

# PROJECT REPORT

on

# MyCl

*Submitted by*

Thushar T *(32020802002)*

AKSHAY GOPAL (32020802011)

ASHINA MOL (32020802021)

GEORGE MATHEW (32020802025)

in partial fulfillment of the requirements for the award of

Bachelor of Computer Application degree of University of Kerala

***Department of Computer Science***

# COLLEGE OF APPLIED SCIENCE ADOOR 2021-2024



***CERTIFICATE***

Certified that this report titled *“MyCl”* is a bonafide record of the project work done by

Thushar T (32020802002)

AKSHAY GOPAL (32020802011)

ASHINA MOL (32020802021)

GEORGE MATHEW (32020802025)

Under our supervision and guidance, towards partial fulfillment of the requirements for the award of the Degree of BCA (Bachelor of Computer Applications) of

the University of Kerala

**Internal Guide Head of Department**

Asst. Prof. Femeena Jinoob Prof. Viji Balakrishnan

**Head of Institution External Guide**

Dr. Santhosh Babu K

# ACKNOWLEDGEMENT

Through this acknowledgement, we express our sincere gratitude to all the people who are associated with this project and have helped with it and made it a valuable experience. We take immense pleasure in thanking Dr. Santhosh Babu K, Principal of College of Applied Science Adoor for providing us with necessary facilities for doing this project. We express our deepest gratitude to the Head of the Department, Prof. Viji Balakrishnan, for their excellent guidance and constant encouragement towards successfully completing this project.

We wish to express our heartfelt thanks to our internal guide Asst. Prof. Afsana N, for their valuable guidance and readiness to clear all our doubts and for guiding us towards the right path to make this project a successful one. We also extend our thanks to the various people who have shared their opinion and experience through which we received information, crucial for our report.

Finally, yet important, we would like to express our heartfelt thanks to our beloved parents for their blessings, our friends/classmates for their help and wishes for the successful completion of this project.

Thushar T Akshay Gopal Ashina Mol George Mathew

# ABSTRACT

The major goal of the website is to give students more option once they graduate from high school. They receive calls from colleges all across the world for enrolling without enough information. But with the aid of this website, the students can obtain information on colleges, including their rankings, locations, photos, and user reviews, enabling users to select the colleges of their dreams. The students will be able to search the colleges by names of the college using our search query. And the other way is to use our filter option, where students may select a certain university and the state to receive all the colleges in that state and under that institution, is another approach to get information about that specific college. MyCl also provides the students to save the college details in a special bookmark section which helps in the students to compare the colleges and choose the one which is more suitable for the student.

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT i](#_bookmark0)

[ABSTRACT ii](#_bookmark1)

[TABLE OF CONTENTS iii](#_bookmark2)

[LIST OF FIGURES v](#_bookmark3)

[LIST OF TABLES vi](#_bookmark4)

[LIST OF ABBREVATIONS vii](#_bookmark5)

1. [INTRODUCTION 1](#_bookmark6)
   1. [PROJECT OVERVIEW 2](#_bookmark7)
   2. [OBJECTIVES 2](#_bookmark8)
2. [SYSTEM ANALYSIS 3](#_bookmark9)
   1. [INTRODUCTION 4](#_bookmark10)
   2. [EXISTING SYSTEM 4](#_bookmark11)
   3. [PROPOSED SYSTEM 5](#_bookmark12)
   4. [FEASIBILITY STUDY 6](#_bookmark13)
3. [SYSTEM ENVIRONMENT 8](#_bookmark14)
   1. [INTRODUCTION 9](#_bookmark15)
   2. [SYSTEM REQUIREMENTS 9](#_bookmark16)
   3. [TECHNOLOGIES USED 10](#_bookmark17)
      1. PROGRAMMING LANGAUGES 10
      2. [SERVICES/TOOLS 13](#_bookmark18)
      3. FRAMEWORKS 16
4. [SYSTEM DESIGN 17](#_bookmark19)
   1. [INTRODUCTION 18](#_bookmark20)
   2. PROCESS DESIGN 18
   3. [USE CASE DIAGRAM 20](#_bookmark21)
   4. DATA FLOW DIAGRAM 24
   5. STRUCTURE CHART 33
   6. [SYSTEM DATA DESIGN 36](#_bookmark22)
   7. USER INTERFACE DESIGN 45
5. [CODING 55](#_bookmark23)
   1. [INTRODUCTION 56](#_bookmark24)
   2. CODING SNIPPETS 56
6. SYSTEM TESTING 64
   1. [INTRODUCTION 65](#_bookmark25)
   2. CLI APPLICATION 65
   3. [WEBSITE 67](#_bookmark26)
7. SYSTEM IMPLEMENTATION 70
   1. [INTRODUCTION 71](#_bookmark27)
   2. [WEBSITE 71](#_bookmark28)
8. CONCLUSION 72
9. FUTURE ENHANCEMENT 74
10. [BIBLIOGRAPHY 76](#_bookmark29)

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Caption** | **Page No.** |
| Fig. 4.4.1 | Elements of Data Flow Diagram (DFD) | 19 |
| Fig. 4.4.2 | Context level Data Flow Diagram | 20 |
| Fig. 4.4.3 | Level 1 Data Flow Diagram | 21 |
| Fig. 4.4.4 | Level 1 Data Flow Diagram of Website - User | 22 |
| Fig. 4.4.5 | Level 2 Data Flow Diagram of Website - College | 23 |
| Fig. 4.4.6 | Level 3 Data Flow Diagram of Website - Admin |  |
| Fig. 4.7.7 | Screenshot of Website’s home page | 32 |
| Fig. 4.7.8 | Screenshot of login page in website | 32 |
| Fig. 4.7.9 | Screenshot of registration in website | 33 |
| Fig. 4.7.10 | Screenshot of user dashboard | 33 |
| Fig. 4.7.11 | Screenshot of admin dashboard | 34 |
| Fig. 4.7.12 | Screenshot of college dashboard | 34 |
| Fig. 4.7.13 | Screenshot of college details entering form | 35 |
| Fig. 4.7.14 | Screenshot of college requests | 35 |
| Fig. 4.7.15 | Screenshot of search form | 36 |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Caption** | **Page No.** |
| Table 4.6.1 | Structure of video table | 24 |
| Table 4.6.2 | Structure of calorie table | 25 |
| Table 4.6.3 | Structure of diet chart | 25 |
| Table 4.6.4 | Structure of diet table | 26 |
| Table 4.6.5 | Structure of exercise table | 26 |
| Table 4.6.6 | Structure of feedback table | 27 |
| Table 4.6.7 | Structure of login table | 27 |
| Table 4.6.8 | Structure of registration table | 28 |
| Table 4.6.9 | Structure of request dietplan table | 29 |
| Table 4.6.10 | Structure of medical table | 29 |
| Table 4.6.11 | Structure of dietitan table | 30 |
| Table 4.6.12 | Structure of nurtition table | 30 |
| Table 4.6.13 | Structure of chat table | 31 |

# LIST OF ABBREVATIONS

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Abbrevation** | **Full Form** |
| 1 | Asst. Prof. | Assistant Professor |
| 2 | Prof. | Professor |
| 3 | HTML | HyperText Markup Langauge |
| 4 | CSS | Cascading Style Sheet |
| 5 | JS | JavaScript |
| 6 | PHP | PHP Hypertext Preprocessor |
| 7 | SQL | Structured Query Language |
| 8 | DFD | Data Flow Diagram |

# 1. INTRODUCTION

## PROJECT OVERVIEW

The BEFIT is a web based application intended for online users. The main objective of this application is to make it interactive and easy to use. This application is helpful in finding diet plans and calorie counting. The main motive of this project is to find the finding diet plans and calorie counting , dietician , neutrian very easily and time saving. Admin is the main authority of the project. The admin can add nutrian, dietician details. The user can view the exercise and diet plan details added by the dietician. He or she can also view food calories chart. The user can request for their own diet plan. The dietician can view the diet plan request send by the user. He or she can add the diet plan chart to the specified user. The user will get the diet chart. The dietician also add the food calorie details. User can add the payment details and send a message on neutrian and view its reply. The neutrian can view the messages and send a reply.

## OBJECTIVES

* + 1. Provide the users with an easy to use application that can help them manage software from external sources (websites, GitHub).
    2. Provide the users a website with a list of available dietician, diet plan information.
    3. Provide the functionality for users to give message for neutrian in the form of chat.
    4. Give the users helpful documentation on how to use the application to manage diet plan.
    5. Provide the users with a forum where they can interact with the neutiran

# SYSTEM ANALYSIS

## INTRODUCTION

A system is simply a set of components to accomplish an objective. Developing a new system, investigating into the operation and making possible changes in the existing system are called System Analysis. Analysis comprises a detailed study of the various operations performed by a system and their relationships within and outside the system. It is the process of gathering and interpreting facts, diagnosing problems and improving the system using the information obtained.

The objectives of System Analysis include the following

* + - Identifying the user’s needs.
    - Performing economic and technical analysis.
    - Establishing cost and schedule constraints.

Here the system analyst should study a system with an eye on solving the problem using computers. It is an essential part for the development of a project by a system analyst. System analysis is for finding out what happens in the existing systems, deciding on what changes and new features are required and defining exactly what the proposed system must be. This process of system analysis is largely concerned with determining, developing and agreeing to the user’s requirements. It provides prime opportunities to communicate well with the user and conceive a joint understanding of what a system should be doing, together with a view of the relative importance of the system facilities using interactive techniques.

To analyze a system, one has to study the system's work in detail. The system analyst has to understand the functioning and concept of the system in detail, before designing the appropriate computer based system that will meet all the requirements.

## EXISTING SYSTEM

The web application is able to allow end users or requestors to sign up for an account. This requires a lot of hard work and time consumption to complete the task. This may include human errors. In the existing system, it is difficult to retrieve some particular information. Also, all the records are stored manually and it is a tedious task. As a result the security of these records is always a challenging task. In spite of all the efforts undertaken, the destruction of data may happen often. Hence, the computerization of the system of record maintenance is the only solution to reduce the shortcomings of the existing system

## Disadvantages

* Chance of loosing information.
* Need a large volume of storage area.
* Reports could not be prepared in time.
* Use and maintenance of paper files, registers and other written matter was very difficult.
* Security of the system depended on the person dealing with it.
* Searching of information was time consuming.
* More man power required.
* Data verification and validation was manual.
* Redundancy of records may occur.

