



### VIVEKANAND COLLEGE FOR

**ADVANCED COMPUTER AND INFORMATION SCIENCE SURAT VEER NARMAD SOUTH GUJARAT UNIVERSITY, SURAT**

**PROJECT REPORT ON**

**POWER PLUS GYM**

### AS A PARTIAL REQUIREMENT FOR THE DEGREE OF

**BACHELOR OF COMPUTER APPLICATION (B.C.A) 2022 - 2023**

**GUIDED BY : SUBMITTED BY :**

**→ Khyati Kariya → Sartanpara Sahil C.**

**→ Saliya Dashrath R.**

**→ Zanzmera Tushal J.**

**→ Savani Dharmik A.**

1



# ACKNOWLEDGMENT

The reason for completing the project work

success fully is not just our e fforts but the e fforts of many people. The people t rusted, guided, and encouraged us with every means.

A guide is a pers on who provides you the direction towards

success, so we feel great pleas ure to express our gratitude to a l l our guides, our faculty me mbers as well as every person who

helped me directly or indirectly with our project.

We would l ike to th ank **Mrs. Sumathi Shenoy (** Ca mpus director of Vivekanand college **) , Mr. Sandip Surti (** Incharge

Principal of Vivekanand college **)** heart fully, who granted us the project. Her skills and experience were a guiding path in this

learning process. She made cons tant e fforts to s hape upour s kills as per the indus t ry s tandards. She provided us with a very ho mely and fr iendly environment which made it the best place to work.

We are a lso indebted to our Professor Khyati Kariya who provided cons tant encouragement, s upport & valuable guidance before and during our project. I t was his e f fort that led us to this place for project work. His guidance and s uggestions were invaluable.

Thank you very much.

**Sartanpara Sahil C.**

**Saliya Dashrath R.**

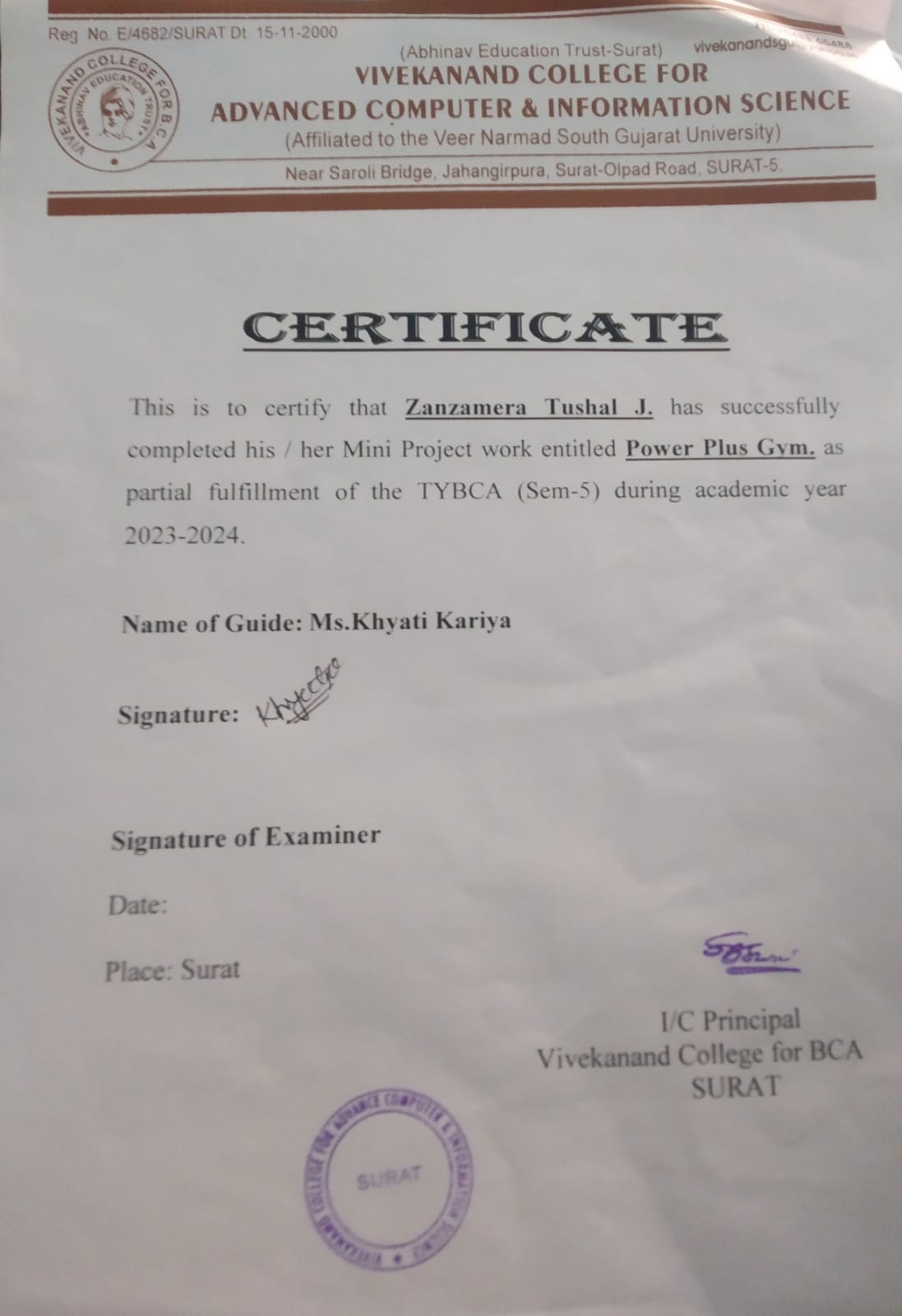
**Zanzmera Tushal J.**

**Savani Dharmik A.**

2



# INDE

****

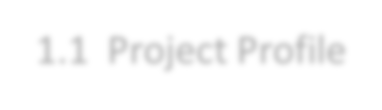
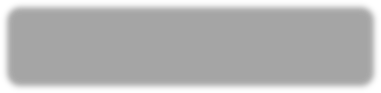
# X

|  |  |  |
| --- | --- | --- |
| C ha pt er | Par t icu la r | Page No |
|  |  |  |
| 1 | **Int rodu ct i on** |  |
|  | 1 . 1 Pr o je c t Pr o f i le | 4 |
| 1 . 2 Pro je c t I nt r o duct io n | 5 |
| 2 | **Envi ron ment D es c ript i on** |  |
|  | 2 . 1 Har dwar e / So ft war e r equ i r e me nt | 6 |
| 2 . 2 Too ls & T e c hno lo g y | 8 |
| 3 | **P ro pos ed Sy st e m** |  |
|  | 3 . 1 Sco pe | 16 |
| 3 . 2 Aim& O bje c t ive | 17 |
| 3 . 3 E xcept ed Ad va nt age | 19 |
| 3 . 4 Requ i r e me nt Spe c i f ic a t io n | 20 |

3



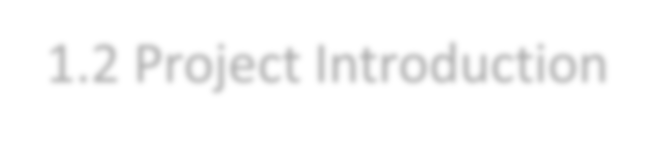
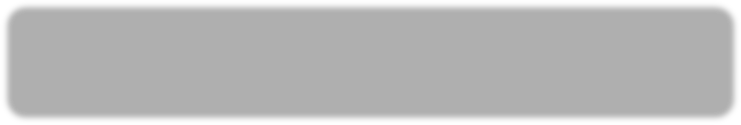
**INTRODUCATION**



