A PROJECT REPORT

On

HEALTHY DIET: FOOD NUTRITION

Submitted in partial fulfillment of the requirement of
University of Mumbai for

Internet Programming Mini Project
In
Information Technology

Submitted By
Harsh Patil
Piyush Pandey
Rudein Salim
Raj Kakodkar

Department Of Information Technology
PILLAI COLLEGE OF ENGINEERING
New Panvel – 410 206
UNIVERSITY OF MUMBAI

Academic Year 2023 – 24

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

CERTIFICATE

This is to certify that the requirements for the report entitled '**Healthy Diet: Food Nutrition**' have been successfully completed by the following students:

Name	**Roll No.**
Harsh Patil	B546
Piyush Pandey	B543
Rudein Salim	B528
Raj Kakodkar	B526

In partial fulfillment of Internet Programming mini Project in the Department of Information Technology, Pillai College of Engineering, New Panvel – 410 206 during the Academic Year 2023 – 2024.

Mrs. Prerna Kulkarni

Subject Incharge

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

PROJECT APPROVAL FOR

This project entitled “Healthy Diet” by Harsh Patil, Piyush Pandey, Rudein Salim, Raj Kakodkar is approved for the degree of Bachelor of Engineering in Information Technology.

Examiners:

1. _____

2. _____

3. _____

4. _____

5. _____

Date

DEPARTMENT OF INFORMATION TECHNOLOGY
Pillai College of Engineering
New Panvel – 410 206

DECLARATION

We declare that this written submission for the Internet Programming Mini Project entitled “Healthy Diet” represents our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

Project Group Members:

Harsh Patil

_____

Piyush Pandey

_____

Rudein Salim

_____

Raj Kakodkar

_____

Abstract

In today's age of heightened health consciousness and a growing emphasis on personal well-being, the "Food Nutrition system" stands as a beacon of innovation and user-centricity. This dynamic web project has emerged to address the pressing need for promoting healthy eating habits and nutrition awareness in our society. It serves as a multifaceted resource, catering to individuals from all walks of life who are eager to embark on a journey towards a healthier lifestyle. The "Food Nutrition system" is designed to provide a comprehensive, easily accessible hub of information and guidance. It offers a wide array of tools, articles, and interactive features that empower users to make informed choices about their dietary habits. With user-friendliness at its core, this platform ensures that anyone, regardless of their background or previous knowledge, can access the information they need to make healthier food choices. Furthermore, the "Food Nutrition system" fosters a sense of community and support, allowing users to connect with like-minded individuals on similar journeys. Through forums, discussion boards, and social features, it encourages the exchange of experiences, tips, and success stories, creating a positive and motivating environment. In summary, the "Food Nutrition system" is not just a website; it's a dynamic movement towards better health and well-being. It recognizes the importance of making healthy choices accessible, understandable, and enjoyable, ultimately empowering individuals to take control of their nutrition and embark on a path to improved quality of life.

Table of Contents

Abstract		i
Table of content		ii
Table of figures		iii
1.	**Introduction**	1
2.	**Requirement Analysis**	2
3.	**Wireframe**	3
4	**Using different types of CSS**	5
5	**HTML5 based form validation**	9
6	**Javascript based form validation**	12
7	**MySQL database operations**	15
8	**Local hosting**	19
	Acknowledgement	

Table of Figures

3.1	**Screenshot of Landing Page**	3
3.2	**Screenshot of BMI Calculator**	3
3.3	**Screenshot of Calorie Counter**	4
3.4	**Screenshot of About us Page**	4
4.1	**Screenshot of Landing Page**	8
5.1	**Screenshot of BMI Calculator**	12
6.1	**Screenshot of Calorie Counter**	14
7.1	**Screenshot of MySQL**	17
7.2	**Screenshot of Database Table**	17
7.3	**Screenshot of Database**	18
8.1	**Snapshot of Landing page**	20
8.2	**Snapshot of BMI Page**	20
8.3	**Snapshot of Calories Calculator Page**	21
8.4	**Snapshot of About us page**	21

Chapter 1
Introduction

In an age where health has become a paramount concern for individuals around the world, the significance of a healthy diet and proper nutrition has never been more pronounced. The project at hand, "Healthy Diet: Food Nutrition," emerges as a pivotal initiative aimed at addressing this imperative need. In an era characterized by busy lifestyles, readily available fast food, and sedentary habits, the project recognizes the pressing importance of promoting a balanced and nutritious diet. It seeks to empower individuals with the knowledge and tools they need to make informed choices about their food consumption, ultimately leading to improved overall health and well-being. This project serves as a comprehensive guide and resource hub for anyone looking to embark on a journey toward a healthier lifestyle through better dietary habits. It not only provides valuable information about the nutritional value of various foods but also offers practical tips, meal plans, and recipes to facilitate the transition to a more health-conscious diet. Furthermore, "Healthy Diet: Food Nutrition" endeavors to foster a sense of community by creating a platform where individuals can connect, share experiences, and support each other in their pursuit of better health. This project represents a holistic approach to wellness, acknowledging that a healthy diet is not just a means to a physical end but a key component of a fulfilling and vibrant life.

Chapter 2

Requirement Analysis

Functional requirements

1.Nutrition Database:

- Include an extensive database using mySQL for food items with detailed nutritional information (calories, macronutrients, vitamins, minerals, etc.).
- Users should be able to search and browse this database.

2.User-Friendly Design:

- Ensure that the web project is with the user friendly navigation bar.

3.Operation Landing Page:

- It includes an attractive landing page made by using CSS and JAVASCRIPT effects where users can visit every aspect of the project.

4.Attractive BMI and Calorie counting calculators:

- Using CSS and HTML two calculators are created which give precise values for each person and input.

Non-functional requirements

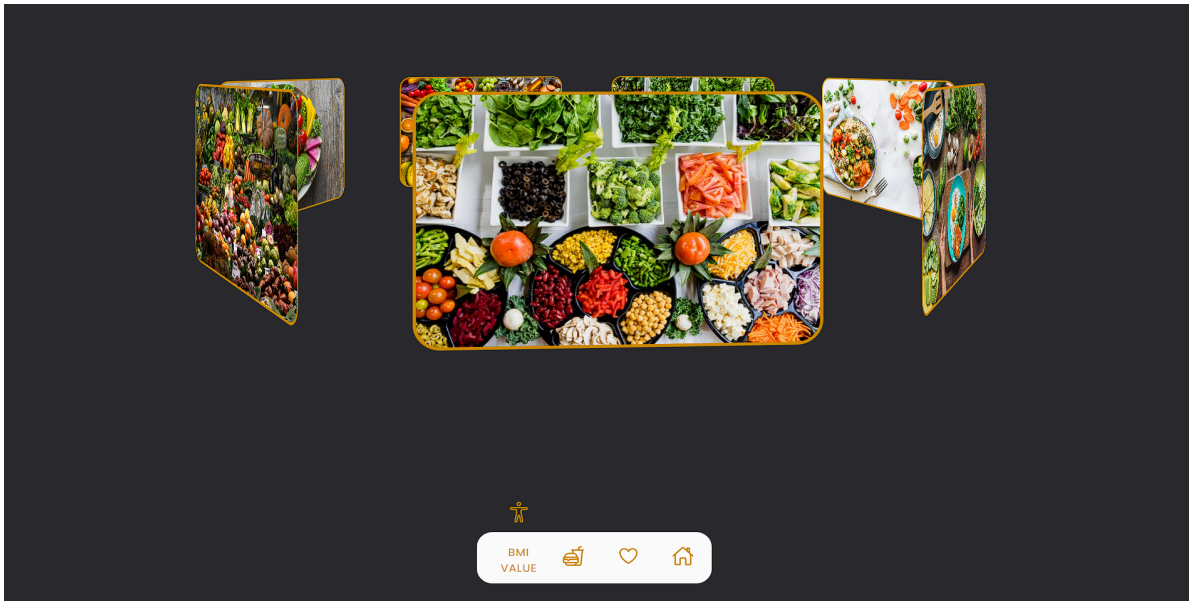
UI Preference:

For easy and friendly user interaction we have used an user interface created by using HTML, CSS and JAVASCRIPT.

Chapter 3

Wireframe

Wireframe 1

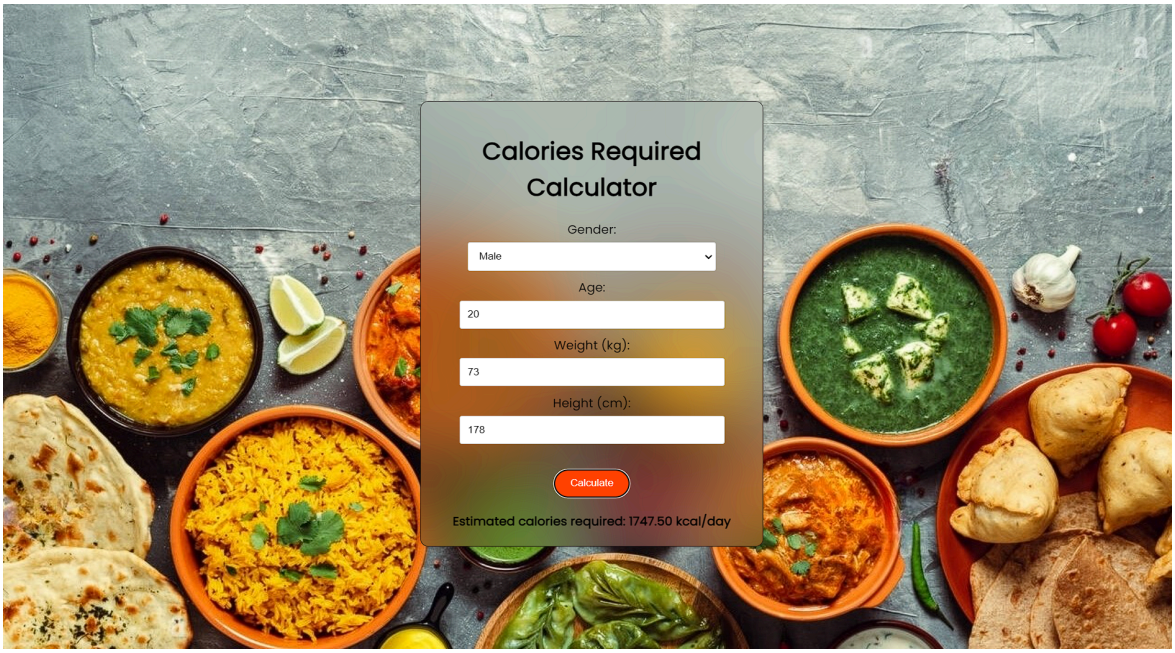


(fig 3.1 Screenshot of Landing Page)

Wireframe 2

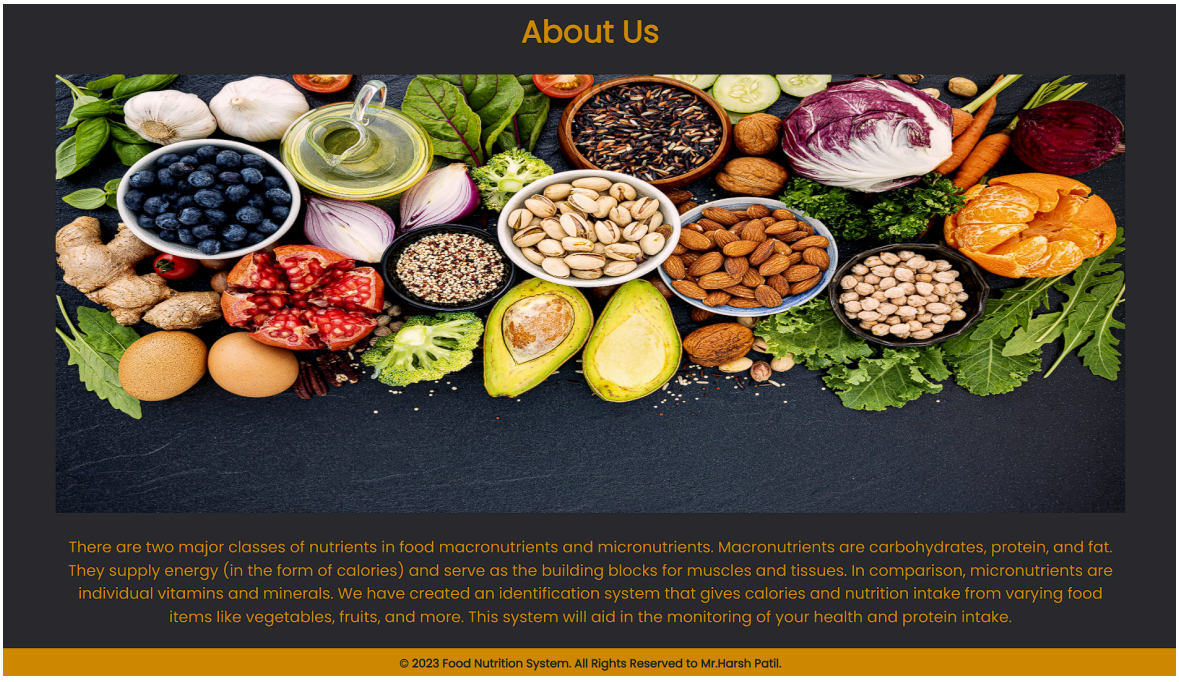
(fig 3.2 Screenshot of BMI Calculator)

Wireframe 3



(fig 3.3 Screenshot of Calorie Counter)

Wireframe 4



(fig 3.4 Screenshot of About us Page)

Chapter 4

Using different types of CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents. CSS is a language that describes the style of an HTML document. CSS describes how HTML elements should be displayed.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C).

Syntax of CSS

A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts :

Selector − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

Property - A property is a type of attribute of an HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.

Value - Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.

selector { property: value } selector { property: value, property: value }
h1 {‘color’:’blue’}

Types of css selectors:

- The element Selector :h1 {color: red;}
- The id Selector / type selector :#para1 {text-align: center;color: red;}
- The Descendant Selectors : ul li {color: red;}
- The class Selector : .center {text-align: center;color: red;}
- The Attribute Selectors : input[type = "text"]{color: #000000; }
- The Child Selectors : body > p {color: #000000; }
- The Universal Selectors : * {color: red;}
- The Adjacent Sibling Selector : H2+P {color: red;}

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style

External style sheet

The <link> element can be used to include an external stylesheet file in your HTML document. An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Here is the generic syntax of including external CSS file −

```
<head>
  <link type = "text/css" href = "..." media = "..." />
</head>
```

Consider a simple style sheet file with a name mystyle.css having the following rules −

```
h1, h2, h3 {
   color: #36C;
   font-weight: normal;
   letter-spacing: .4em;
   margin-bottom: 1em;
   text-transform: lowercase;
}
```

Now you can include this file mystyle.css in any HTML document as follows −

```
<head>
   <link type = "text/css" href = "mystyle.css" media = " all" />
</head>
```

Internal style sheet

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document

<style> Attribute type =”text/css” <style type=”text/css” > </style>

Specifies the style sheet language as a content-type (MIME type). This is required attribute.

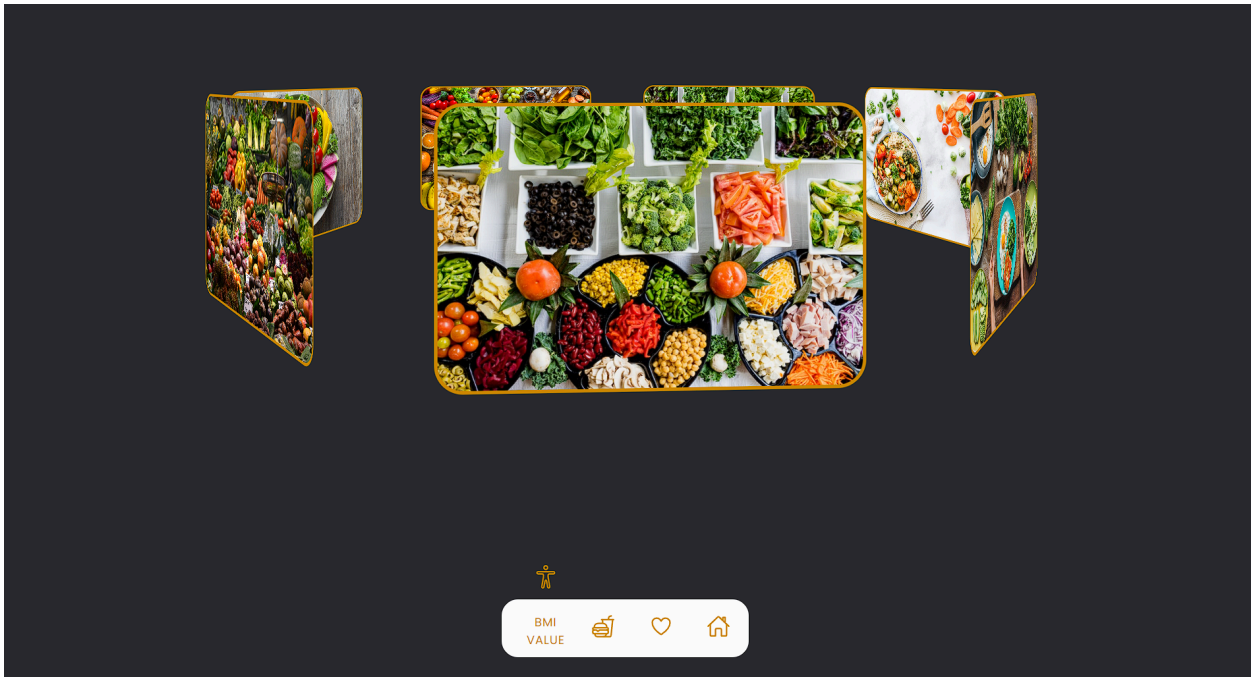
media attribute <style type = "text/css" media = "all">

```
<head>
<style type=”text/css” >
body {
        background-color: linen;
}
h1 {
        color: maroon;
```

```
        margin-left: 40px;
}
</style>
</head>
```