## PROPOSED SYSTEM

This application , to book an dietician t need to visit users for their diet plan and food chart. As a result, most of the process consumes lots of time and man power. The software ‘BEFIT’ will allow speeding up the works of users. The main objective of this application is to make it interactive and easy to use. This application is helpful in finding diet plans and calorie counting. The main motive of this project is to find the finding diet plans and calorie counting , dietician , neutrian very easily and time saving.

## Advantages

* Reduces time consumption.
* Avoids unauthorized access on data.
* Redundancy can be avoided.
* Generating food chart
* Generating diet plan chart
* Generate chat for neutrian .

## FEASIBILITY STUDY

A preliminary investigation examines project feasibility. Feasibility study is a small scale system analysis. It is necessary as it evaluates the feasibility of a project at the earliest possible time.

Types of feasibility study:

* + - Technical feasibility
    - Operational feasibility
    - Functional feasibility
    - Economical feasibility
    - Social feasibility

## Technical feasibility

It is the study of resource availability that may affect the ability to achieve an acceptable system. The system must be evaluated from the technical viewpoint first. The assessment of this feasibility must be based on an outline design of the system requirements in terms of input, output, program procedure, etc. Having identified the outline of the system, the investigation must go on to suggest the type of equipment, required method of developing the system, and the method of running the system. The outcome of the study was found to be positive.

## Operational feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The outcome of the study was satisfactory.

## Functional feasibility

Here we examine the functions of the system which may work properly when implemented. The proposed system has functions that can be implemented successfully. Hence the project was found to be functionally feasible.

## Economical feasibility

It is considered as the final stage of most systems, it includes a broad range of concerns that include cost benefit analysis. The proposed system was found to be economically feasible as its requirements did not require huge expenditure, the group also had the knowledge to undertake this task without any difficulty.

## Social feasibility

Social feasibility is a detailed study on how one interacts with others within a system or an organization. Social impact analysis is an exercise aimed at identifying and analyzing such impacts in order to understand the scale and reach of the project’s social impacts. This project has a great impact on the adoption of Linux on PCs by making software more accessible and easy to access and maintain.

# SYSTEM ENVIRONMENT

## INTRODUCTION

A system environment refers to the collection of hardware, software, and data that make up a computer system. This includes the physical components of a computer such as the CPU, memory, storage devices, and input/output devices, as well as the software applications and operating system that run on it. The system environment also includes the network connections and protocols used to communicate with other devices and systems. Understanding the system environment is crucial for troubleshooting, optimizing performance, and developing software applications that work effectively in a particular system environment.

## SYSTEM REQUIREMENTS

* + 1. **Website**

Server requirements:

Hardware:

Processor: Any modern dual core processor or greater RAM: 512MB or greater

Storage: 512MB of free space or greater Software:

Operating system: Windows Database: MySQL

Web server: Apache

Programming languages: PYTHON DJANGO

Client requirements:

Any device with an active internet connection and a browser with HTML5

support

## TECHNOLOGIES USED

* + 1. **Programming languages**
       1. **Python**

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamic typed language. For example, x=10, here x can be anything such as String, into etc.

## Features in Python

There are many features in Python, some of which are discussed below –

## Easy to code:

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

## Free and Open Source:

Python language is freely available at official website and you can download it. Since, it is open-source; this means that source code is also available to the public. So you can download it as, use it as well as share it.

## Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

## GUI Programming Support:

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, python or Tk in python.PyQt5 is the most popular option for creating graphical apps with Python.

## High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

## Extensible feature:

Python is an Extensible language. We can write some python code into c or c++ language and also we can compile that code in c/c++ language.

## Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

## Python is integrated language:

Python is also an integrated language because we can easily integrated python with other language like c, c++ etc.

## Interpreted Language:

Python is an Interpreted Language. Because python code is executed line by line at a time. Like other language c, c++, java etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called byte code.

## Large Standard Library:

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

## Dynamically Typed Language

Python is dynamically-typed language. That means the type (for example- int, double, long etc.) for a variable is decided at run time not in advance. Because of this feature we don’t need to specify the type of variable

.

## Django

Django is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) based [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [web framework](https://en.wikipedia.org/wiki/Web_framework), which follows the model- template-view (MTV) [architectural pattern.](https://en.wikipedia.org/wiki/Architectural_pattern_(computer_science)) Django's primary goal is to ease the creation of complex, database driven websites. The framework emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability) and plug ability of components, less code, low coupling, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself). Python is used throughout, even for settings files and data models. Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create%2C_read%2C_update_and_delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Type_introspection) and configured via admin models.

Despite having its own nomenclature, such as naming the callable objects generating the [HTTP](https://en.wikipedia.org/wiki/HTTP) responses views, the core Django framework can be seen as an [MVC](https://en.wikipedia.org/wiki/Model-view-controller) architecture. It consists of an [object-relational mapper](https://en.wikipedia.org/wiki/Object-relational_mapping) (ORM) that mediates between [data models](https://en.wikipedia.org/wiki/Data_modeling) (defined as Python classes) and a [relational database](https://en.wikipedia.org/wiki/Relational_database) (Model), a system for processing HTTP requests with a [web templating](https://en.wikipedia.org/wiki/Web_template_system) [system](https://en.wikipedia.org/wiki/Web_template_system) (View), and a [regular-expression](https://en.wikipedia.org/wiki/Regular_expression)-based [URL](https://en.wikipedia.org/wiki/Uniform_Resource_Locator) dispatcher (Controller).

Also included in the core framework are:

* A lightweight and standalone web server for development and testing.
* A form serialization and validation system that can translate between [HTML](https://en.wikipedia.org/wiki/HTML) forms and values suitable for storage in the database.
* A template system that utilizes the concept of [inheritance](https://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming)) borrowed from object-oriented programming.
* A [caching](https://en.wikipedia.org/wiki/Web_cache) framework that can use any of several cache methods
* Support for [middleware](https://en.wikipedia.org/wiki/Middleware) classes that can intervene at various stages of request processing and carry out custom functions
* An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signal.
* An [internationalization](https://en.wikipedia.org/wiki/Internationalization_and_localization) system, including translations of Django's own components into a variety of languages
* A [serialization](https://en.wikipedia.org/wiki/Serialization) system that can produce and read [XML](https://en.wikipedia.org/wiki/XML) and/or [JSON](https://en.wikipedia.org/wiki/JSON) representations of Django model instances
* A system for extending the capabilities of the template engine
* An interface to Python's built-in [unit test](https://en.wikipedia.org/wiki/Unit_test) framework
* Django REST framework is a powerful and flexible toolkit for building Web APIs

## HTML

HTML, or Hypertext Markup Language, is a markup language used for creating web pages and applications. HTML provides the structure and content of a web page, defining headings, paragraphs, links, images, and other elements that make up the page. It is often assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. HTML is a cornerstone technology for web development and is essential for creating any kind of web page or application. HTML is easy to learn and understand, making it accessible to developers of all skill levels.

## CSS

CSS, or Cascading Style Sheets, is a style sheet language used to describe the presentation and styling of HTML documents. CSS provides the means to control the layout, typography, colors, and other visual aspects of a web page, allowing developers to create visually appealing and engaging websites. CSS is an essential component of web development, working in conjunction with HTML to create a seamless and visually appealing user experience.

## JavaScript

JavaScript is a popular programming language that is used to create interactive web applications and dynamic user interfaces. Developed by Netscape in 1995, JavaScript has become one of the most widely used programming languages in the world. JavaScript is a client-side scripting language, which means that it runs in the user's web browser and can be used to modify the content and behavior of a web page. JavaScript is an essential tool for web developers, enabling them to create engaging and interactive web applications that respond to user input and events. JavaScript is also used extensively in web development frameworks and libraries, such as React and Angular, which provide a range of pre-built components and tools for building complex web applications. The popularity of JavaScript has led to a large community of developers who contribute to its development, making it a powerful and versatile language that is constantly evolving to meet the needs of modern web development.

The project has made use of the following external JavaScript libraries:

* + 1. Ajax

Ajax (Asynchronous JavaScript and XML) is a web development technique that allows data to be retrieved from a server without reloading the entire page. Ajax enables websites to be more responsive and interactive, as it allows for data to be loaded in the background while the user continues to interact with the page. This technique involves using a combination of JavaScript and XML (or JSON) to send and receive data between the client and server, without disrupting the user experience.

* + 1. jQuery

jQuery is a fast, small, and feature-rich JavaScript library that simplifies HTML document traversal and manipulation, event handling, and AJAX. It is

designed to make client-side scripting of HTML easier, and it is widely used for creating interactive web applications. jQuery is an open-source library that supports a variety of browsers, and its easy-to-use syntax makes it a popular choice for web developers.

## Services/Tools

1. **Apache Web Server**

Apache Web Server is a popular open-source web server software that is used to host and serve web pages and other content over the internet. It is highly configurable, scalable and secure, and can run on various operating systems, including Linux, Windows, and macOS. Apache Web Server is widely used and popular because of its simplicity, flexibility, and extensive documentation. Apache Web Server is also popular for its support of multiple programming languages such as PHP, Perl, and Python, making it a great choice for developers who want to build dynamic web applications.

## MySQL

Relational database systems are the most important database systems used in the software industry today. One of the most outstanding systems is MySQL.

The important aspects of SQL Server are:

* + MySQL is easy to use.
  + Embedded database library.
  + Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) open source web application software stack (and other "[AMP](https://en.wikipedia.org/wiki/List_of_AMP_packages)" stacks). LAMP is an acronym for "[Linux](https://en.wikipedia.org/wiki/Linux), [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](https://en.wikipedia.org/wiki/Perl)/[PHP](https://en.wikipedia.org/wiki/PHP)/[Python](https://en.wikipedia.org/wiki/Python_(programming_language))." [Free-software](https://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL.

On all platforms except Windows, MySQL ships with no [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](https://en.wikipedia.org/wiki/Command_line) tools, or install [MySQL workbench](http://dev.mysql.com/downloads/workbench/) via a separate download. Many third-party GUI tools are also available.

# 4.SYSTEM DESIGN

## INTRODUCTION

System design is a crucial phase in the software development life cycle where a high-level conceptual design is created for the proposed system. It involves defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. The objective of system design is to translate the requirements gathered during the analysis phase into a detailed design that can be implemented by the developers. A well-designed system is essential for ensuring that the final product meets the user's expectations, is maintainable, scalable, and can be easily modified or updated in the future. A thorough system design is also important for estimating the development effort, cost, and resources required to build the system.

## PROCESS DESIGN Modules

* + 1. Website
       1. User
          1. Dietician information page

View all dieticians

Send application for dietician

View application status

View diet chart

View food chart

* + - * 1. Available nutrition page

View all nutrition

Send message for nutrition

View reply from nutrition

* + - * 1. Payment
      1. Admin
         1. Website statistics page
         2. Manage dietician
         3. Manage nutrition
         4. View users
         5. View diet report for users
         6. View payment
      2. Dietician
         1. View applications
         2. Manage diet plan
         3. Send report to admin
         4. View food calories
         5. Add food chart
         6. Add exercise
      3. Nutrition
         1. View message
         2. Send reply for message.
    1. Forum
       1. User
          1. Create/View applications

Create/View application and status

View diet plan

Add medical details

Chat details

## DATA FLOW DIAGRAM (DFD)

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow,  there are no decision rules and no loops.

For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes. The data-flow diagram is a tool that is part of structured analysis and data modeling.

The basic elements of Data Flow Diagram are :

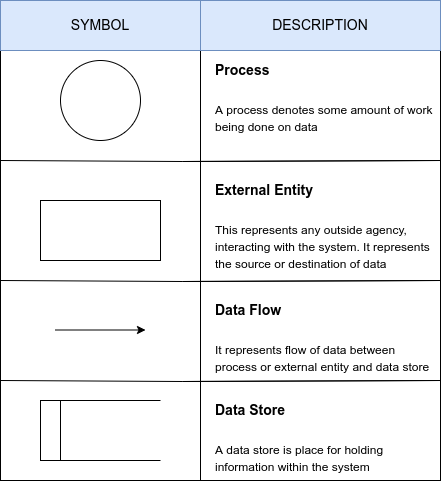


Fig 4.4.1 Elements of Data Flow Diagram (DFD)

## BEFIT

Context level

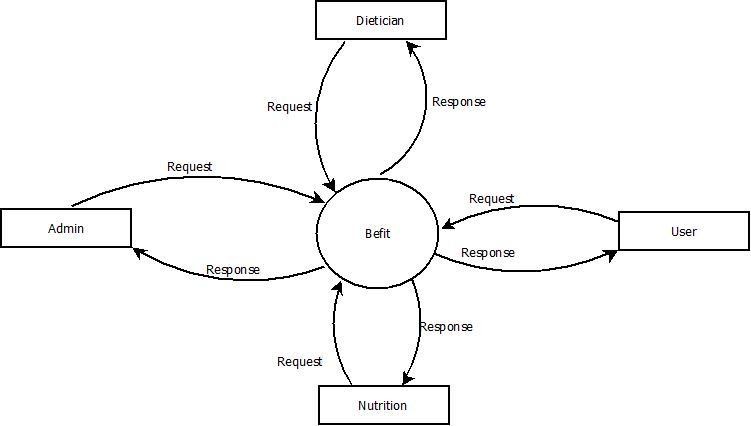


Fig 4.4.2 Context level Data Flow Diagram of BEFIT

Level 1

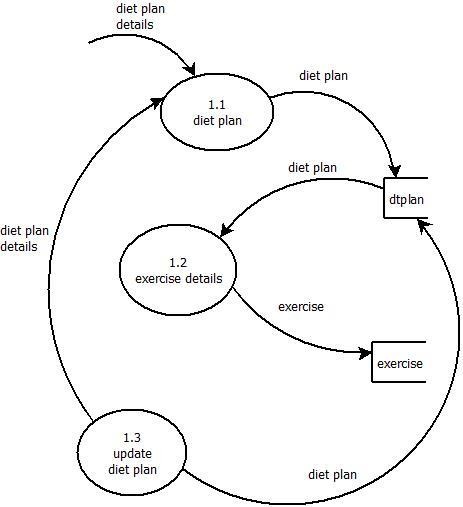


Fig 4.4.3 Level 1 Data Flow Diagram of BEFIT

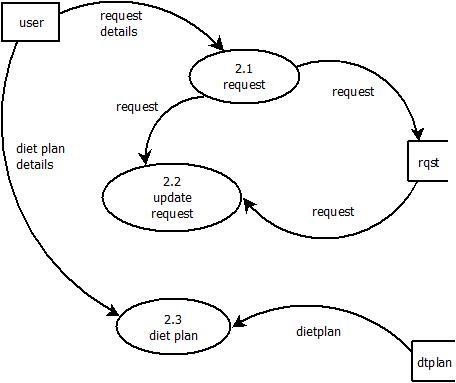


Fig 4.4.5 Level 1 Data Flow Diagram of Website - User

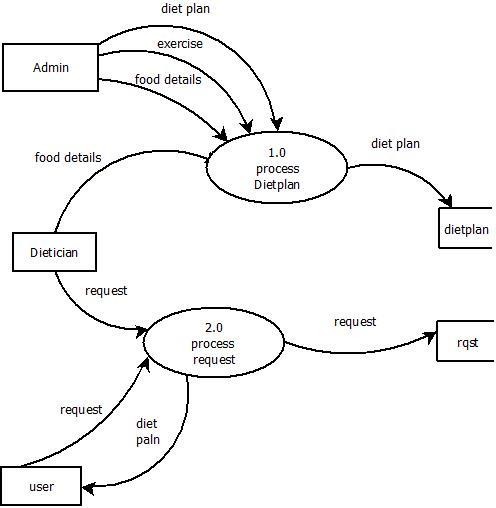


Fig 4.4.6 Level 2 Data Flow Diagram of Website - User

## SYSTEM DATA DESIGN

System data design is an essential aspect of system development that involves the creation of a structured and organized data storage and retrieval system. The database and configuration design are critical components of the data design process, as they provide the foundation for storing and managing the system's data. The database design includes the creation of tables, relationships, and constraints to ensure that data is stored accurately and efficiently. Configuration design, on the other hand, involves defining the various settings and parameters that govern the behavior of the system. Together, these design elements enable the system to operate effectively and efficiently, ensuring that data is stored, retrieved, and utilized in a manner that supports the system's overall functionality.

## Website

* + 1. Table Name: video

Description: This table is used for exercise details.

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constrain | Description |
| YID | Int | Primary key | Video id |
| YName | Varchar | Not null | video name |
| YTYPE | Varchar | Not null | video type |
| YDESC | Varchar | Not null | description |
| YPIC | varchar | Not null | Video |

* + 1. Table Name: Calorie

Description: This table is used for calorie details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| Calid | Int | Primary key | Calorie id |
| fname | Varchar | Not null | Food name |
| amount | Varchar | Not null | Amount |
| calories | Varchar | Not null | Calories |

1. Table Name:diet chart

Description: This table is used for diet details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | constraints | Descriptions |
| ID | int | Primary key | Calorie id |
| DTID | int | Foreign key | Dietician id |
| DAY | varchar | Not null | Day |
| PBF | varchar | Not null | Morning drink |
| BF | varchar | Not null | Breakfast |
| LU | varchar | Not null | Lunch |
| EV | varchar | Not null | Evening food |
| DN | varchar | Not null | Dinner |
| TNOT | varchar | Not null | total number of  days |
| des | varchar | Not null | description |

1. Table Name: Diet

Description: This table is used for diet details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| DTID | Int | Primary key | Dietician id |
| SDATE | date | Not null | Starting date |
| EDATE | date | Not null | Ending date |
| NOD | int | Not null | Number of days |
| RQID | int | Foreign key | Rgistration id |

Table Name:Exercise

Description: This table is used for exercise details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| EXID | Int | Primary key | Exercise id |
| ENAME | varchar | Not null | Exercise name |
| ETYPE | varchar | Not null | Exercise type |
| EDESC | varchar | Not null | xercise description |
| EVIDEO | varchar | Not null | Exercise video |

name:feedback

Description: This table is used for feedback details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| fid | Int | Primary key | feedback id |
| Fdk | varchar | Not null | Feedback |
| Dte | Date | Not null | Date |
| Uid | Int | Foreign key | User id |

* 1. Table Name:Login

Description: This table is used for login details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| uid | Int | Foreign key | Login id |
| uname | varchar | Not null | user name |
| upass | varchar | Not null | User password |
| utype | varchar | Not null | User type |

* 1. Table Name:Registration

Description: This table is used for registration details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| RID | Int | Primary key | Registration od |
| FIRST\_NAME | varchar | Not null | First name |
| LAST\_NAME | varchar | Not null | last name |
| ADDRESS | varchar | Not null | address |
| PHONE | varchar | Not null | Phone number |
| EMAIL | varchar | Not null | email |
| GENDER | varchar | Not null | Gender |
| DOB | varchar | Not null | Date of birth |
| PASSWORD | varchar | Not null | password |

* 1. Table name: Request diet plan details

Description: This table is used for request diet plan details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| rqid | Int | Primary key | Request id |
| cid | varchar | Foreign key | Dietician child id |
| wght | varchar | Not null | Weight |
| hght | varchar | Not null | Height |
| bmi | varchar | Not null | Bmi |
| res | varchar | Not null | request |
| status | varchar | Not null | Status |
| Uid | Int | Foreign key | User id |
| Uname | Varchar | Foreign key | User name |
| Dte | Date | Not null | Date |
| Did | Int | Foreign key | Dietician id |