1.1 Project Profile

|  |  |
| --- | --- |
| **Project Title :** | **POWER PLUS GYM** |
| **Project Type :** | **Web B ase d P roj e ct** |
| **P urpose :** | **B CA ( 5 th S EM )** |
| **Technology :** | **React , Nod e . Js, Javascript** |
| **Front End :** | **React** |
| **Back End :** | **E xp res s , M ongo Db , Nodejs** |
| **Other Tools Used :** | **Visua l st udio** |
| **Framework :** | **React** |
| **Guided By :** | **As st . P ro f Khyati Kariya** |
| **Submitted To :** | **Vive ka nand Co l l age Fo r Advan ced C omput e r & Inf o rmat i on S c i en ce , Su rat .** |
| **Develop By:** | **Sartanpara Sahil**  **Saliya Dashrath**  **Zanzmera Tushal**  **Savani Dharmik** |

4



1.2 Project Introduction

* **Project Definition :**

**Gyms are become the essential part of our lives, providing best exercise and body**

**building facilities to our society. Therefore, at the management end there are some necessary steps to maintain the records of every individual including trainer, trainees, and staff But maintaining the records on paper is very difficult So, it is necessary to have a computerized system that manages all these issues.**

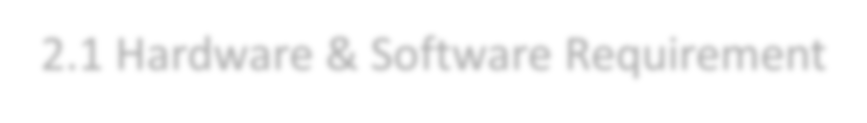
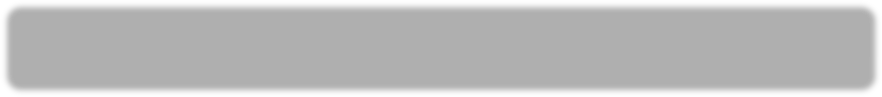
**Thus working on the management system for Gym Industry are the basis of our  
project. We have developed an automated version of the manual system, named as Gymnasium Management System. This system also provides excellent security of data at to its user.**

* **Features :**
  1. **Admin Panel Feature :**
     + All user data.
     + Sales analysis.
     + Can add product to mongoDB.
     + Can delete product.
     + Can delete user.

5



# ENVIRONMENT DESCRIPTION



2.1 Hardware & Software Requirement

The Efficient Hardware And Software Configuration Requires To Run The Sys te m Is As Suggest Below. The

Configuration Sugges ted Is For Better Performance. Same Functionality Or Higher Configuration Will Always Better

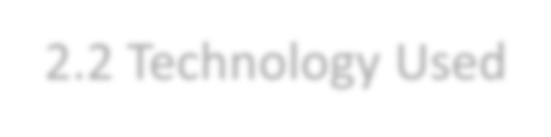
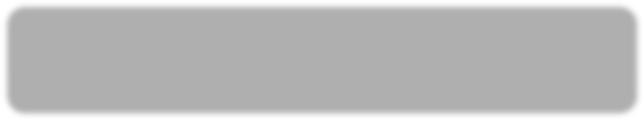
* Hardware :
* Client Side :
  + Internet- enabled device.
* Server Side :
  + RAM: 2 GB
  + Disk Space: 10 GB Minimum
* Development Side :
  + Hard Disk: 30 GB
  + RAM: 8 GB
  + Processor: i 5 7 th Generation

6



* Software:
* Client Side :
  + Client On Internet: Chrome any Browser .
* Server Side :
  + Client On Internet: Chrome any Browser
  + Server: Node Js
  + Database: Mongo DB Atlas
* Development Side :
  + Operating System : window 10
  + Front- end : React.Js, Javascript, Css
  + Backend : Node Js, MongoDB
  + Software : Visual studio

7



2.2 Technology Used

* **Overview Of React :**

React is a popular Java Script l ibrary used to build user

interfaces for web applications. I t was developed by Facebook and is now widely us ed by developers a l l over the world .

React uses a " Virtual DOM " to e ff ic iently update and render these co mponents on the web page. Instead of directly

manipulating the actual DOM ( Document Object Model), React creates a virtual representation of the DOM in me mory. When

there are changes to the data or s t ate of a co mponent, React

compares the virtual DOM with the real DOM and updates only the necessary parts, making the process much faster.

One of the essential concepts in React is the concept of

" one - way data f low. " This means that data f lows fro m the parent components down to their child components. When data changes in the parent co mponent, React automatically updates the child

components that depend on that d a ta.

React a ls o s upports a declarative programming s tyle. Ins tead of explicitly describing how to update the DOM, you define what the UI s hould look l ike at any given t ime, and React takes care of the rest. This makes the code eas i er to read and maintain.

Another s ignificant feature of React is i ts ability to work

seamless ly with other l ibraries and frameworks. I t can be used in

8



combination with tools l ike Redux for s tate manage ment, React Router for handling navigation, and more.

Overall, React s implifi es the process of building interactive and dyna mic user interfaces by providing a c lean and e ff ic ient

way to manage and update UI co mponents. This has made i t the go- to choice for many developers when building modern web

applications.

## Overview Of Node. JS:

Node. js is an open - source, cross - platfor m Java Script

runtime environment that a l lows developers to run Java Script

code outs ide of a web brows er. In s imple te r ms, Node. js le ts you use Java Script on the server - s ide,

which means you can build server appl ications us ing the sa me language that you use for front - end web develop ment.

Traditionally, Java Script was primarily used for c l ient - s ide scripting in web brows ers. However, Node. js extends its

capabilities by running Java Script code on the server. This

enables developers to build server applications, web servers, APIs, and even command - l ine tools us ing Java Script.

One of the key features of Node. js is i ts non - blocking,

event- driven architecture. Ins tead of handling tas ks sequentially, Node. js uses an as ynchronous approach. I t can perform multiple tasks concurrently, a l lowing i t to handle many requests without

9



gettin g s tuck waiting for a particular operation to complete. This makes Node. js highly scalable and e ffic ient, es pecially for

applications with a large number of concurrent connections.

Node. js co mes with a built - in package manager called npm ( Node Package Manager) that a l lows developers to eas i ly install, manage, and s hare reusable packages of code. This extens ive

ecos ystem of open- source modules and l ibraries makes i t e ffortless to integrate various functionalities into your

applications.

Since Node. js us es the sa me language ( Java Script) both on

the front - end and back- end, i t a ls o makes i t eas ie r for developers to switch between front - end and back- end development and

pro motes code reusability.

Overall, Node. js has beco me a popular choice for building

server- s ide applications due to its e ffic iency, scalability, and the familiarity of us ing Java Script for both c l ient - s ide and server-

s ide develop ment. I t has s ignificantly contributed to the growth of modern web development, making i t eas ie r to build real - t ime, fast, and feature - r ich web applications.

## Express. Js

Express. js , commo nly referred to as Express, is a minimalistic

and fast web application framework for Node. js . I t provides a set of robust features and utilities that s implify the process of

building web applications and APIs. As one of the mos t popular Node. js frameworks, Express has a large co mmunity of

developers and a r ich ecos ys tem of plugins and middleware.

