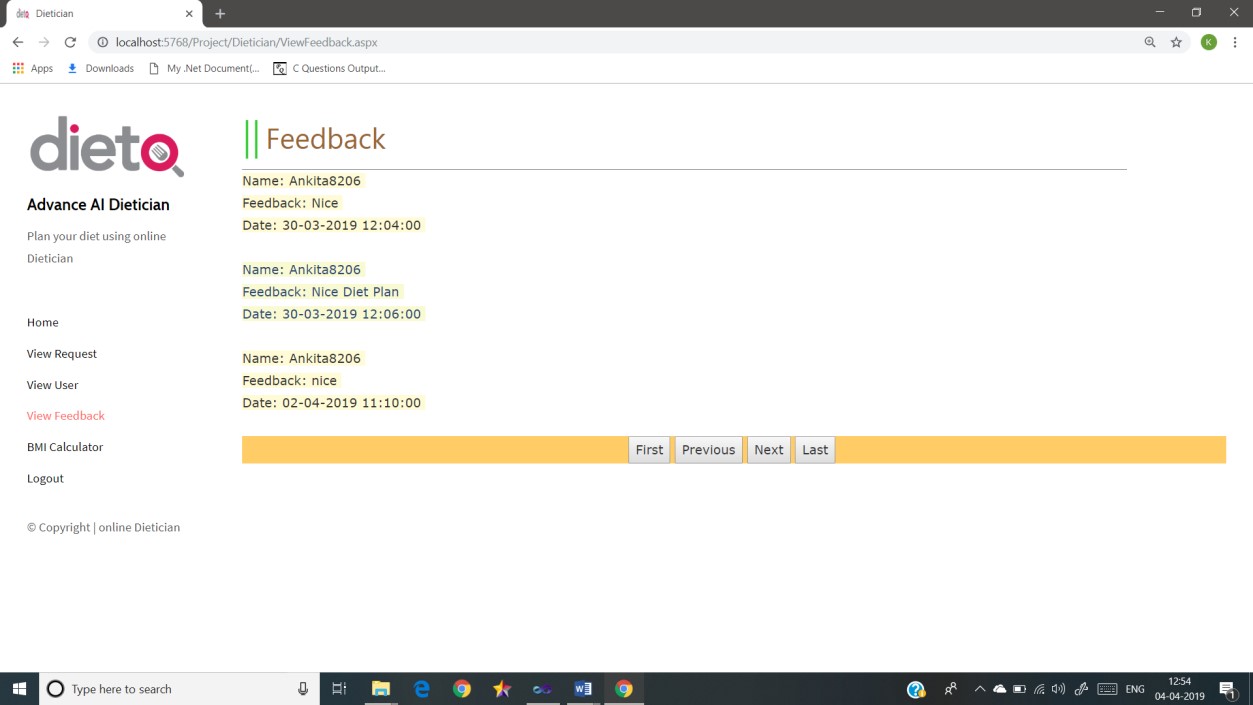


1. **Create Diet Plan**

Dietitian need to create the diet plan for breakfast, Lunch, Evening Snacks and Dinner for requested user. Then send it to the User. 

1. **View Feedback**

Dietitian can view the Feedback given from the User.

****

**CHAPTER 7**

**CONCLUSION**

**7.1 Significance of the System**

No need of consulting doctor for diet plan.

Save money and very effective and give accurate result as it is coded with keeping diet chart in mind. This system provide full details of the nutrient constitution in body and if required more or not along with the plan by just answering to some queries and user can request for alternate diet plan.

**7.2 Limitation of the System**

One has to be sure about their details while entering the fields like height weight and many more otherwise this system would give result that is not suitable for the user if not sure about what they entered.

User cannot communicate with Dietitian on the chart.

**7.3 Future Scope of Project**

**1.** Being a computer system has lots of scope. It not only efficient but effective also.

**2.** As this is computerized system the owner/Admin is completely depend on computer for accessing details about Customers.

**3.** Also the software needs to be maintained properly time to time, i.e. if it requires be updating or modifying etc.

**4.** Some more advance feature will be added for the user like User can Communication with Dietitian through Online Chart.

**5.** Mobile application should be made also for user convenience.

**6.** Interface will be made more user friendly and attractive GUI.

**Reference and biblography:**

https://nevonprojects.com/artificial-intelligence-dietician/

https://melsatar.blog/2018/02/16/the-waterfall-model-a-different-perspect

ive/

https://nevonprojects.com/advanced-ai-dietitian/

https://www.google.com/search?q=waterfall+model&rlz=1C1YTUH\_enIN9

44IN944&sxsrf=ALeKk01mi7r4\_4u3JE3UIZn5KbR8BUpTDQ:162574053079

6&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi33ta2o9PxAhUr3jgGHQtk

D3oQ\_AUoAXoECAEQAw&biw=1366&bih=625#imgrc=CniB5ucSHkVzyM

https://searchsoftwarequality.techtarget.com/definition/application-securi

ty

https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview