

Nutritionist-Client App

A Project Report

Submitted by

Bhavya Chopra

Under the Guidance of

Pranav More

*in partial fulfillment for the award of the
degree of*

BACHELORS OF TECHNOLOGY

CE

At



**SCHOOL OF TECHNOLOGY MANAGEMENT AND
ENGINEERING, NAVI MUMBAI
APRIL 2023**

Specimen B

DECLARATION

I, Bhavya Chopra, Roll No. A001 B.Tech (CE), IV semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did write what. (Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidents of plagiarism in this body of work.

Signature of the Student:

Name: Bhavya Chopra

Roll No. A001

Place: NMIMS Navi Mumbai

Date: 5th April, 2023

Specimen C

CERTIFICATE

This is to certify that the project entitled “Nutritionist Client App” is the bonafide work carried out by Bhavya Chpora of B.Tech, School of Technology, Management & Engineering, NMIMS, Navi Mumbai, during the IVth semester of the academic year 2022-2023, in partial fulfillment of the requirements for the award of the Degree of Bachelors of Engineering as per the norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

Examiner 1

Pranav More

Examiner 2

Specimen D

Table of contents

CHAPTER NO.	TITLE	PAGE NO.
	List of Figures	i
	Abbreviations	ii
1.	INTRODUCTION	iii
	1.1 Project Overview	
	1.2 Hardware Specification	
	1.3 Software Specification	
2.	TECHNOLOGY USED	vi
3.	ANALYSIS & DESIGN	vii
	3.1 Analysis	
	3.2 Design	
4.	OUTPUT SCREENSHOTS	xi
5.	DATABASE SCHEMA	xiii
6.	CONCLUSION & FUTURE SCOPE	xiv
	6.1 - CONCLUSION	
	6.2 - FUTURE SCOPE	
7.	LINKS	xv

List of Figures		
CHAPTER NO.	TITLE	PAGE NO.
3.	Analysis and Design	vii
	Fig 3.1 - Validation Code	
	Fig 3.2 - Angular Code	
	Fig 3.3 - Angular in HTML	
	Fig 3.4 - Database connection	
4.	Output Screenshots	xi
	Fig 4.1 - Landing page	
	Fig 4.2 - BMI Calculator	
	Fig 4.3 - Login page	
	Fig 4.4 - Signup page	
5.	DATABASE SCHEMA	xiii
	Fig 5.1 - Database Schema	

NOTE:

- *Figures must be properly explained in the text*
- *Figure No. and caption of the figure must be below the figure in all the chapters.*

Abbreviations

Abbreviation	Description
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JS	Java Script

Chapter 1

Introduction

1.1 Project Overview

Vision

To create a user-friendly website that empowers clients to take control of their health through proper nutrition and enables the nutritionist to effectively manage and monitor client progress.

Mission

Our mission is to provide a comprehensive platform that allows clients to easily input and track their measurements, view their progress over time, and access personalized nutrition plans. The website will also provide a BMI calculator for new users to assess their health status and offer valuable resources and information on healthy eating habits.

Feature List

Current functionality of the app includes -

- 1) BMI Calculator: A landing page feature that allows new users to calculate their BMI (Body Mass Index) by inputting their height and weight. The website can provide instant feedback on the user's health status based on the calculated BMI.
- 2) User Registration and Login: The website should have a user registration and login system that allows clients to create an account with their personal information, including their name, contact details, and relevant health measurements. Clients can log in securely to access their account and view their progress.
- 3) Responsive Design: The website should be designed to be responsive, meaning it should adapt to different screen sizes and devices, such as desktops, laptops, tablets, and mobile phones, to ensure a seamless experience for users across various devices.

- 4) Feedback and Reviews: The website can allow clients to provide feedback and reviews on their experience with the nutritionist and the website, helping to improve the services and gather testimonials.
- 5) Appointment Scheduling: The website can include a feature for clients to schedule appointments with the nutritionist, providing an easy and convenient way for clients to book and manage their appointments.

1.2 Hardware Specifications

Server: A machine with sufficient processing power, memory, and storage to host the Node.js server and MongoDB database. Recommended specifications would be a multi-core processor, 8GB or more of RAM, and ample storage space for the database.

Network: Reliable internet connectivity to ensure smooth communication between the server and clients accessing the website.

1.3 Software Specification

Operating System: The server should be running a compatible operating system, such as Linux, Windows, or macOS, that supports the installation and operation of Node.js and MongoDB.

Node.js: The latest stable version of Node.js installed on the server, along with necessary dependencies and libraries for building server-side applications.

MongoDB: The latest stable version of MongoDB installed on the server, configured with appropriate security measures, such as authentication and encryption, to protect the stored data.

Front-end Technologies: HTML, CSS, and Bootstrap for building the user interface and responsive design of the website. JavaScript for client-side interactivity and dynamic content.