Code for rotating images on landing page:

```
.carousel {
  position: relative;
  width: 100%;
  height: 100%;
  transform-style: preserve-3d;
  animation: rotate360 60s infinite forwards linear;
}
```



(fig 4.1 Screenshot of Landing Page)

Chapter 5

HTML5 based form validation

Forms are used in web pages for the user to enter their required details that are further send it to the server for processing. A form is also known as web form or HTML form. Form validation helps us to ensure that users fill out forms in the correct format, making sure that submitted data will work successfully with our applications.

Go to any popular site with a registration form, and you will notice that they give you feedback when you don't enter your data in the format they are expecting. You'll get messages such as:

"This field is required" (you can't leave this field blank)

"Please enter your phone number in the format xxx-xxxx" (it enforces three numbers followed by a dash, followed by four numbers)

"Please enter a valid email address" (if your entry is not in the format "somebody@example.com")

"Your password needs to be between 8 and 30 characters long, and contain one uppercase letter, one symbol, and a number"

This is called form validation — when you enter data, the web application checks it to see that the data is correct. If correct, the application allows the data to be submitted to the server and (usually) saved in a database; if not, it gives you an error message explaining what corrections need to be made. Form validation can be implemented in a number of different ways.

We want to make filling out web forms as easy as possible. So why do we insist on validating our forms? There are three main reasons:

We want to get the right data, in the right format — our applications won't work properly if our user's data is stored in the incorrect format, or if they don't enter the correct information, or omit information altogether.

We want to protect our users' accounts — by forcing our users to enter secure passwords, it makes it easier to protect their account information.

We want to protect ourselves — there are many ways that malicious users can misuse unprotected forms to damage the application they are part of (see Website security).

Different types of form validation

There are two different types of form validation which you'll encounter on the web:

Client-side validation is validation that occurs in the browser before the data has been submitted to the server. This is more user-friendly than server-side validation as it gives an instant response. This can be further subdivided:

JavaScript validation is coded using JavaScript. It is completely customizable.

Built-in form validation using HTML5 form validation features. This generally does not require JavaScript. Built-in form validation has better performance, but it is not as customizable as JavaScript.

Server-side validation is validation which occurs on the server after the data has been submitted. Server-side code is used to validate the data before it is saved into the database. If the data fails authentication, a response is sent back to the client to tell the user what corrections to make. Server-side validation is not as user-friendly as client-side validation, as it does not provide errors until the entire form has been submitted. However, server-side validation is your application's last line of defense against incorrect or even malicious data. All popular server-side frameworks have features for validating and sanitizing data (making it safe).

1. Specialized Input Types

HTML5 introduced several new input types. They can be used to create input boxes, which will accept only a specified kind of data.

The new input types are as follows:

Color, date, datetime, email , month .number, range . search, tel , time , url , week To use one of the new types, include them as the value of the type attribute: <input type="email"/>

2. Required Fields

By simply adding the "required" attribute to a <input>, <select> or <textarea>, you tell the browser that a value must be provided in this field. Think of this as the red asterisk* we see in most registration forms.