* 1. Table Name: medical

Description: This table is used for medical details details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| RID | Int | Primary key | medical od |
| Rid | Int | Foreign key | Registration id |
| Name | varchar | Not null | Name |
| Age | Int | Not null | Age |
| Gnder | varchar | Not null | Gender |
| Height | varchar | Not null | Height |
| Weight | varchar | Not null | Weight |

* 1. Table Name: dietician

Description: This table is used for dietician details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| DID | Int | Primary key | Dietician id |
| Name | varchar | Not null | Name of dietician |
| Email | varchar | Not null | Email |
| Gender | varchar | Not null | Gender |
| PHONE | varchar | Not null | Phone number |
| EMAIL | varchar | Not null | email |

* 1. Table Name: nutrition

Description: This table is used for nutrition details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| RID | Int | Primary key | Nutrition id |
| Name | varchar | Not null | Name |
| Age | Int | Not null | Age |
| Details | varchar | Not null | Details |
| File | varchar | Not null | File |
| Email | varchar | Not null | email |
| Cno | Bigint | Not null | Contact number |
| Addr | varchar | Not null | Address |

* 1. Table Name: chat

Description: This table is used for chat details

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Constraints | Descriptions |
| CID | Int | Primary key | Chat |
| Name | varchar | Foreign key | Name |
| Uid | Int | Foreign key | User id |
| Nid | Int | Foreign key | Nutrition id |
| Nname | Varchar | Foreign key | Nutrition name |
| Date | Date | Not null | Date |
| Msg | varchar | Not null | Message |
| Rdte | Date | Not null | Reply date |
| Rmsg | varchar | Not null | Reply messages |