10



Here' s an Feature of Express. js in s imple te r ms :

### Web Application Framework:

Express is a fra mework that runs on top of Node. js and enables developers to c reate web applications and APIs

us ing J ava Script. I t handles the complexities of handling HTTP reques ts and res ponses, making i t eas ier to build

server- s ide applications.

### Middleware:

One o f the essential concepts in Express is

middleware. M iddleware functions are s mall pieces of code that can intercept inco ming HTTP requests and modify the response or perform s pecific tasks before pass ing the

request to the next middleware in the chain.

This a l lows developers to add various functionalities to their applications, s uch as logging, authentication, e r ror

handling, and more.

### Routing:

Express provides an eas y - to- us e routing s yste m that

a l lows developers to define different endpoints ( URL paths ) and associate the m with s pecific callback functions. When a request is made to a

particular URL, Express matches i t with the corresponding route and executes the ass ociated callback.

### HTTP Methods :

Express s upports the s tandard HTTP methods l ike GET, POST, PUT, DELETE, e tc. You can c reate routes that handle specific HTTP methods and their corresponding actions.

11



### Template Engines:

While Express i tse l f does not co me with a specific

template engine, i t a l lows developers to use various popular template engines l ike EJS, Handlebars, Pug ( formerly Jade), and more. Te mplate engines help in rendering dyna mic

HTM L pages by injecting data into tem plates.

### Error Handling:

Express provides a s t ra ightforward way to handle errors in applications. You can define e r ror - handling middleware that will be invoked whenever an error occurs during the

request- response cycle.

## Overview of Mongo DB :

Mongo DB is a popular open- source No SQL database management s yste m that s tores data in a f lexible and scalable

manner. I t belongs to the fa mily of document - oriented databases, meaning i t s tores data in J SON - l ik e documents rather than

t raditional tables and rows.

Here So me Features of M ongo DB .

### Document- Oriented:

Mongo DB s tores data in documents, which a re s imilar to JSON objects. Each document can have i ts own s t ructure

and f ie lds, making i t eas y to s tore and manage complex data.

12



### NoSQL:

Mongo DB is a No SQL database, which means i t does not us e the t raditional SQL ( Structured Query Language) used in re lat ional databases. Ins tead, it provides a s imple

and powerful query language that a l lows you to interact with the data us ing Java Script - l ike s yntax.

### Scalability:

Mongo DB is des igned to handle la rge a mounts of data and high t raffic by scaling horizontally. I t can dis t r ibute

data across multiple servers, a l lowing i t to handle the ever - increas ing data demands of modern applications.

### Dynamic Schema:

Unlike t raditional re lat ional databases that require a

f ixed schema with predefined tables and columns, Mongo DB has a dynamic sche ma. This means that documents within a

collection can have different f ie lds, giving developers the f lexibility to change the data s t ructur e as needed.

## Mongo DB Atlas

Mongo DB Atlas is a fully managed c loud - based database

service provided by Mongo DB, Inc. I t a l lows developers to eas i ly deploy, manage, and scale Mongo DB databases without the need

to worry about the underlying infr astructure. M ongo DB Atlas is des igned to s implify the database management process and

provide a sea mless experience for developers, making i t a popular choice for hosting Mongo DB databases in the c loud.

13



1. Fully Managed:

Mongo DB Atlas is a fully managed database service.

This means that Mongo DB handles all the operational aspects of database manage ment, including backups, monitoring, security, patches, and

updates. As a developer, you can focus more on building your application and less on database adminis t ration.

1. Scalability:

With M ongo DB Atlas, you can easily scale your databases to acco mmodate changes in data volume and

t raffic. I t s upports horizontal scaling, a l lowing you to add more nodes or c lusters to handle in creas ed de mand.

1. Security:

Mongo DB Atlas provides robust s ecurity features to protect your data. I t s upports encryption at rest and in

t rans i t , ens uring that your data is securely s tored and

t rans mitted. Additionally, i t offers f ine - grained access

14



controls and integration with popular identity providers for user authentication.

**CSS :**

Cascading Style Sheets ( CSS) is a W 3 C s tandard for

defining the of web documents. Presentation refers to the way a document is displayed or delivered to the user, whether i t ’ s on a

computer monitor, a cell phone display, or read a loud by a screen reader.

This book focuses primarily on the vis ual aspects of

presentation, s uch as typography, colors, a l ignment, layout, and so on. CSS is the mechanis m for providing thes e types of s tyle

instructions to e le ments in a document that has been marked up with XHTM L, HTM L, or any XM L language.

Most important, CSS keeps these presentation instructions

separate from the content and i ts s t ructural and semantic markup.

Separation of for matt ing and content makes i t poss ib le to present the sa me markup page in different s tyles for different rendering methods, s uch as onscreen, in print, by voice( via

speech- based browser or screen reader) and on Braille - based tactile devices.

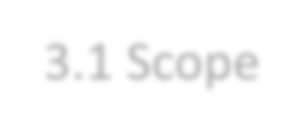
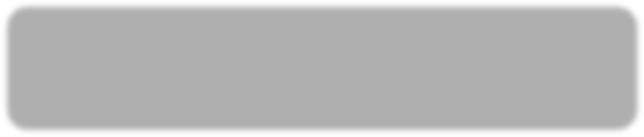
I t can a lso display the web page differently depending on the screen s ize or viewing devices . Reader can a ls o specify a

different s tyle s heets, s uch as a CSS f i le s tored on their own computer, to override the one the author s pecified.

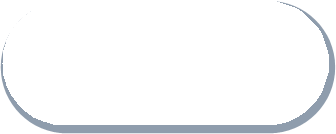
15



**3)Proposed System**



3.1 Scope

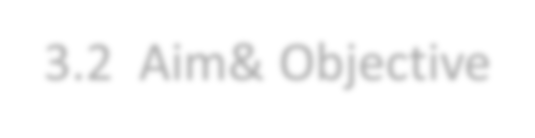
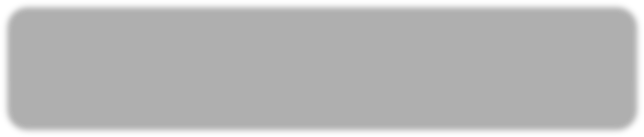


**ADMIN**

**CUSTOME**

**VISITOR**

16



3.2 Aim& Objective

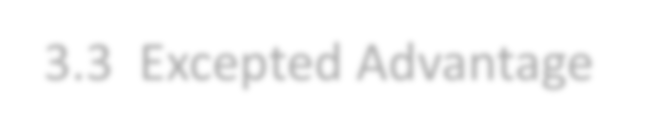
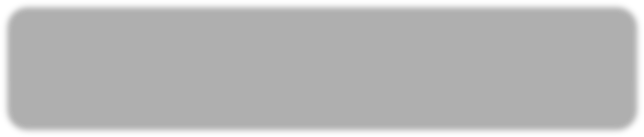
### High- level Security:

` Our webs i te ens ures the security and protection of our users, ens uring that there are no chances of any spyware or malicious

software that could potentially compro mise sens it ive data, s uch as bank or UPI- re lated infor mation. We prioritize the safety of our users ' data and s t r ive to maintain a secure and t rus tworthy

online environment for a l l our vis i tors.

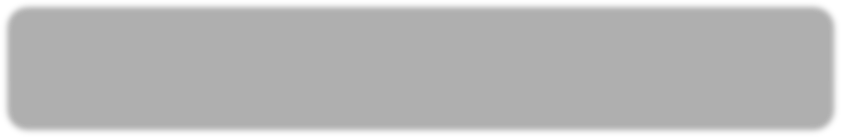
18



3.3 Excepted Advantage

* **To remove the manual or paper work in the Fitness club.**
* **Provide a platform with interactive user interface for both customer and admin.**
* **Save the time of both admin and customer.**
* **Get online plans in effective manners.**
* **To save cost of each user.**
* **The one who cannot afford gym packages ,also get benefits by viewing different exercise , diet plan tutorials.**
* **User friendly.**

19



3.4 Requirement Specification

### Admin:

* + - All user data.
    - Sales analysis.
    - Can add product to mongoDB.
    - Can delete product.
    - Can delete user.

### Visitor :

* + Vis i tor can view webs i te.
  + Vis i tor can view the plan.
  + Vis i tor can view the product.
  + Vis i tor can not purchase product.
  + Vis i tor can as k query.

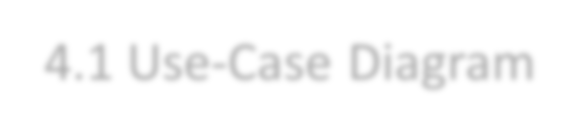
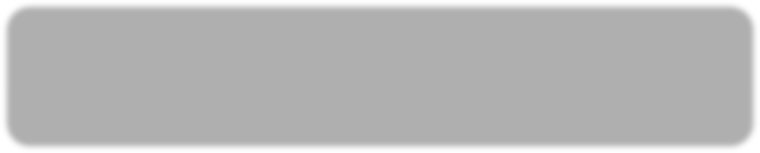
### Cus tomer :

* + User database, purchase history.
  + Register new account.
  + Login.
  + Cany bye product.

20

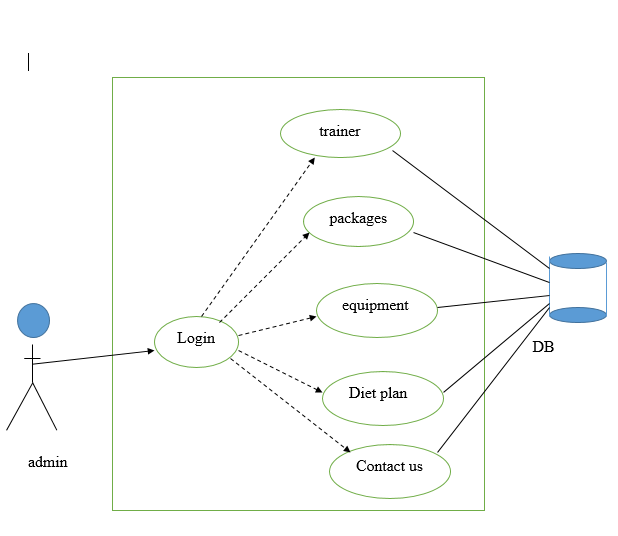


# SYSTEM MODEL ARCHITECTURE



4.1 Use-Case Diagram

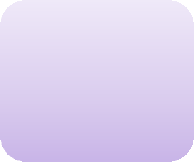
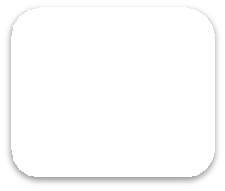
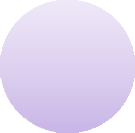
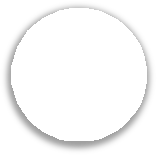
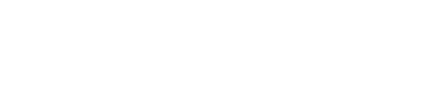
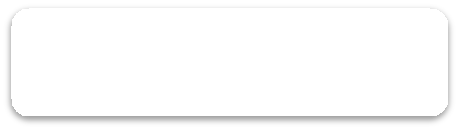
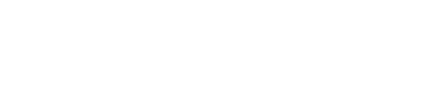
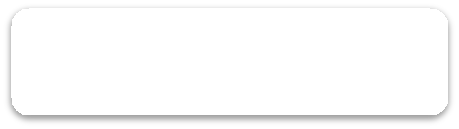
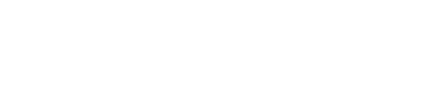
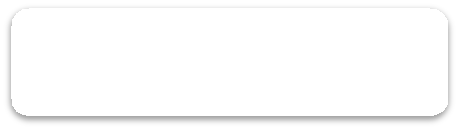
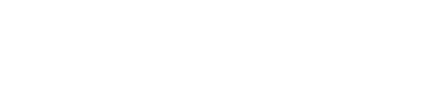
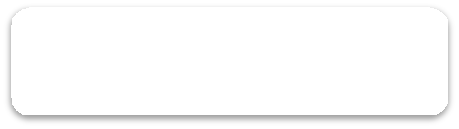
1. **Admin:**

****

21



1. **Visitor:**



View Stone

Like Stone

Ask Query

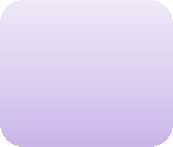
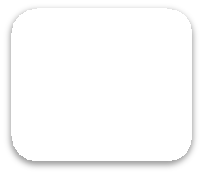
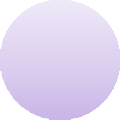
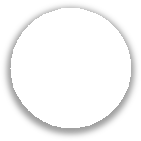
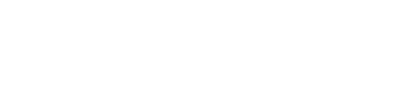
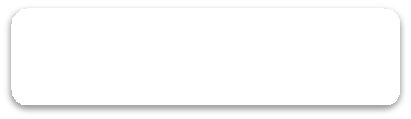
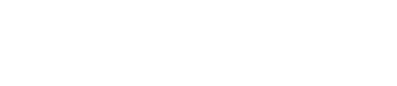
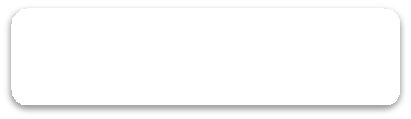
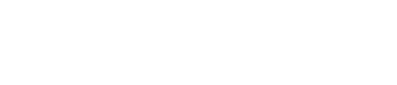
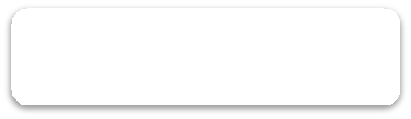
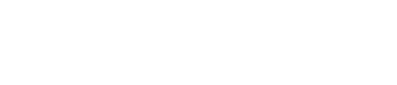
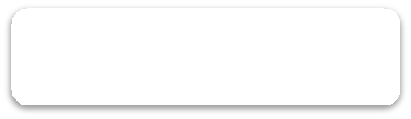
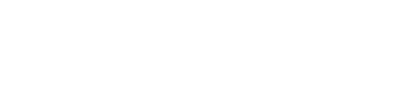
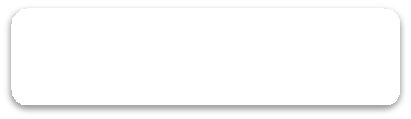
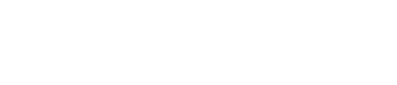
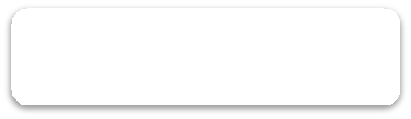
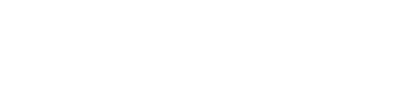
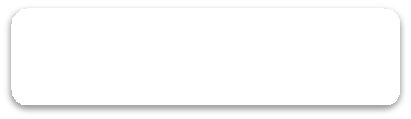
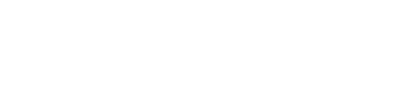
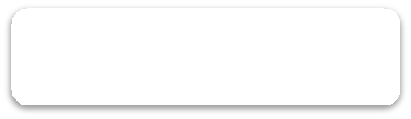
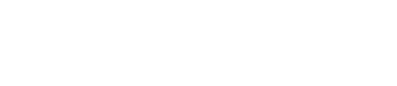
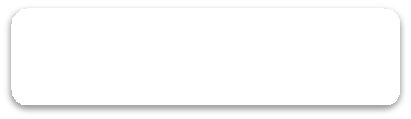
**Visitor**