<input type="checkbox" name="terms" required >

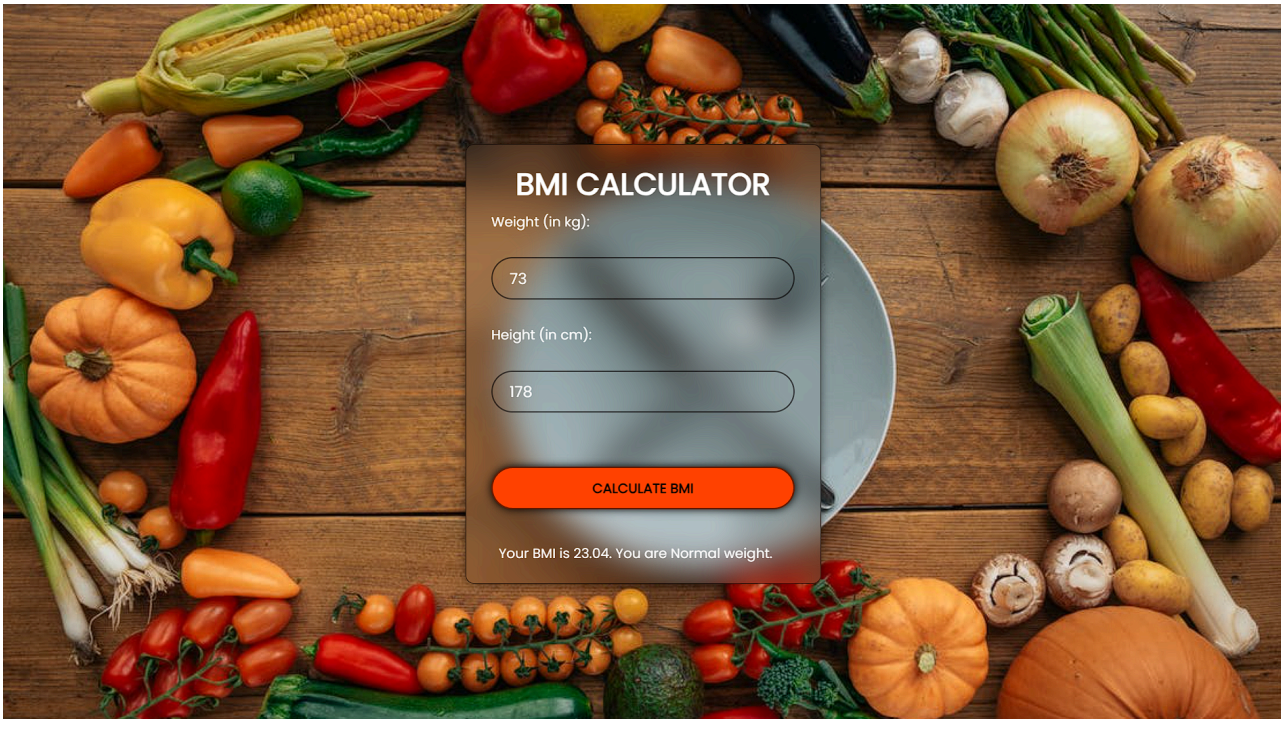
3. Limits

We can set some basic limitations like max length and minimum and maximum values for number fields. To limit the length of input fields and textareas, use the "maxlength" attribute. What this does is to forbid any string longer than the field's "maxlength" value to be entered at all. If you try to paste a string which exceeds this limit, the form will simply clip it.

<input type="text" name="name" required maxlength="15">

The <input type="number"> fields use "max" and "min" attributes to create a range of possible values - in our example we've made the minimum allowed age to be 18 (too bad you can be whatever age you want on the internet).

<input type="number" name="age" min="18" required>



(fig 5.1 Screenshot of BMI Calculator)

Chapter 6

Javascript based form validation

Validating form input with JavaScript is easy to do and can save a lot of unnecessary calls to the server as all processing is handled by the web browser. It can prevent people from leaving fields blank, from entering too little or too much or from using invalid characters.

Forms validation on the client-side is essential — it saves time and bandwidth, and gives you more options to point out to the user where they've gone wrong in filling out the form. Having said that, I don't mean that you don't need server-side validation. People who visit your site may use an old browser or have JavaScript disabled, which will break client-only validation. Client and server-side validation complement each other, and as such, they really shouldn't be used independently.

Why is Client Side Validation Good?

There are two good reasons to use client-side validation:

1. It's a fast form of validation: if something's wrong, the alarm is triggered upon submission of the form.
2. You can safely display only one error at a time and focus on the wrong field, to help ensure that the user correctly fills in all the details you need.

Two Major Validation Approaches

1. Display the errors one by one, focusing on the offending field
2. Display all errors simultaneously, server-side validation style

While displaying all errors simultaneously is required for server-side validation, the better method for validation on the client-side is to show one error at a time. This makes it possible to highlight only the field that has been incorrectly completed, which in turn makes revising and successfully submitting the form much easier for the visitor. If you present users with all errors at the same time, most people will try to remember and correct them at once, instead of attempting to re-submit after each correction.

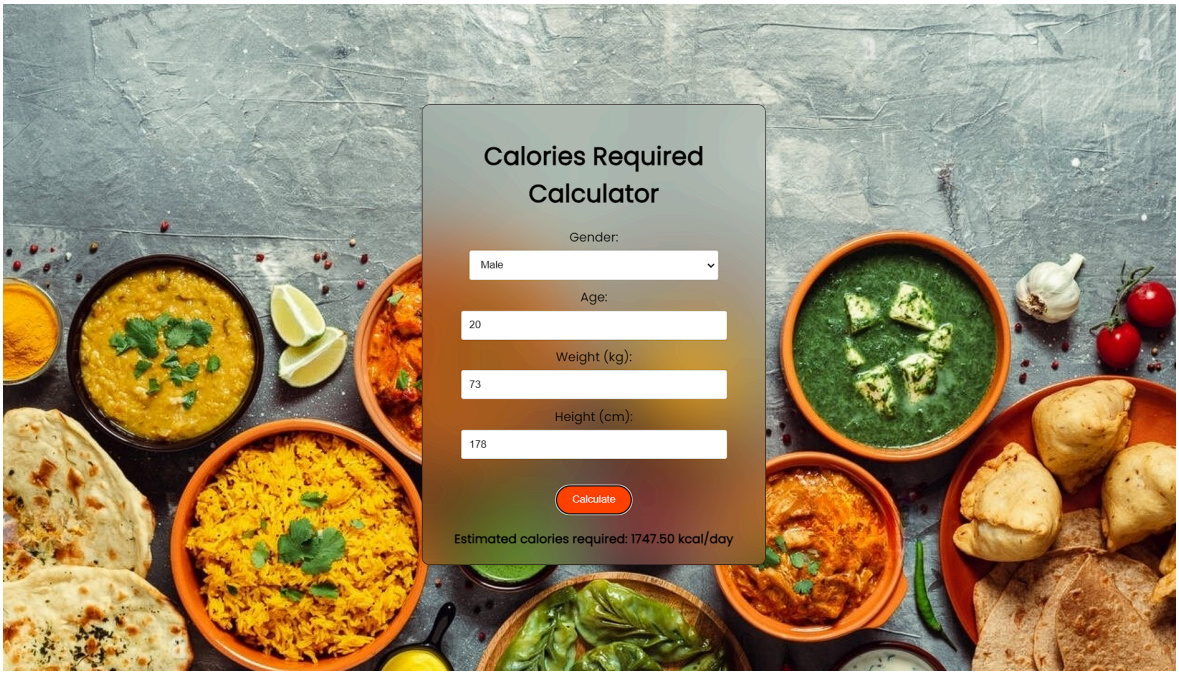
```
function validateForm() {
  var x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
```

```
        return false;
    }
}
<form name="myForm" action="/action_page.php" onsubmit="return validateForm()" method="post">
Name: <input type="text" name="fname">
<input type="submit" value="Submit">
</form>
function checkpassword(pform1){
var str=pform1.password.value;

//check required fields
//password should be minimum 4 chars but not greater than 8
if ((str.length < 4) || (str.length > 8)) {
function checkpassword(pform1){
var str=pform1.password.value;

//check required fields
//password should be minimum 4 chars but not greater than 8
if ((str.length < 4) || (str.length > 8)) {
alert("Invalid password length.")
pform1.password.focus()
return false
}
}
function checkemailphone(pform1){
var email = pform1.email.value;
var phone = pform1.phone.value;
var cleanstr = phone.replace(/[().- ]/g, '');
var validemail =/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,4}$/;
```

```
if(!(validemail.test(email))){
alert("Invalid email address")
pform1.email.focus()
return false
}
//check phone number
if (isNaN(parseInt(cleanstr))) {
alert("The phone number contains unwanted characters.") }
}
```