## USER INTERFACE DESIGN

**Website**

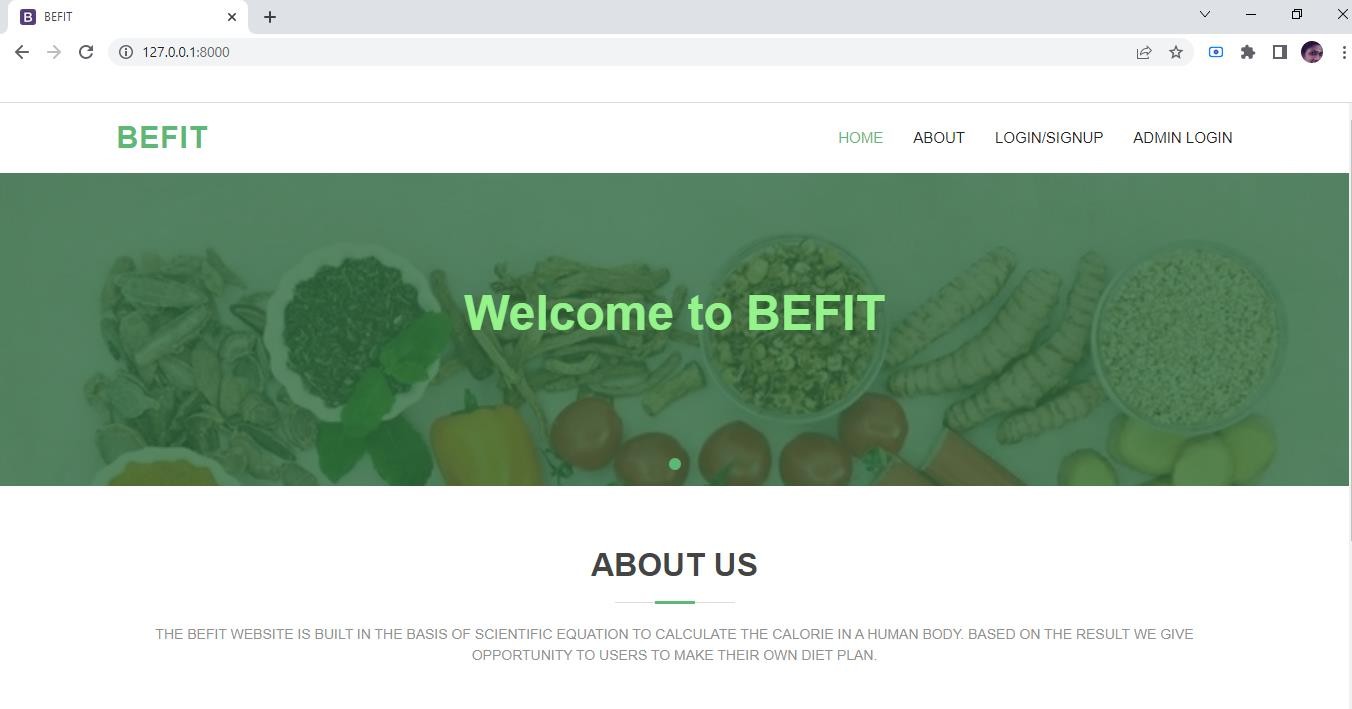


Fig 4.7.6 Screenshot of Website’s home page

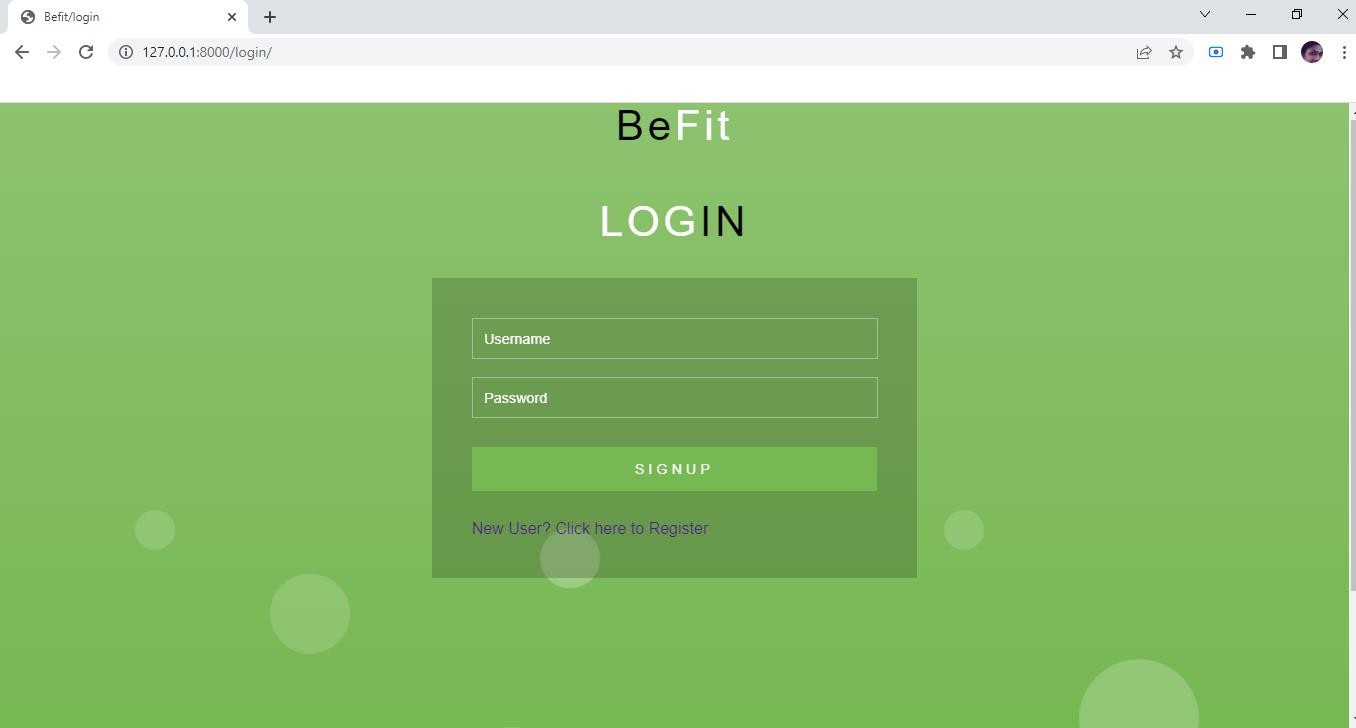


Fig 4.7.7 Screenshot of login page in website

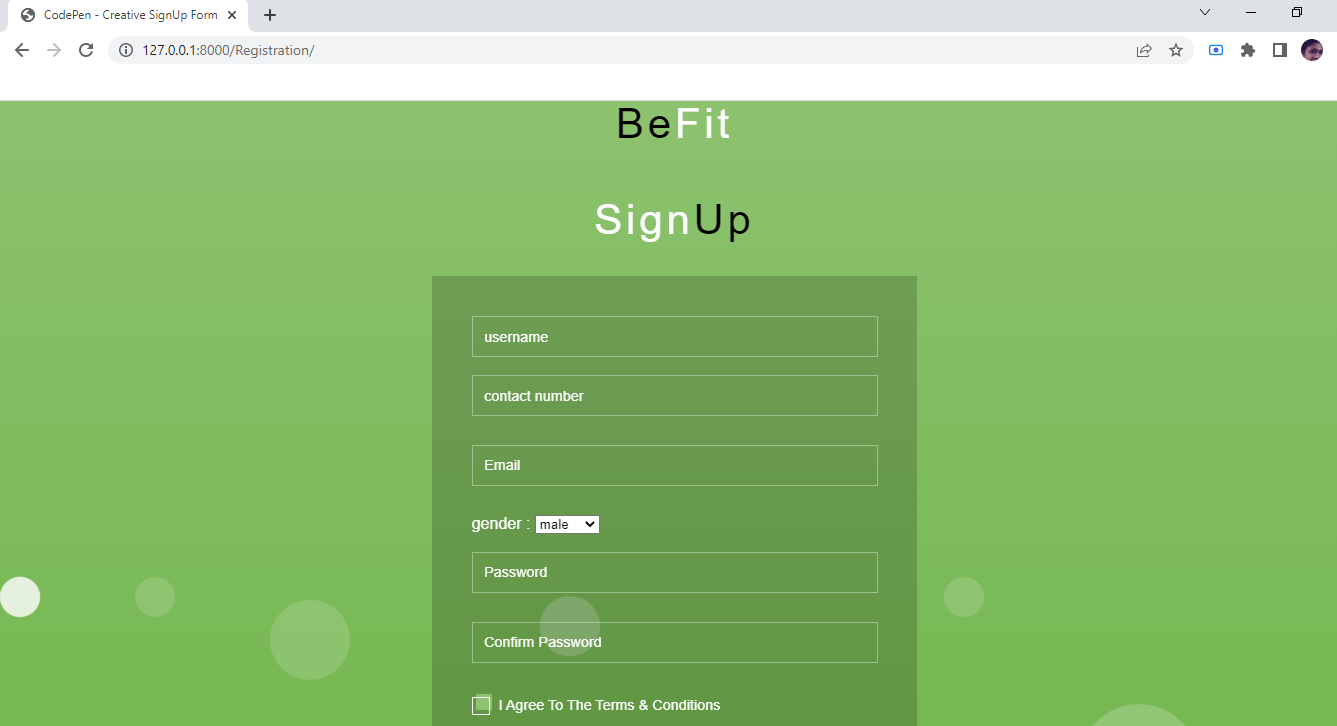


Fig 4.7.8 Screenshot of registration in website

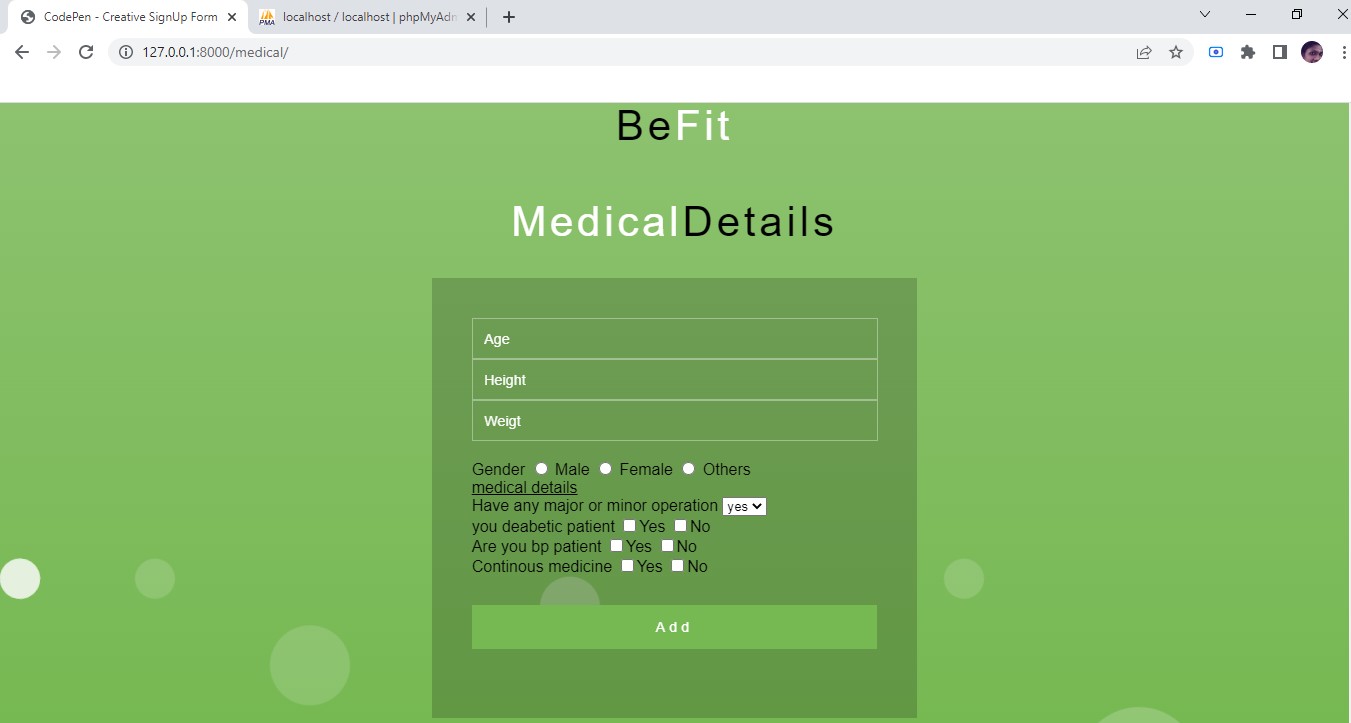


Fig 4.7.9 Screenshot of medical details in website

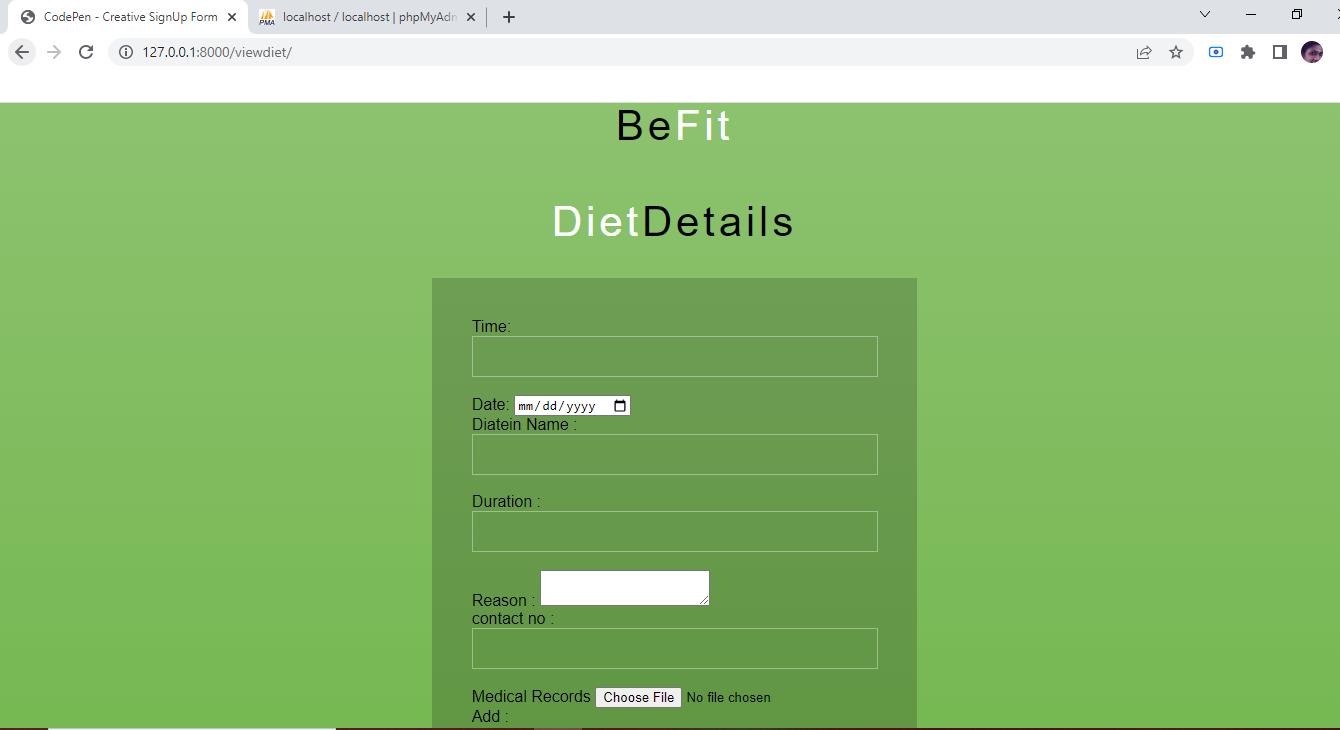


Fig 4.7.10 Screenshot of diet in website

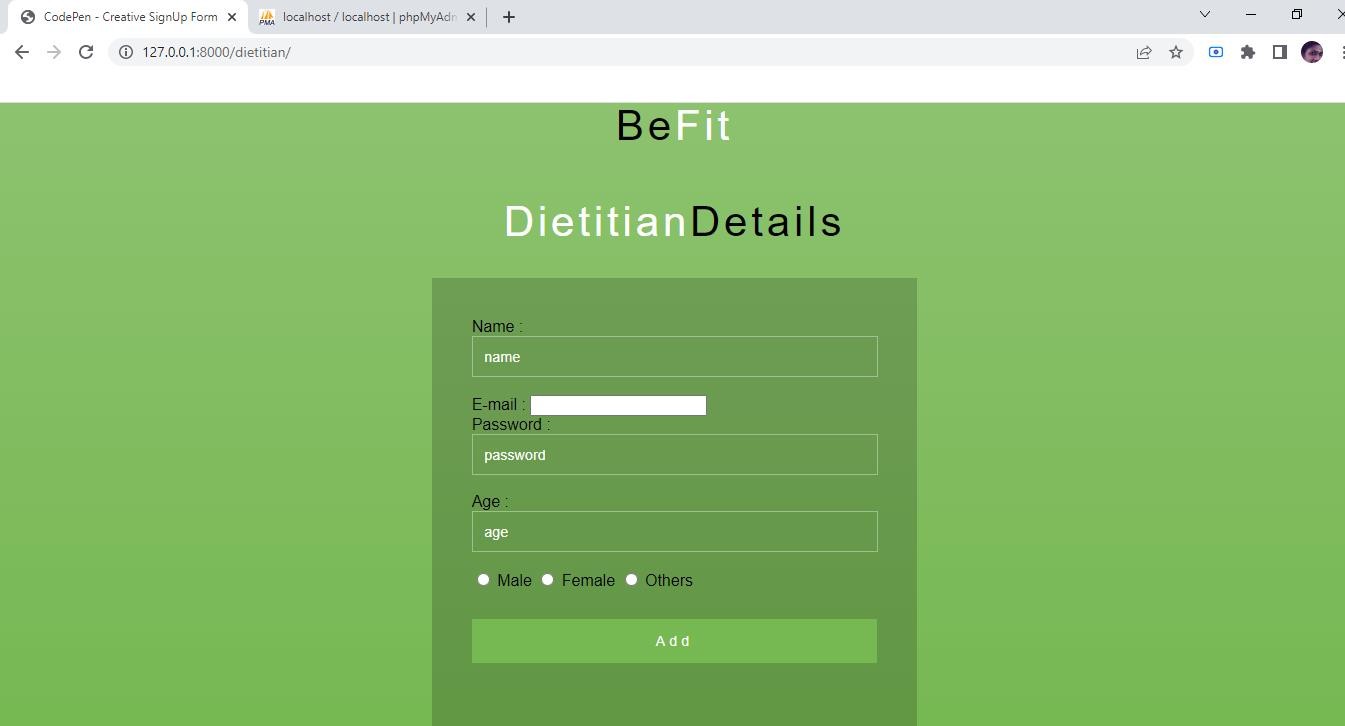


Fig 4.7.11 Screenshot of dietician form in website

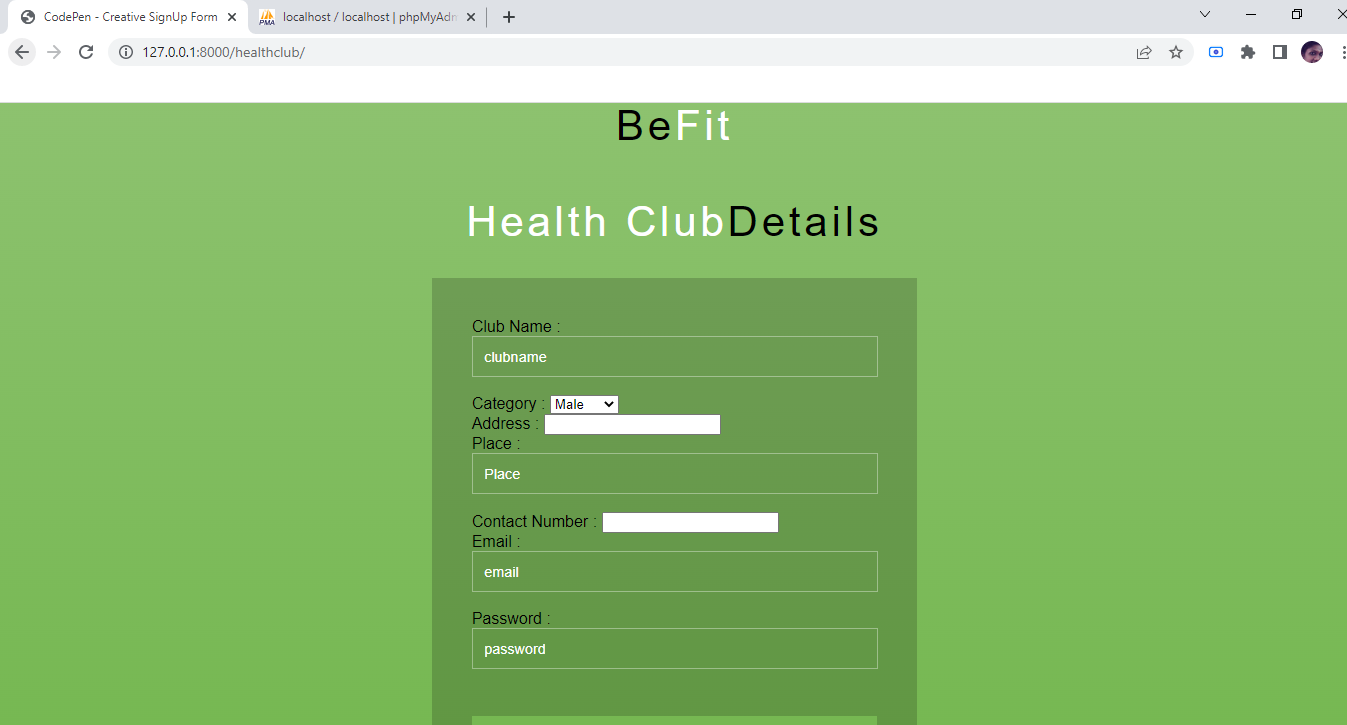


Fig 4.7.12 Screenshot of health club in Forum

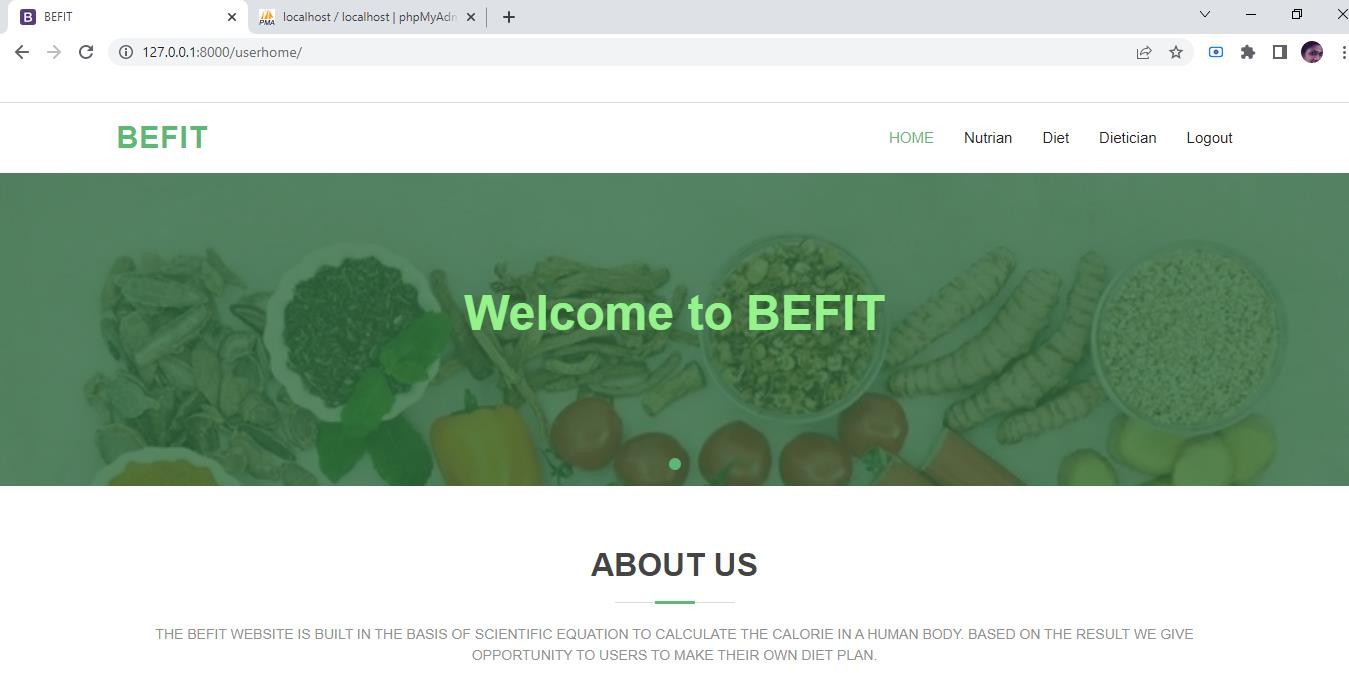


Fig 4.7.13 Screenshot of user home page

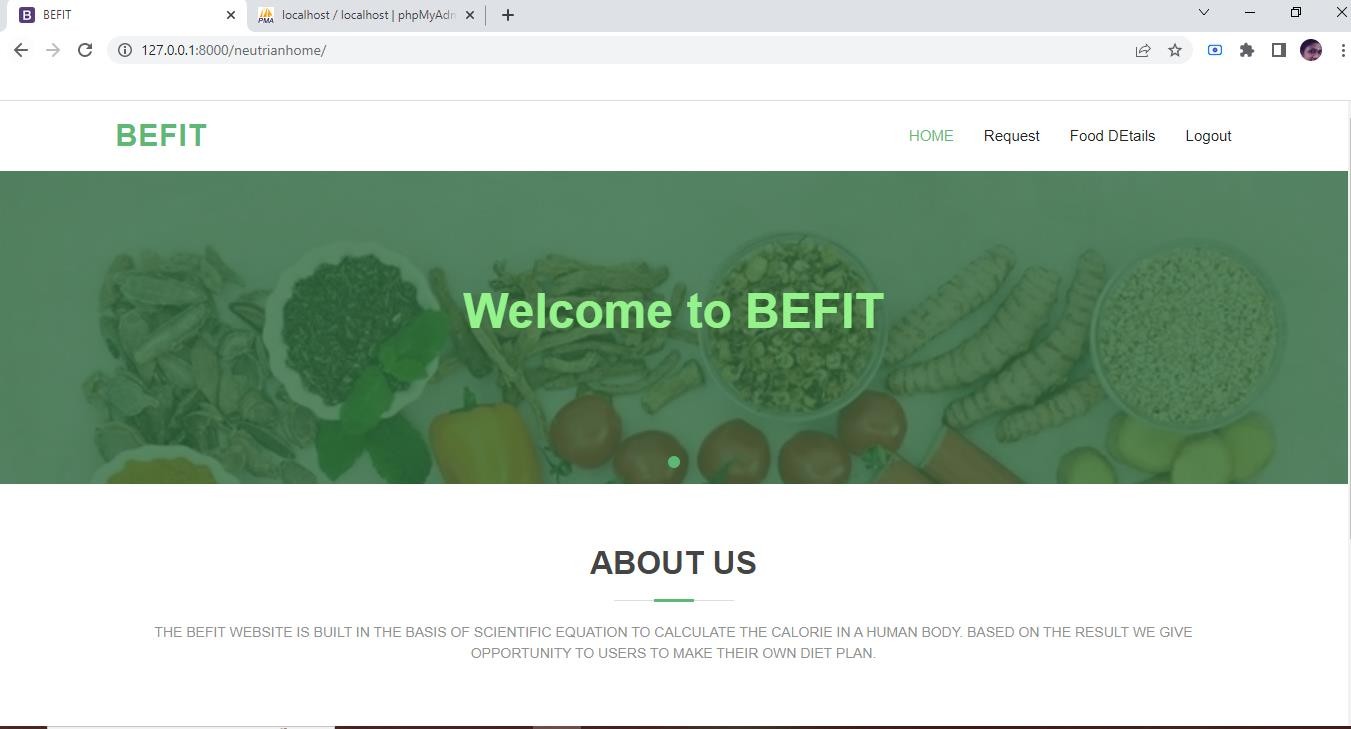


Fig 4.7.14 Screenshot of nutrition home page

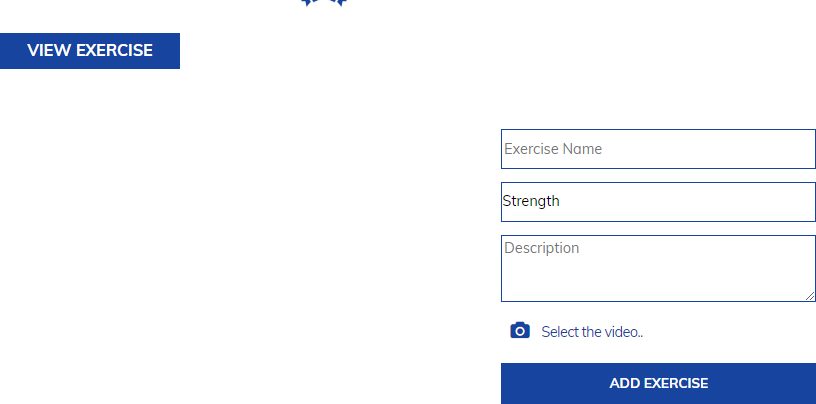


Fig 4.7.15 Screenshot of Exercise page



Fig 4.7.16 Screenshot of diet chart page



Fig 4.7.17 Screenshot of Diet plan request

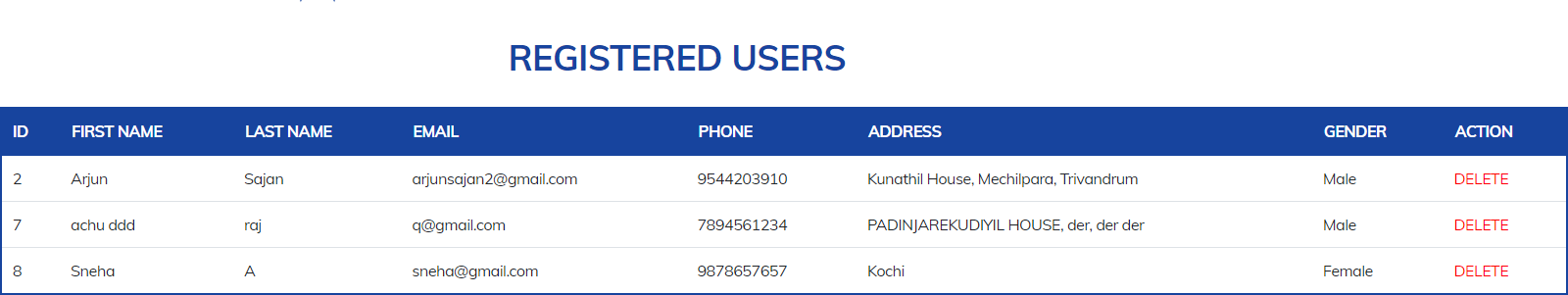


Fig 4.7.18 Screenshot of view users

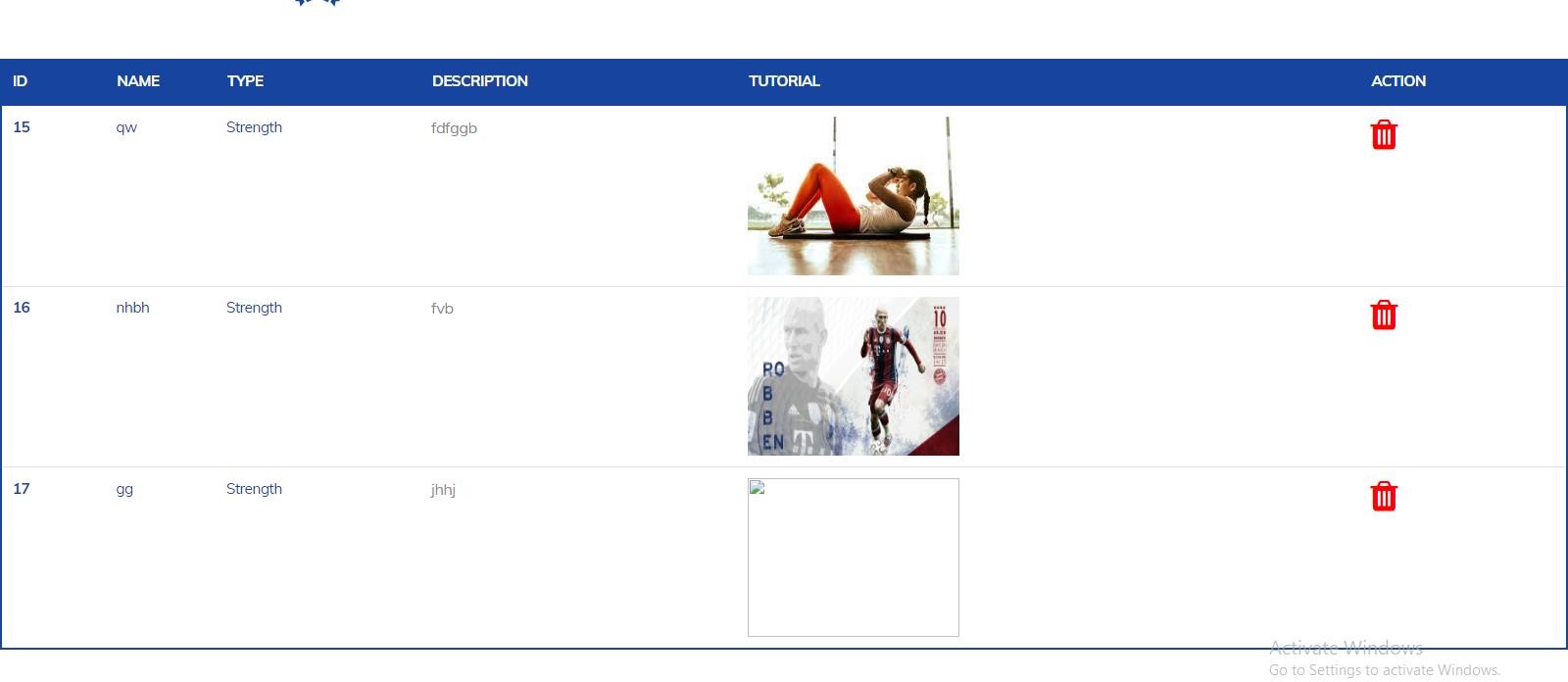


Fig 4.7.19 Screenshot of view exercise

# CODING

## INTRODUCTION

The coding phase in the software engineering paradigm is usually defined after the designing phase. In this phase, the developers or the coders have to implement the software design practically using any computer language(s) so that the software can be created and the user can use it. Now, coding is not just a matter of implementing the code in any suitable language as per the developer's choice. Some norms and standards are set for this purpose which is known as the coding standards.

In this project we have used GoLang as the programming language for the CLI application while HTML, PHP and JS have been used for the website. We Have adhered to the coding standards and ensured that good programming practices have been followed and there is a uniform appearance of the code.