Back-end Technologies: Node.js for building the server-side logic and APIs for handling requests and responses. Express.js or any other web framework for Node.js for simplifying server-side development.

Authentication and Authorization: Passport.js or any other relevant library for implementing user authentication and authorization on the server-side, along with appropriate security measures, such as hashing and salting of passwords.

Database Connectivity: Mongoose or any other relevant library for connecting and interacting with the MongoDB database from Node.js.

Chapter 2

Technologies used

- Node.js: A JavaScript runtime environment that allows server-side execution of JavaScript code.
- Express.js: A popular web framework for Node.js that simplifies server-side development and handles routing and HTTP requests and responses.
- MongoDB: A NoSQL database management system that stores data in JSON-like documents and provides flexibility in storing and retrieving data.
- Mongoose: A library for Node.js that provides a convenient way to interact with MongoDB and perform CRUD operations on data.
- HTML, CSS, Bootstrap: HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and Bootstrap (a popular front-end framework) are used for designing and building the user interface of the website.
- JavaScript: A programming language used for client-side interactivity, form validation, and dynamic content generation on the website.

Chapter 3

Analysis and design

3.1 Analysis

The analysis phase involved gathering and analyzing the requirements from the nutritionist and other stakeholders. This included understanding the functionalities and features that the nutritionist wanted to include in the website, such as client measurement tracking, client login functionality, and BMI calculation. Additionally, technical requirements such as the use of Node.js for server-side development, MongoDB for database management, and HTML, CSS with Bootstrap for front-end development were identified. The team also conducted a thorough review of a pre-existing website that was used as a reference for design inspiration, analyzing its layout, features, and user experience to gather insights for the new nutritionist website.

3.2 Design

The design phase involved creating a comprehensive design plan for the website, incorporating the requirements identified during the analysis phase and drawing inspiration from the pre-existing website reference. The following design aspects were considered:

- **Data Modeling:** A data model was designed to define the structure of the database, including the entities and their relationships, such as clients, measurements, and other relevant data. This ensured efficient storage and retrieval of data from the MongoDB database.
- **System Architecture:** The system architecture was designed using Node.js and Express.js for server-side development, with appropriate routing and middleware for handling HTTP requests and responses. The front-end was designed using HTML, CSS with Bootstrap, and JavaScript for interactivity and dynamic content generation.
- **User Interface (UI) Design:** The UI design followed the inspiration from the pre-existing website, incorporating a clean and visually appealing layout with easy-to-navigate menus, forms for data input, and visual elements that enhanced the user experience. The use of Bootstrap ensured responsiveness, making the website accessible across various devices.
- **Interaction Design:** Interaction design was focused on providing an intuitive and seamless user experience. This included designing forms with input validation, error handling, and

feedback mechanisms to guide users through the process of calculating BMI and managing their measurements.

3.3 Codes

```
<script type="text/javascript">
function validate() {
    var fname = document.getElementById("fname").value;
    var fnameout = document.getElementById("fnameout");

    var pass = document.getElementById("pass").value;
    var passout = document.getElementById("passout");

    var cpass = document.getElementById("cpass").value;
    var cpassout = document.getElementById("cpassout");

    var phone = document.getElementById("phone").value;
    var phoneout = document.getElementById("phoneout");

    var remail = /. * @ . * \ . * . + /;
    var rename = /^[A-Za-z]{3,10}$/;
    var repass = /^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&(){}]).{8,32}$/;
    var rephone = /^\d{10}$/;

    if (rename.test(fname)) {
        fnameout.innerHTML=""
    }
    else {
        fnameout.innerHTML="Not a valid first name"
    }

    if (rephone.test(phone)) {
        phoneout.innerHTML=""
    }
    else {
        phoneout.innerHTML="Not a valid phone number"
    }

    if (repass.test(pass)) {
        passout.innerHTML=""
    }
    else {
        passout.innerHTML="Not a valid Password, have a capital, number and a symbol"
    }

    if (pass == cpass) {
        cpassout.innerHTML=""
    }
    else {
        cpassout.innerHTML="Password do not match"
    }
}
</script>
```