(fig 6.1 Screenshot of Calorie Counter)

Chapter 7
MySQL database operations

With PHP, you can connect to and manipulate databases. MySQL is the most popular database system used with PHP.

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful for storing information categorically. A company may have a database with the following tables:

- Employees
- Products
- Customers
- Orders

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = mysqli_connect($servername, $username, $password);
```

```
// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

If you want to use PHP to query your MySQL database you can do that by either entering the MySQL query command in the PHP script or define the command as a variable and use the variable when needed

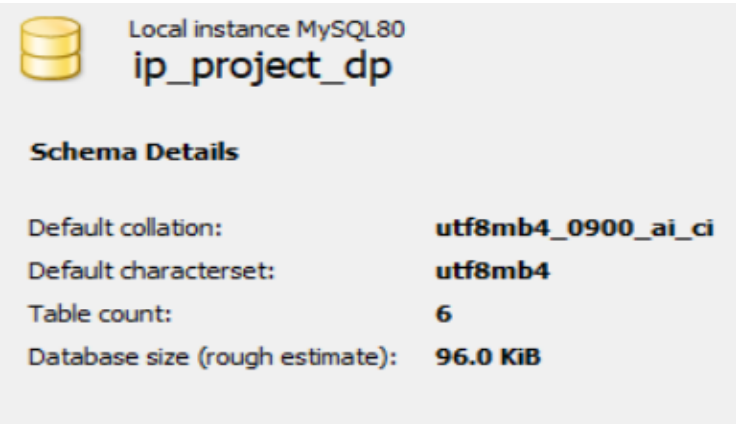
```
mysqli_query($query);
```

The command can be repeated again in the source code. All you need to do is to change the $query variable.

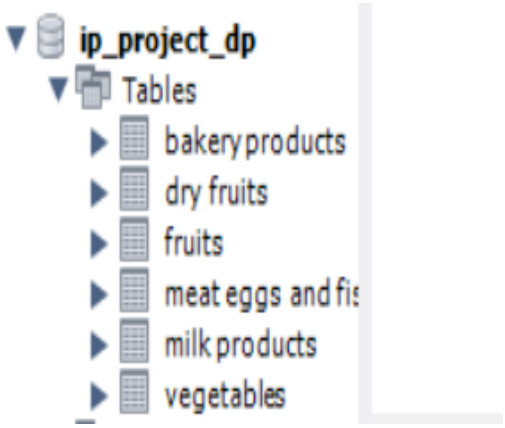
For example, here is the complete code that could be used to create a MySQL table in PHP:

```
<?php
$username = "your_username";
$password = "your_password";
$database = "your_database";
$mysqli = new mysqli("localhost", $username, $password, $database);
$query="CREATE TABLE tablename(id int(6) NOT NULL auto_increment,first varchar(15) NOT NULL,last varchar(15) NOT NULL,field1-name varchar(20) NOT NULL,field2-name varchar(20)NOT NULL,field3-name varchar(20) NOT NULL,field-name varchar(30) NOT NULL, field5-name varchar(30)NOT NULL,PRIMARY KEY (id),UNIQUE id (id),KEY id_2 (id))";
$mysqli->query("$query");
$mysqli->close();
?>
<?php
$servername = "localhost";
$username = "username";
$password = "password";
```

```
$dbname = "myDB";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
      echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>"; }
} else {
   echo "0 results";
}
mysqli_close($conn);
?>
```



(fig 7.1 Screenshot of MySQL)



(fig 7.2 Screenshot of Database Table)

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	Bakery_name	Calories	Portion_size_per_100g	Nutritional_content
S	Bread	80 kcal	1 slice of whole wheat bread (28 grams).	Whole wheat bread is a good source of complex...
34	Bagel	270 kcal	1 small bagel (85 grams).	Bagels are primarily made from refined flour and...
35	Croissant	165 kcal	small croissant (37 grams).	Croissants are made with butter and flour and a...
36	Muffin	200 kcal	1 small muffin (55 grams)	Muffins can vary widely in their nutritional conte...
37	Doughnut	210 kcal	1 small glazed doughnut (43 grams)	Doughnuts are typically high in calories, sugar, ...
38	Cinnamon Roll	260 kcal	1 small cinnamon roll (75 grams).	Cinnamon rolls are rich in sugar, fat, and calorie...
39	Scone	230 kcal	1 small scone (55 grams).	Scones are typically made with flour, butter, an...
40	Biscuit	90 kcal	1 small biscuit (25 grams).	Biscuits are a source of carbohydrates and fat, ...