## CODING SNIPPETS Code for view.py

from django.shortcuts import render from django.db import connection from django.http import HttpResponse

from django.shortcuts import HttpResponseRedirect,render\_to\_response,redirect # def index(request):

# return render(request,'index.html')

def login(request):

return render(request,'login.html')

def dietitian(request):

return render(request,'dietitian.html')

def healthclub(request):

return render(request,'healthclub.html')

def medical(request):

return render(request,'medical.html')

def nutrition(request):

return render(request,'nutrition.html')

def viewdieti(request):

return render(request,'viewdieti.html') def Registration(request):

return render(request,'Registration.html') def Registrationaction(request):

cursor=connection.cursor() un=request.GET['username'] c=request.GET['contact'] e=request.GET['email'] g=request.GET['gender'] p=request.GET['password'] cp=request.GET['conpass']

sql="insert into

registration(name,email,contactno,gender,password,conpass)values('%s','%s','%s','%s','%s','%s')"%(un,e

,c,g,p,cp)

cursor.execute(sql)

return render(request,'Registration.html') def medicalaction(request):

cursor=connection.cursor() a=request.GET['age'] h=request.GET['height'] w=request.GET['weight'] g=request.GET['gender'] op=request.GET['yes-no'] db=request.GET['deabetic'] bp=request.GET['bp'] mc=request.GET['medical']

sql="insert into

medical(age,height,weight,gender,operation,deabetic,bp,medicalcontinous)values('%s','%s','%s','%s','%s ','%s','%s','%s')"%(a,h,w,g,op,db,bp,mc)

cursor.execute(sql)

return render(request,'medical.html') def nutritionaction(request):

cursor=connection.cursor() n=request.GET['name'] a=request.GET['age'] d=request.GET['details'] f=request.GET['file'] e=request.GET['email'] c=request.GET['contact'] ad=request.GET['address'] p=request.GET['password']

sql="insert into

nutrition(name,age,details,file,email,contactno,address,password)values('%s','%s','%s','%s','%s','%s','%s'

,'%s')"%(n,a,d,f,e,c,ad,p) cursor.execute(sql)

return render(request,'nutrition.html') def dietitianaction(request):