Fig 3.1 - Validation Code

```

1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-bmi-calculator',
5    templateUrl: './bmi-calculator.component.html',
6    styleUrls: ['./bmi-calculator.component.css']
7  })
8  export class BmiCalculatorComponent {
9    age: number;
10   height: number;
11   weight: number;
12   male: boolean;
13   female: boolean;
14   result: string;
15   bmi: number;
16   modalDisplay: string;
17
18   calculate() {
19     if (!this.age || !this.height || !this.weight || (!this.male && !this.female)) {
20       alert("All fields required to calculate BMI");
21     } else {
22       this.countBmi();
23     }
24   }
25
26   countBmi() {
27     const heightInMeter = this.height / 100;
28     this.bmi = this.weight / (heightInMeter * heightInMeter);
29
30     if (this.bmi < 18.5) {
31       this.result = 'Underweight';
32     } else if (this.bmi >= 18.5 && this.bmi <= 24.9) {
33       this.result = 'Healthy';
34     } else if (this.bmi >= 25 && this.bmi <= 29.9) {
35       this.result = 'Overweight';
36     } else if (this.bmi >= 30 && this.bmi <= 34.9) {
37       this.result = 'Obese';
38     } else if (this.bmi >= 35) {
39       this.result = 'Extremely obese';
40     }
41
42     this.modalDisplay = 'block';
43   }
44
45   closeModal() {
46     this.modalDisplay = 'none';
47   }
48 }
49

```

Fig 3.2 - Angular JS Code

```

<div>
  <input [(ngModel)]="age" id="age" type="number" placeholder="Age">
  <input [(ngModel)]="height" id="height" type="number" placeholder="Height (cm)">
  <input [(ngModel)]="weight" id="weight" type="number" placeholder="Weight (kg)">
  <label>
    <input [(ngModel)]="male" id="m" type="radio" name="gender">Male
  </label>
  <label>
    <input [(ngModel)]="female" id="f" type="radio" name="gender">Female
  </label>
  <button (click)="calculate()">Calculate</button>
</div>

<div class="comment" [style.display]="modalDisplay">
  You are <span id="comment">{{result}}</span>
</div>

<div id="myModal" class="modal">
  <div class="modal-content">
    <span class="close" (click)="closeModal()">&times;</span>
    <h2 id="modalText">{{bmi.toFixed(2)}}</h2>
  </div>
</div>

```

Fig 3.3 - Angular implementation in HTML

```

const mongoose = require("mongoose")

mongoose.connect("mongodb://localhost:27017/LoginFormPractice")
  .then(()=>{
    console.log('mongoose connected');
  })
  .catch((e)=>{
    console.log('failed');
  })

const logInSchema = new mongoose.Schema({
  "name":{
    type:String,
    required:true
  },
  "password":{
    type:String,
    required:true
  }
})

module.exports=mongoose.model(['LogInCollection',logInSchema])

```

Fig 3.4 - Database connectivity code

Chapter 4

Output

Key2Health

[Home](#) [Review](#) [About Us](#) [BMI](#) [Login](#)

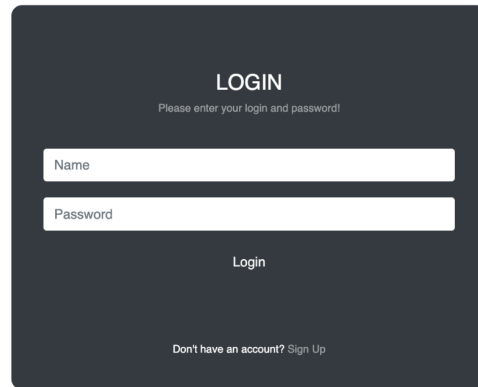


Fig 4.1 - Landing page

' Calculate Your BMI'

Age :	20	<input checked="" type="radio"/> Male <input type="radio"/> Female	
Height :	12	Weight :	21
<input type="button" value="Calculate BMI"/>			
Your BMI Is 1458.33 You Are Extremely Obese			

Fig 4.2 - BMI Calculator



A dark-themed login form with a white title 'LOGIN' and a subtitle 'Please enter your login and password!'. It contains two white input fields labeled 'Name' and 'Password', followed by a 'Login' button. At the bottom, there is a link 'Don't have an account? Sign Up'.

LOGIN
Please enter your login and password!

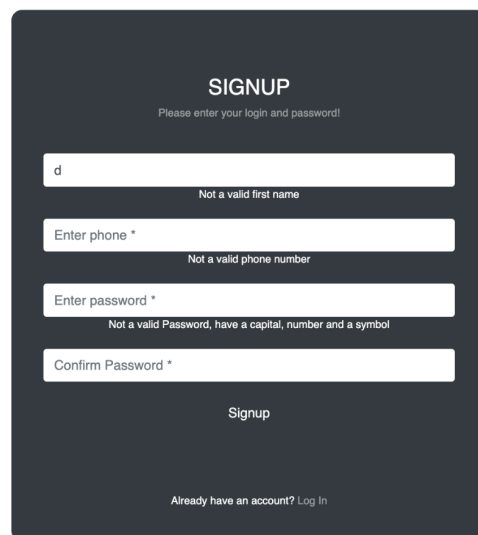
Name

Password

Login

Don't have an account? [Sign Up](#)

Fig 4.3 - Login page



A dark-themed signup form with a white title 'SIGNUP' and a subtitle 'Please enter your login and password!'. It contains four white input fields: 'Name' (with value 'd' and error 'Not a valid first name'), 'Enter phone *' (with error 'Not a valid phone number'), 'Enter password *' (with error 'Not a valid Password, have a capital, number and a symbol'), and 'Confirm Password *'. Below the fields is a 'Signup' button. At the bottom, there is a link 'Already have an account? Log In'.