Give Feedback

22



1. **Customer:**



Register

Login

View Product

Buyproduct

**Customer**

Ask Query

Give Feedback

Payment

Manage Profile

Logout

23



# COLLECTIONS AND THEIR SCHEMA

**Collection : 01 - User :**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| \_id | Object | \_id Field Stores Object Values. |
| Username | String | Name Field Stores String values. |
| Email | string | Email Field Stores String Values. |
| Password | String | Password Field Stores String Values. |
| Weight | Int | Weight Field Stores Integer Values. |
| Height | Int | Height Field Stores Integer Values. |
| Gender | String | Gender Field Store String Values. |

24



**Collection : 02 – Product :**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| \_id | Object | \_id Field Stores Object Values. |
| Name | String | Name Field Stores String values. |

25



**Collection : 03 - Plan :**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| \_id | Object | \_id Field Stores Object Values. |
| catagory | String | Category Field Stores String Values. |
| title | String | Title Field Stores String Values. |

26



**Collection : 04 – Order :**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| \_id | Object | \_id Field Stores Object Values. |
| Product Name | String | Product Name Field Stores String Values. |
| Product Price | Int | Product Price Field Stores Integer Values. |

28



**Collection : 05 - Carts:**

|  |  |  |
| --- | --- | --- |
| **Field\_Name** | **DatType** | **Description** |
| \_id | Object | \_id Field Stores Object Values. |
| Email | String | Email Field Stores String Values. |
| Cart | Array | Cart Field Stores Array Values. |
| Purchese | Array | Purchese Field Stores Array Values. |
| Wishlist | Array | Wishlist Field Stores Array Values. |