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	Dry_fruit_name	Calories	Portion_size_per_100g	Nutritional_content
17	Almonds	160 kcal	28 grams	lmonds are a good source of healthy fats, prote...
18	Walnuts	185 kcal	28 grams	Walnuts are rich in omega3 fatty acids, antioxid...
19	Cashews	160 kcal	28 grams	Cashews offer healthy fats, protein, vitamin K, ...
20	Pistachios	160 kcal	28 grams	Pistachios provide protein, fiber, healthy fats, v...
21	Raisins	42 grams	129 kcal	Raisins are a good source of natural sugars, fib...
22	Dried Apricots	42 grams	80 kcal	Dried apricots are rich in vitamin A (as betacaro...
23	Prunes	42 grams	81 kcal	Prunes are a good source of dietary fiber, vita...
24	Dates	42 grams	120 kcal	Dates provide natural sugars, fiber, potassium, ...
NULL	NULL	NULL	NULL	NULL

(fig 7.3 Screenshot of Database)

Chapter 8

Local hosting

Local hosting, also known as on-premises hosting, is a method of hosting web applications, websites, and other online services on servers and infrastructure located within your own physical premises or local network. This approach offers a range of features and advantages that cater to specific needs and preferences. Here are some key features of local hosting:

1. Complete Control: One of the most prominent features of local hosting is the absolute control it provides over your hosting environment. You manage the hardware, software, configurations, and security measures according to your specific requirements. This level of control is essential for organizations with strict compliance or security needs.

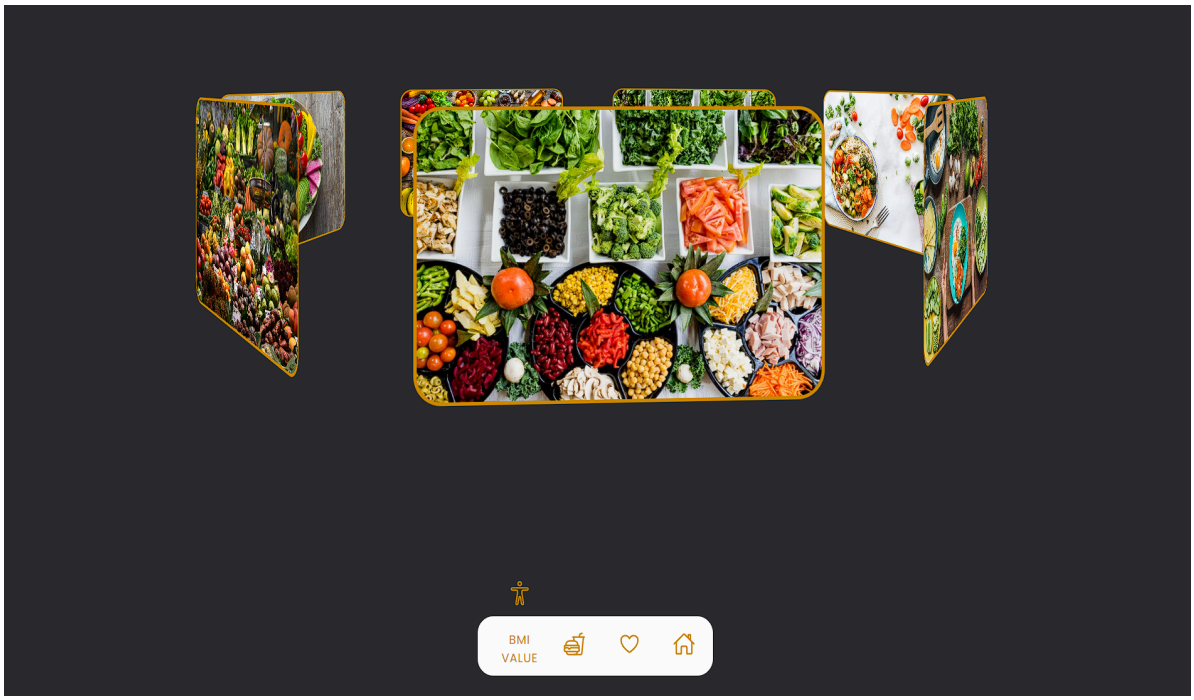
2. Enhanced Security: Local hosting allows you to implement tailored security measures to safeguard your web applications and data. Since you're responsible for the server's physical and digital security, you can establish robust firewalls, access controls, and encryption protocols, reducing the risk of security breaches.

3. Performance Optimization: You can fine-tune the performance of your web applications by optimizing the local server and network infrastructure. This can lead to faster loading times, better response rates, and improved user experiences, especially for high-traffic or resource-intensive applications.

4. Customization: Local hosting allows you to customize your server environment to meet the specific needs of your web applications. You can install, configure, and maintain software and services that precisely match your application's requirements, which may not be possible with standard shared hosting solutions.