cursor=connection.cursor() e=request.GET['email'] p=request.GET['password'] a=request.GET['age'] g=request.GET['gender'] n=request.GET['name']

sql="insert into

dietitian(email,password,age,gender,name)values('%s','%s','%s','%s','%s')"%(e,p,a,g,n) cursor.execute(sql)

return render(request,'dietitian.html') def healthclub(request):

return render(request,'healthclub.html') def healthclubaction(request):

cursor=connection.cursor() cn=request.GET['name'] ad=request.GET['address']

pl=request.GET['place'] ca=request.GET['category'] cno=request.GET['contact'] em=request.GET['email'] pas=request.GET['password']

sql="insert into

club(name,address,place,category,contactno,email,password)values('%s','%s','%s','%s','%s','%s','%s')"% (cn,ad,pl,ca,cno,em,pas)

cursor.execute(sql)

return render(request,'healthclub.html') def viewdiet(request):

return render(request,'viewdiet.html') def viewdietaction(request):

cursor=connection.cursor() t=request.GET['time'] d=request.GET['date'] dn=request.GET['name'] dur=request.GET['duration'] re=request.GET['reason'] c=request.GET['contact'] mr=request.GET['medical']

sql="insert into

viewdietitian(time,date,name,duration,reason,contact,medicalrecord)values('%s','%s','%s','%s','%s','%s','

%s')"%(t,d,dn,dur,re,c,mr) cursor.execute(sql)

return render(request,'viewdiet.html')

# SYSTEM TESTING

## INTRODUCTION

System testing is an important part of the software development life cycle that evaluates the system's compliance with its specified requirements. This testing process is conducted on a complete and integrated system to verify its functionality and performance. The objective of system testing is to detect any defects in the system, including any mismatches in the software and hardware components, and ensure that it meets the user's requirements. It also verifies that the system is ready for deployment and is capable of handling real-world scenarios. System testing can be performed manually or using automated testing tools and techniques to ensure the system is reliable, scalable, and secure.