29

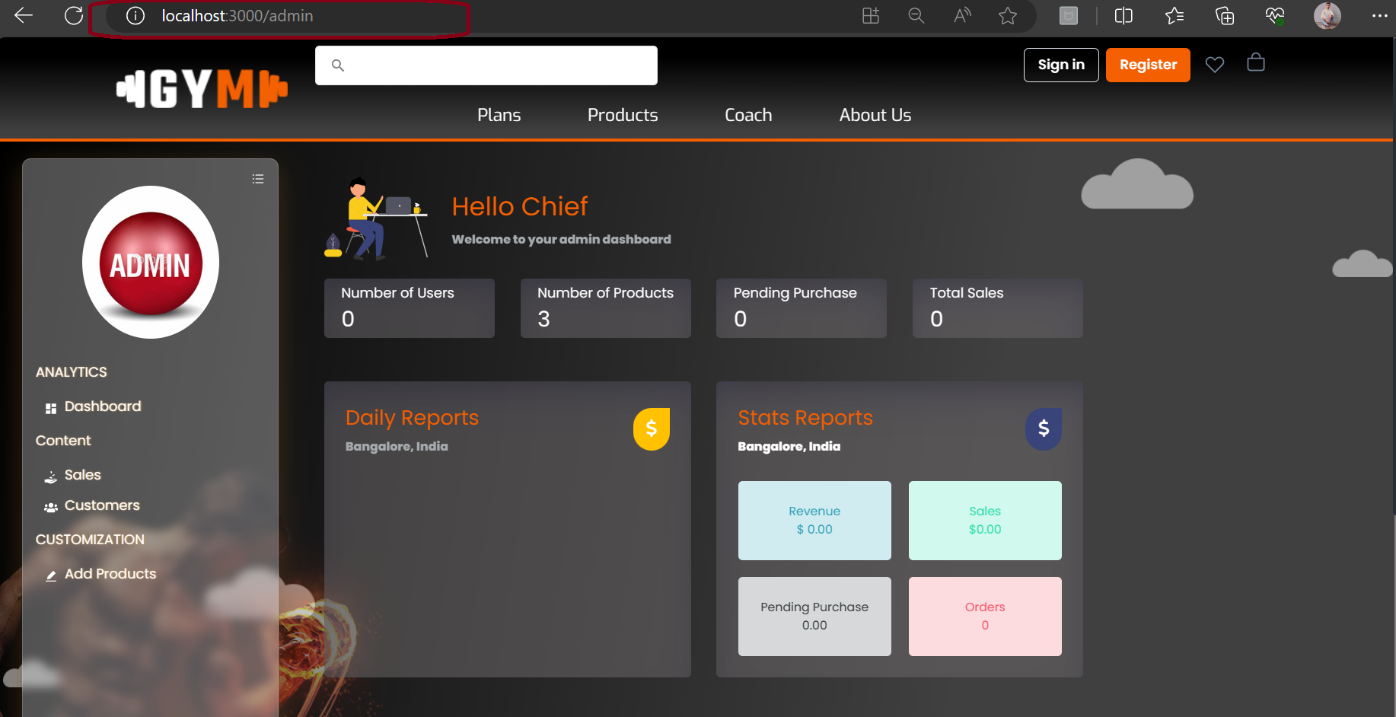


# SCREEN LAYOUT

Admin Page Screen Layout

ADMIN LOGIN

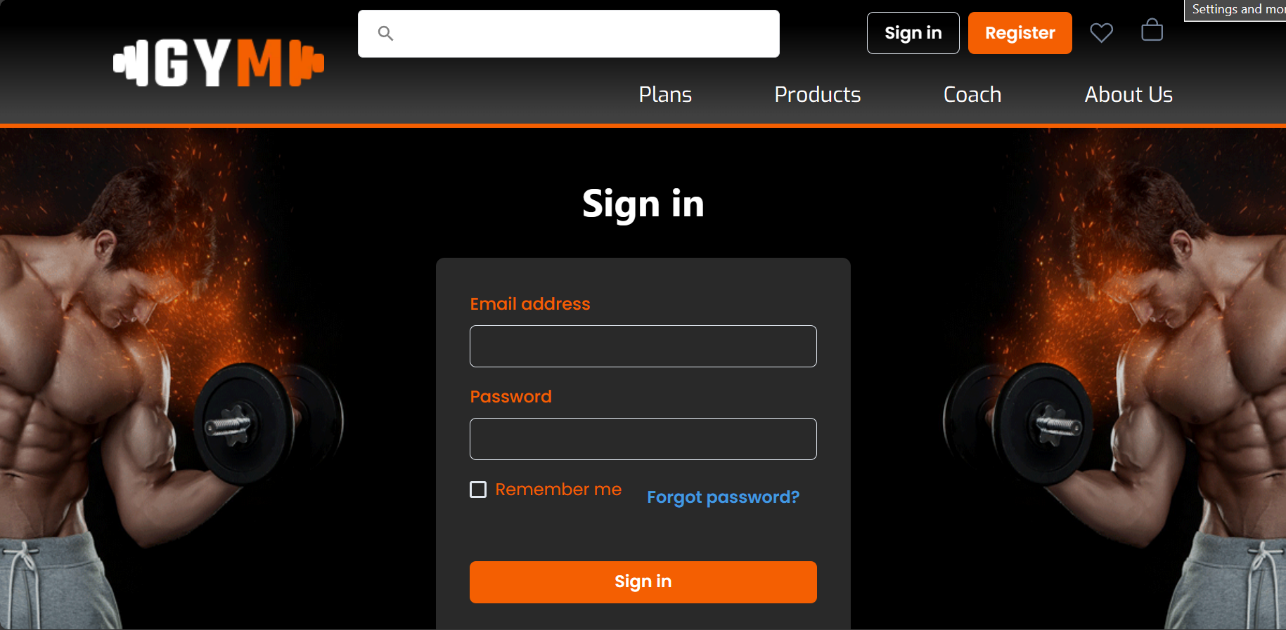




32



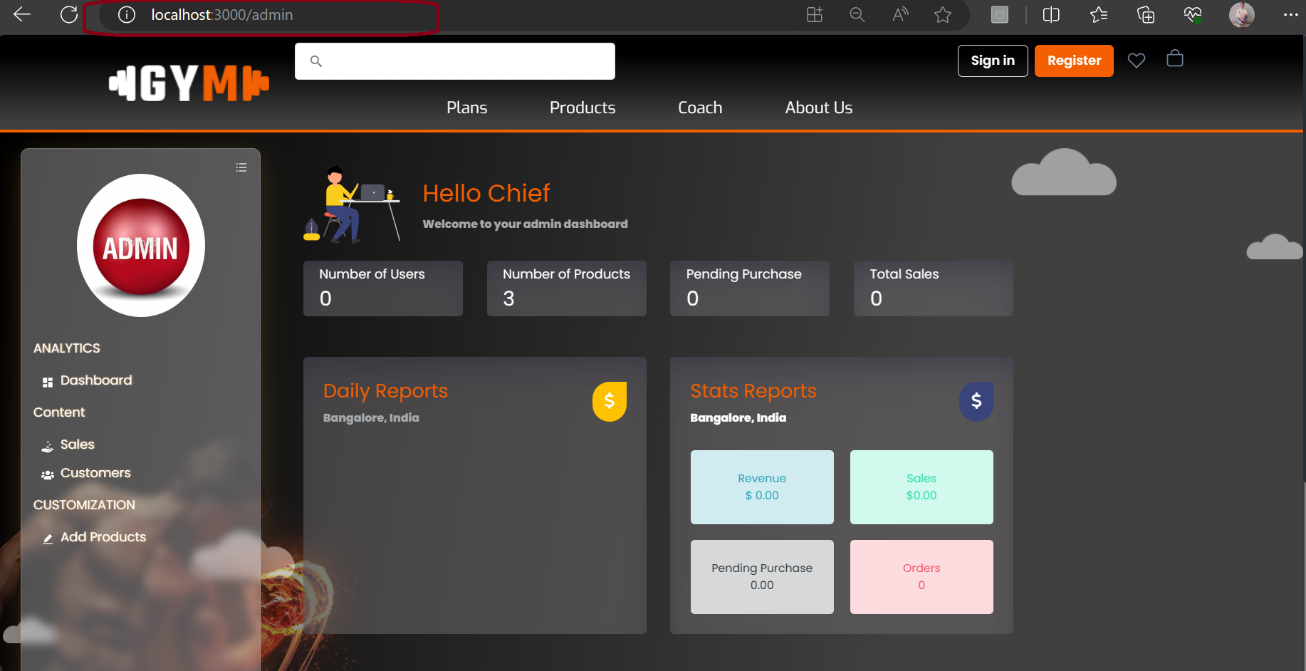
Login



33