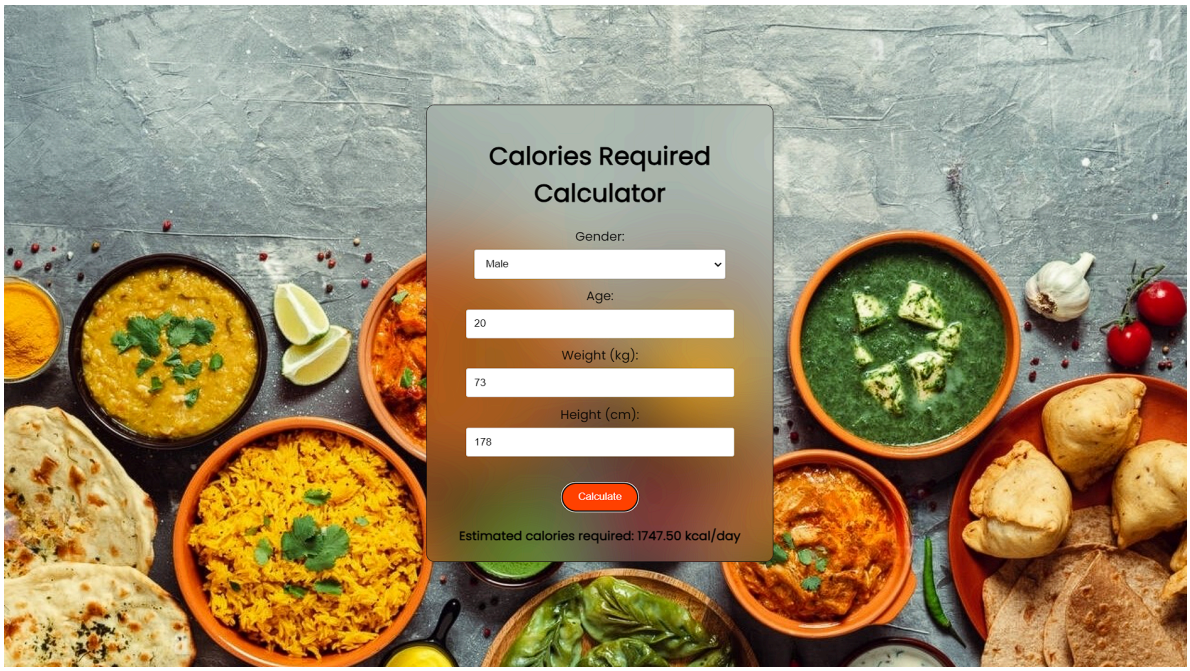
5. Flexibility: Local hosting is highly flexible. You can easily scale your infrastructure up or down as your needs change, allowing for a responsive and agile hosting solution.

6. Offline Access: Local hosting can offer a degree of offline access to your web applications, which can be essential for certain industries or businesses operating in areas with unreliable internet connectivity.

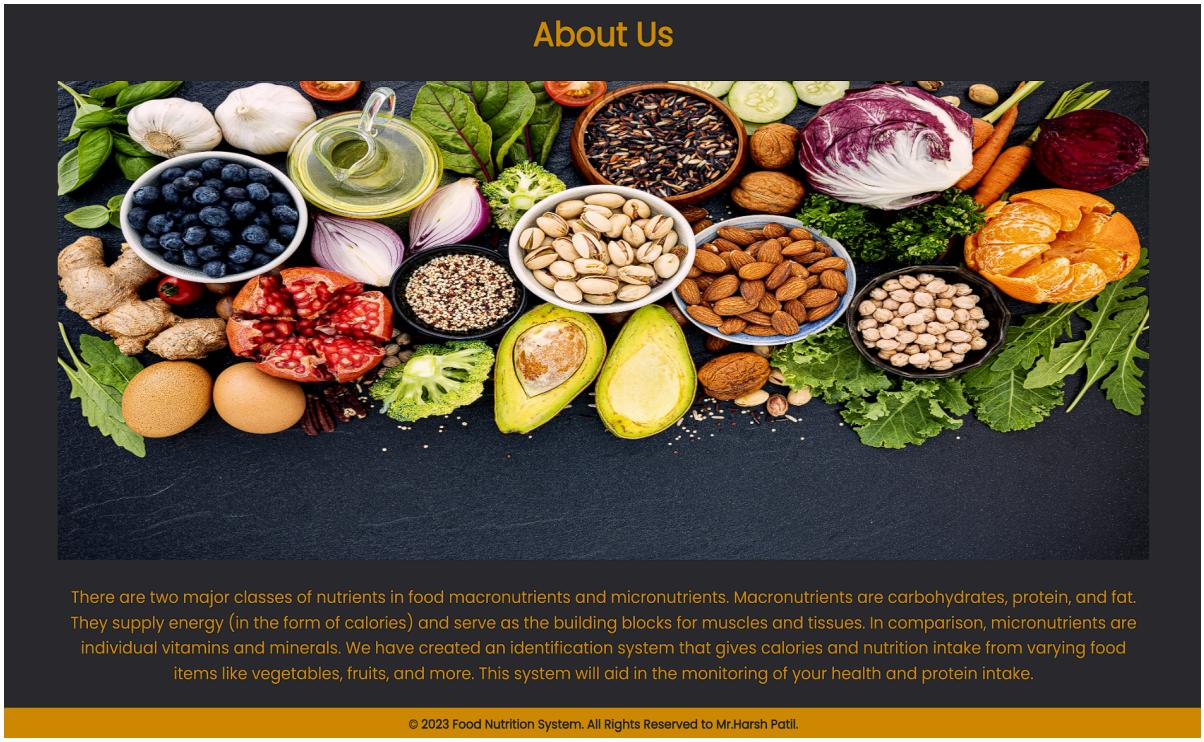


(Fig 8.1 Snapshot of Landing page)

(Fig 8.2 Snapshot of BMI Page)



(Fig 8.3 Snapshot of Calories Calculator Page)



(Fig 8.4 Snapshot of About us page)

Acknowledgement

We would like to express our special thanks to Prof. Prerna Kulkarni, our IP project guide who guided us through the project and who helped us in applying the knowledge that we have acquired during the semester and learning new concepts.

We would like to express our special thanks to Dr. Satishkumar Varma, Head of Department of Information Technology, who gave us the opportunity to do this IP project because of which we learned new concepts and their application.

Finally we would like to express our sincere gratitude to Principal Dr. Sandeep Joshi who gave us the opportunity and facilities to conduct this IP project.

Harsh Patil

Piyush Pandey

Rudein Salim

Raj kakodkar