## Unit Testing

Unit testing is a software testing method in which individual units or components of a software application are tested in isolation. The purpose of unit testing is to ensure that each individual unit is functioning as expected and that there are no defects or errors. This method involves writing test cases for each unit, running those test cases, and analyzing the results. Unit testing is typically automated, and it is performed by developers during the development process. By detecting and correcting defects early in the development cycle, unit testing helps to reduce the overall cost of software development and improve the quality of the final product.

## Integration Testing

Integration testing is a software testing method in which individual units or components of a software application are combined and tested as a group. The purpose of integration testing is to ensure that the different units or components of the software work together as expected and that there are no defects or errors introduced by the integration. This method involves testing the interactions between different units, verifying the data flow between them, and validating the overall behavior of the software application.

Integration testing can be performed manually or automated, and it is typically performed by developers or testers during the development cycle.

## Functional Testing

Functional testing is a software testing method in which the functionality of a software application is tested against the functional requirements or specifications. The purpose of functional testing is to ensure that the software application is performing as expected and that it meets the functional requirements of the end-users. This method involves testing the software application through various scenarios, verifying the input and output data, and analyzing the results. Functional testing can be performed manually or automated, and it is typically performed by testers during the testing phase of the software development life cycle.

## Regression Testing

Regression testing is a software testing method in which the software application is retested after a change or modification has been made to the software. The purpose of regression testing is to ensure that the change or modification has not introduced any new defects or errors into the software application. This method involves running a suite of previously executed test cases on the modified software application, verifying the results, and comparing them with the results from the previous version of the software. Regression testing can be performed manually or automated, and it is typically performed by testers during the testing phase of the software development life cycle.

## Performance Testing

Performance testing is a software testing method in which the performance of a software application is tested under different load conditions. The purpose of performance testing is to ensure that the software application can handle the expected load and that it meets the performance requirements of the end-users. This method involves simulating real-world scenarios, measuring the response time of the software application, and analyzing the results. Performance testing can be performed manually or automated, and it is typically performed by testers during the testing phase of the software development life cycle.

## Acceptance Testing

Acceptance testing is a software testing method in which the software application is tested against the acceptance criteria or requirements specified by the end-users. The purpose of acceptance testing is to ensure that the software application meets the expectations of the end-users and that it is ready for release. This method involves testing the software application through various scenarios, verifying the input and output data, and analyzing the results. Acceptance testing can be performed manually or automated, and it is typically performed by the end-users or a representative during the acceptance phase of the software development life cycle.

## WEBSITE

The following methods of testing were carried out to ensure reliability of the website

## Functional testing

Functional testing is a type of software testing that verifies the behavior of a software application against the functional requirements or specifications. The purpose of functional testing is to ensure that the software application performs all the functions as per the requirement and there are no defects or errors in the functionalities. It involves identifying the functional requirements, creating test cases, executing them, and comparing the actual results with the expected results. Functional testing can be done manually or automated, and it is usually performed by testers during the testing phase of the software development life cycle.

## Performance testing

Performance testing is a type of software testing that checks how well the software application performs under different load conditions. The purpose of performance testing is to identify the performance issues and bottlenecks in the software application, and to ensure that the application meets the performance requirements of the end-users. It involves simulating the real-world scenarios, measuring the response time of the software application, and analyzing the results. Performance testing can be done manually or

automated, and it is usually performed by testers during the testing phase of the software development life cycle.

## Security testing

Security testing is a type of software testing that checks the security of the software application. The purpose of security testing is to identify the vulnerabilities and loopholes in the software application, and to ensure that the application is secure and free from any potential threats. It involves identifying the security requirements, creating test cases, executing them, and analyzing the results. Security testing can be done manually or automated, and it is usually performed by testers during the testing phase of the software development life cycle.

## Usability testing

Usability testing is a type of software testing that evaluates the ease of use of the software application. The purpose of usability testing is to identify the issues and difficulties faced by the end-users while using the software application, and to ensure that the application is user-friendly and easy to use. It involves identifying the usability requirements, creating test cases, executing them, and analyzing the results. Usability testing can be done manually or automated, and it is usually performed by testers during the testing phase of the software development life cycle.

## Compatibility testing

Compatibility testing is a type of software testing that checks the compatibility of the software application with different hardware, software, and operating systems. The purpose of compatibility testing is to ensure that the software application works smoothly on different platforms, and to identify any compatibility issues. It involves identifying the compatibility requirements, creating test cases, executing them on different platforms, and analyzing the results. Compatibility testing can be done manually or automated, and it is usually performed by testers during the testing phase of the software development life cycle.

## Regression testing

Regression testing is a type of software testing that verifies that the changes made in the software application have not affected the existing functionalities. The purpose of regression testing is to ensure that the changes made in the software application have not introduced any new defects or errors, and to ensure that the software application is still working as expected. It involves identifying the changes made, creating test cases, executing them, and analyzing the results. Regression testing can be done manually or automated, and it is usually performed by testers during the testing phase of the software development life cycle.

## User acceptance testing

User acceptance testing is a type of software testing that checks whether the software application meets the expectations of the end-users. The purpose of user acceptance testing is to ensure that the software application is ready for release and that it meets the requirements of the end-users. It involves creating test cases based on the user requirements, executing them, and analyzing the results. User acceptance testing is usually performed by the end-users or the representatives of the end-users during the acceptance phase of the software development life cycle.

# SYSTEM IMPLEMENTATION

## INTRODUCTION

System implementation is the process of putting a software system into action or making it operational. It involves a series of activities that are necessary for deploying the system and making it ready for use by the end-users. This includes the installation and configuration of hardware and software components, database setup, and integration with other systems. The implementation process also involves testing the system to ensure it meets the requirements and specifications outlined during the design and development phases. The goal of system implementation is to ensure a smooth and seamless transition from the old system to the new one, minimizing disruption to business operations and maximizing the benefits of the new system.

## WEBSITE

Implementation is an activity that is contained throughout the development phase. It is the process of bringing a developed system into operational use and turning it over to the user. The new system and its components are to be tested in a structured and planned manner. A successful system should be delivered and users should have the confidence that the system would work efficiently and effectively. The more complex the system being implemented the more involved will be the system analysis and design effort required for implementation. Implementation is the stage of the system when the theoretical design is turned into working system. The implementation involves careful planning investigation of the current system and its constraints on implementing, design of methods to achieve the changeover, training of user over procedure and evaluation change over method. There are three types of implementation:

Maintenance corresponds to restoring something to original conditions, covering a wide range of activities including correcting codes and design errors and updating user support. Maintenance is performed most often to improve the existing software rather than to a crisis or risk failure. The system would fail if not properly maintained. The software maintenance is an important one in the software development because we have to spend more efforts for maintenance. Software maintenance is to improve the software quality according to the requirements. After a system is successfully implemented, it should be maintained in a proper manner. The need for system maintenance is to make the system adaptable to the changes in the system environment. There may be social, economical or technical changes, which affect system being implemented.

# CONCLUSION

All the requirements and specifications was followed as for as possible and few additional features were added that can make the application more user friendly and less complicated. The project was successfully completed within the time span allotted. All the modules are tested separately and put together to form the main system. Finally the system is tested with real data and it worked successfully. Thus the system has fulfilled the entire objective defined.

# FUTURE ENHANCEMENT

Any system which has been in use for a number of years gradually decays and become less effective because of change in environment to which it has to be adapted. For the time being it is possible to overcome problems by amendments and minor modifications to acknowledge the need of fundamental changes.

Nothing is perfect; in future we would like our prototype to be more perfect by including future improvements. The proposed system is very easy to handle so everyone can access.

* + - The entire process of the firm can be computerized.
    - More number of item details can be inserted into the database.
    - It can be integrated with the web for universal access.
    - Upgrading the performance .This system is now implemented at the client machine only but as a future enhancement we can modify the system in such a way to make it work on a client- server network. The system can be even more enhanced by making it an internet based system.
    - The efficiency of the developed system can be improved by modification. Today most of the services are based on web services. This system can also be connected online.

# BIBLIOGRAPHY

1. Max Kanat-Alexander, K. (2012). “Code Simplicity: The Fundamentals of Software” [PDF]. O’Reilly. Available at:

[www.codesimplicity.com/wp-content/uploads/2022/05/CodeSimplicity.pdf](http://www.codesimplicity.com/wp-content/uploads/2022/05/CodeSimplicity.pdf)

1. Bootstrap documentation [online]. Available at: getbootstrap.com/docs
2. W3Schools [online]. Available at: [www.w3schools.com](http://www.w3schools.com/)
3. [www.slideshare.com](http://www.slideshare.com/)
4. www.project management basics.com
5. www.SDLC tutorial.com
6. MySql:https:/[/www.w3schools.com/java/j](http://www.w3schools.com/java/java_mysql_insert.asp)a[va\_mysql\_insert.asp](http://www.w3schools.com/java/java_mysql_insert.asp)