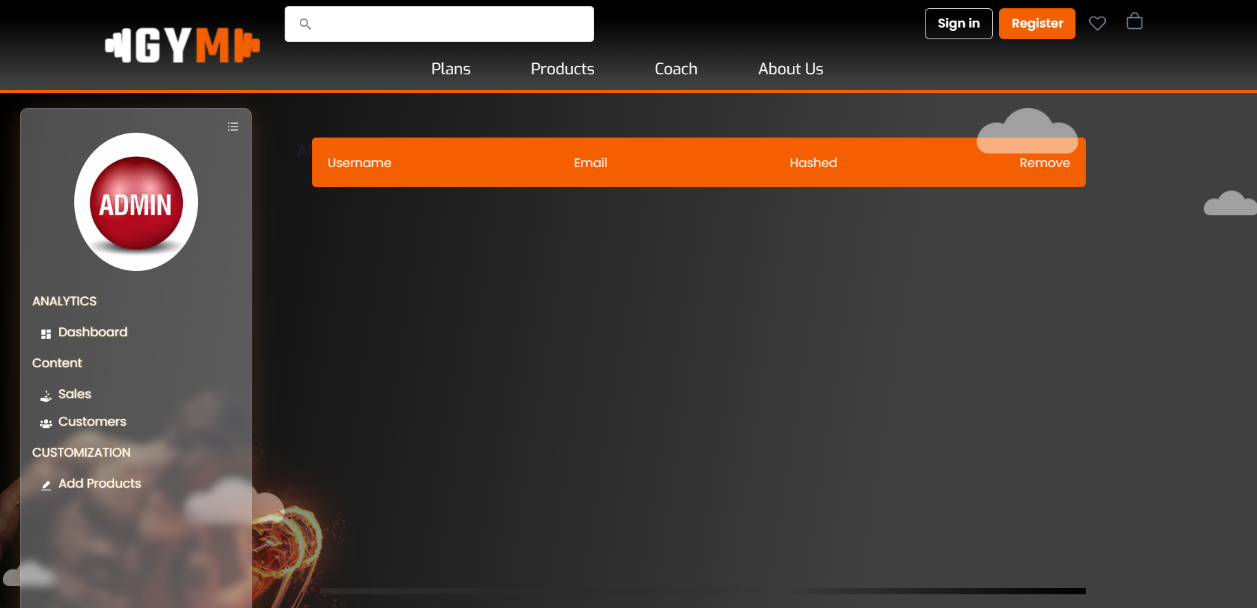


ADMIN DASHBOARD





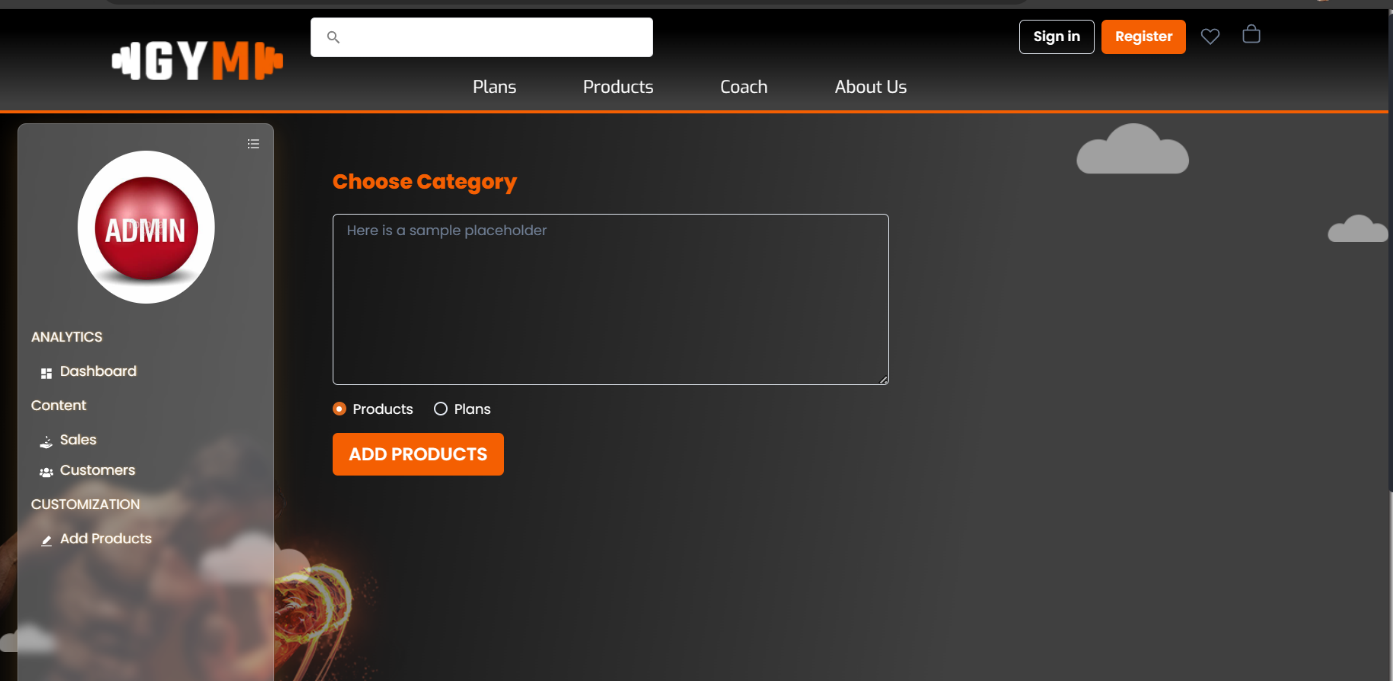
CUSTOMERS



34



ADD PRODUCT



STATUS REPORT

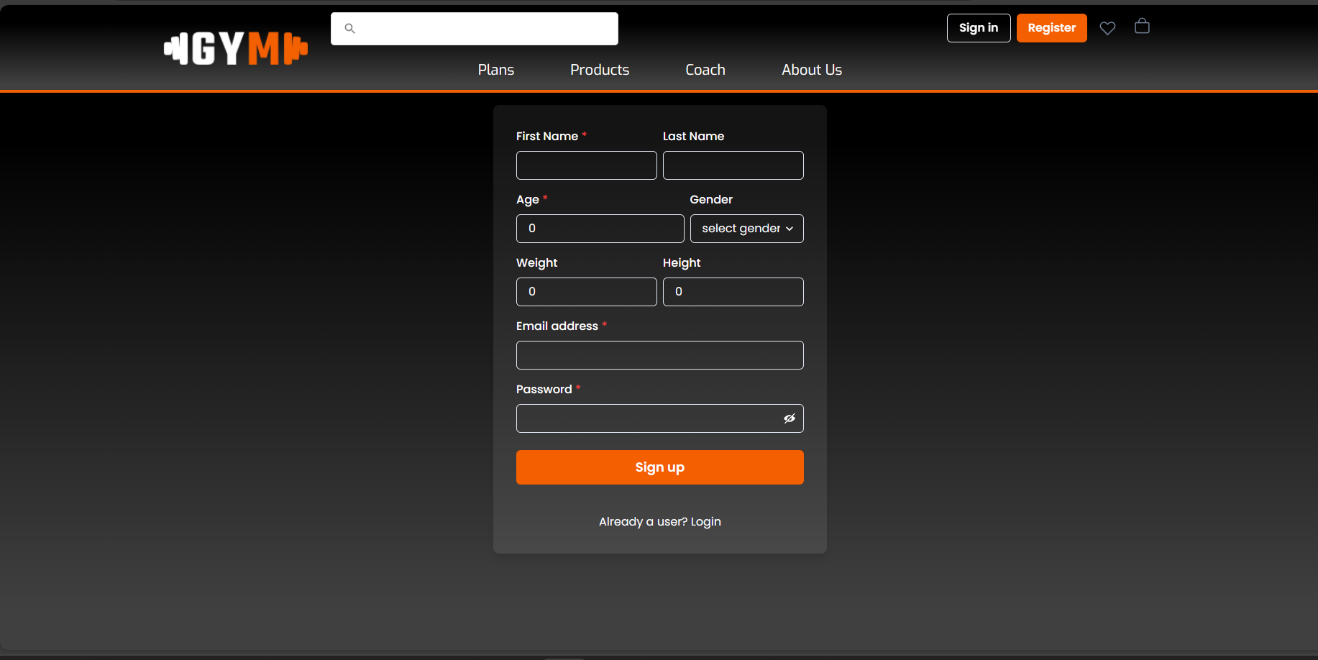
43