SIGNUP
Please enter your login and password!

d
Not a valid first name

Enter phone *
Not a valid phone number

Enter password *
Not a valid Password, have a capital, number and a symbol

Confirm Password *

Signup

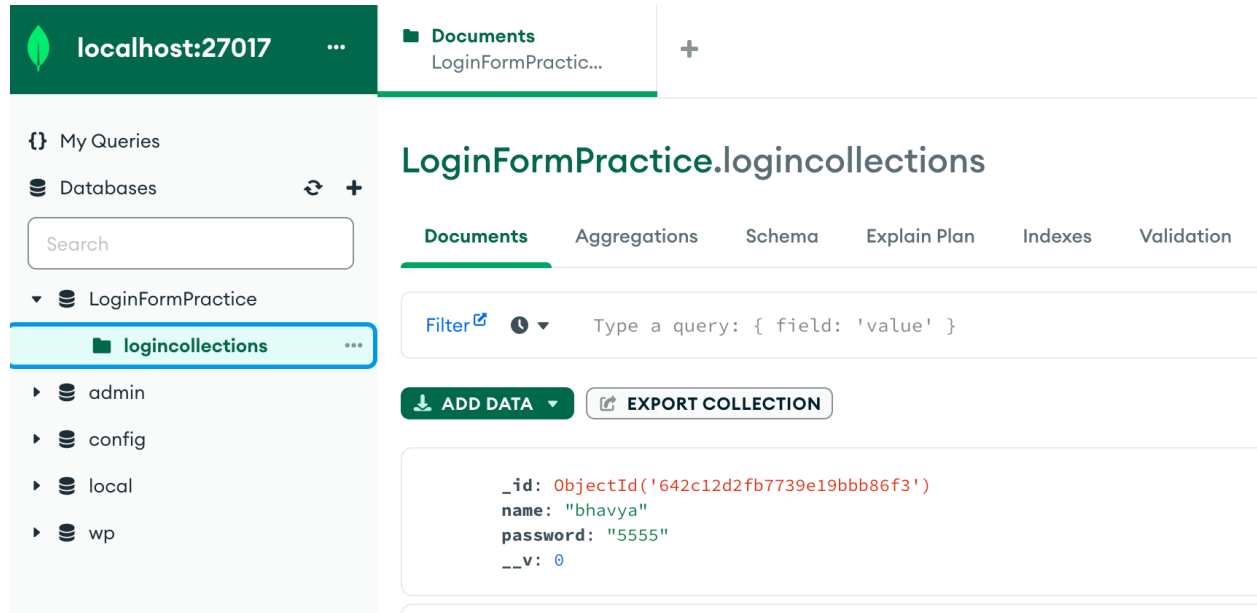
Already have an account? [Log In](#)

Fig 4.4 - Signup page

Chapter 5

Database Schema

Fig 5.1 - an image showing the database schema



Chapter 6

Conclusion and Future scope

6.1 Conclusion -

In conclusion, the nutritionist website project was a successful endeavor that resulted in the development of a functional and user-friendly website for managing client measurements and providing clients with access to their details. The project utilized technologies such as Node.js, MongoDB, HTML, CSS with Bootstrap, and followed best practices in data modeling, system architecture, UI design, interaction design, security design, and test design. The analysis and design phase played a crucial role in setting the direction for the project, ensuring that the website met the requirements and goals of the nutritionist. Through regular reviews and iterations, the design plan was refined to deliver a robust and efficient website. The completion of the project marked a significant achievement in creating a valuable tool for the nutritionist to manage her clients and provide a seamless experience for clients to view their information.

6.2 Future Scope-

- 1) Client Profile Management: Registered clients should be able to update their profile information, including their measurements, goals, and preferences. They can also upload profile pictures and view their personal details in a private and secure manner.
- 2) Measurement Tracking: Clients should be able to input and track their measurements, such as weight, height, body fat percentage, waist circumference, etc., on a regular basis. The website should provide graphical representations of the data to visualize progress over time.
- 3) Reporting and Analytics: The website should provide reporting and analytics features that allow the nutritionist to generate reports on client progress, trends, and outcomes. This can help in evaluating the effectiveness of the nutrition plans and making data-driven decisions.

Chapter 7

Links

Github Link -

<https://github.com/bhavyachopra99/nutritionist-app.git>

Demo Video Link -

https://drive.google.com/file/d/1PXDLELUOT7NBHSG5DuBl36M9UtdbIvv8/view?usp=share_link