Client Page Screen Layout

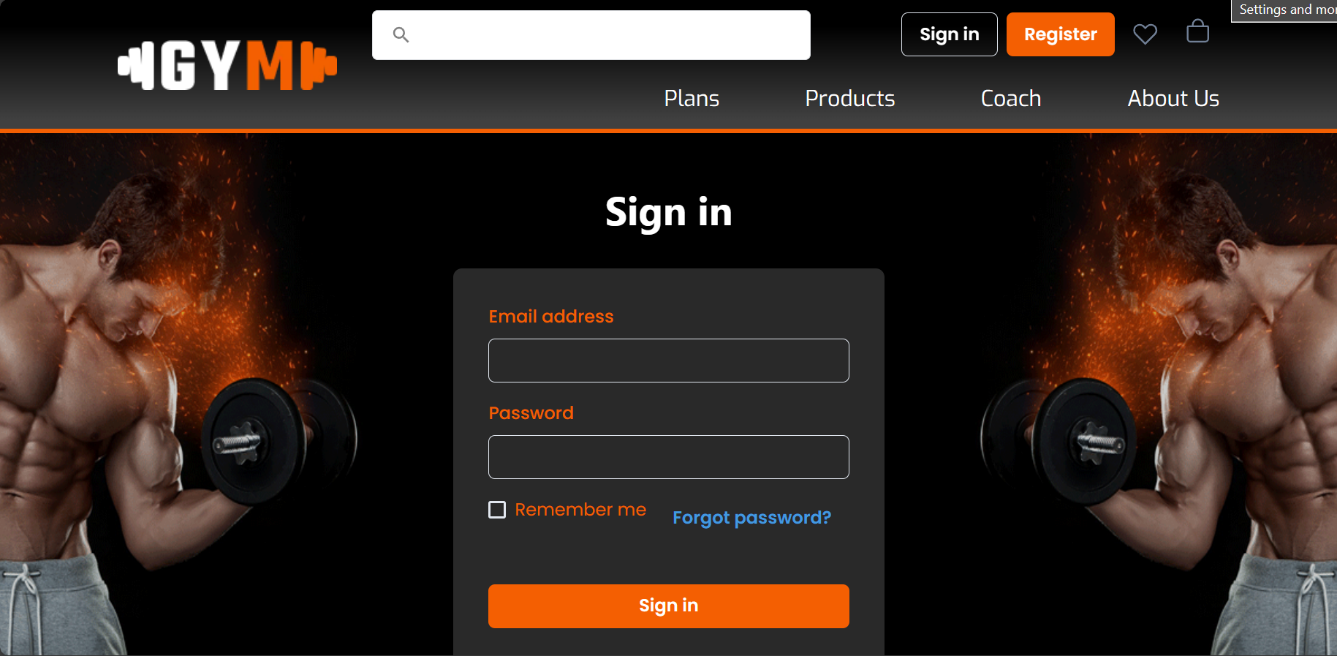


SIGNUP CLIENT



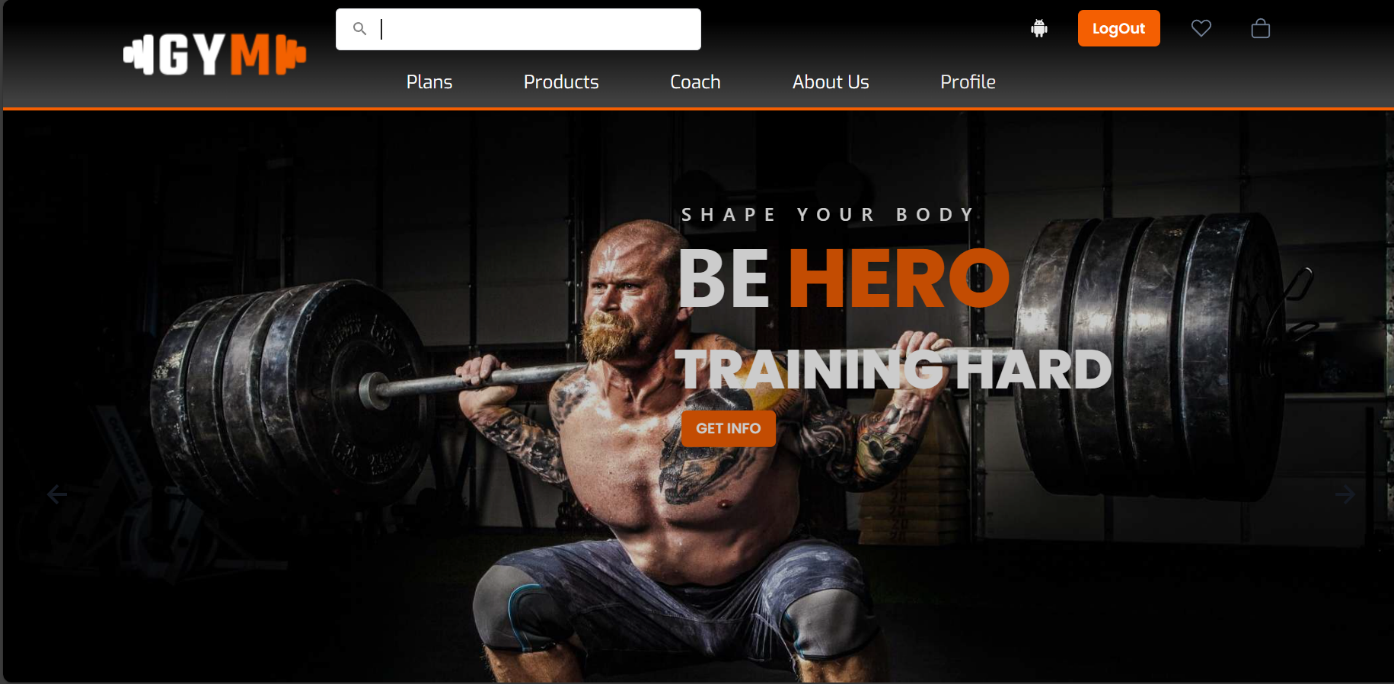


43

SINGIN CLIENT

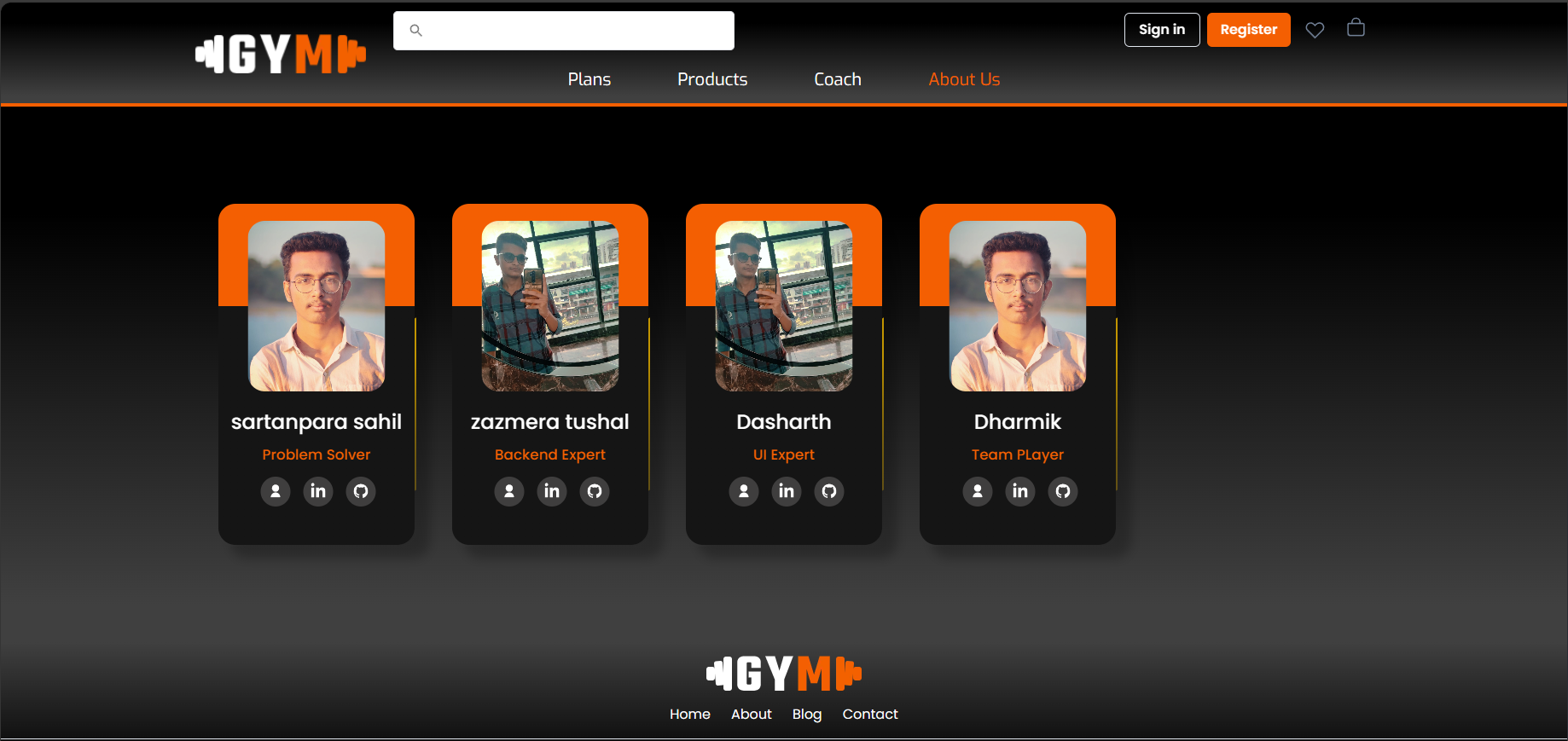


SING IN SUCCESSFULLY



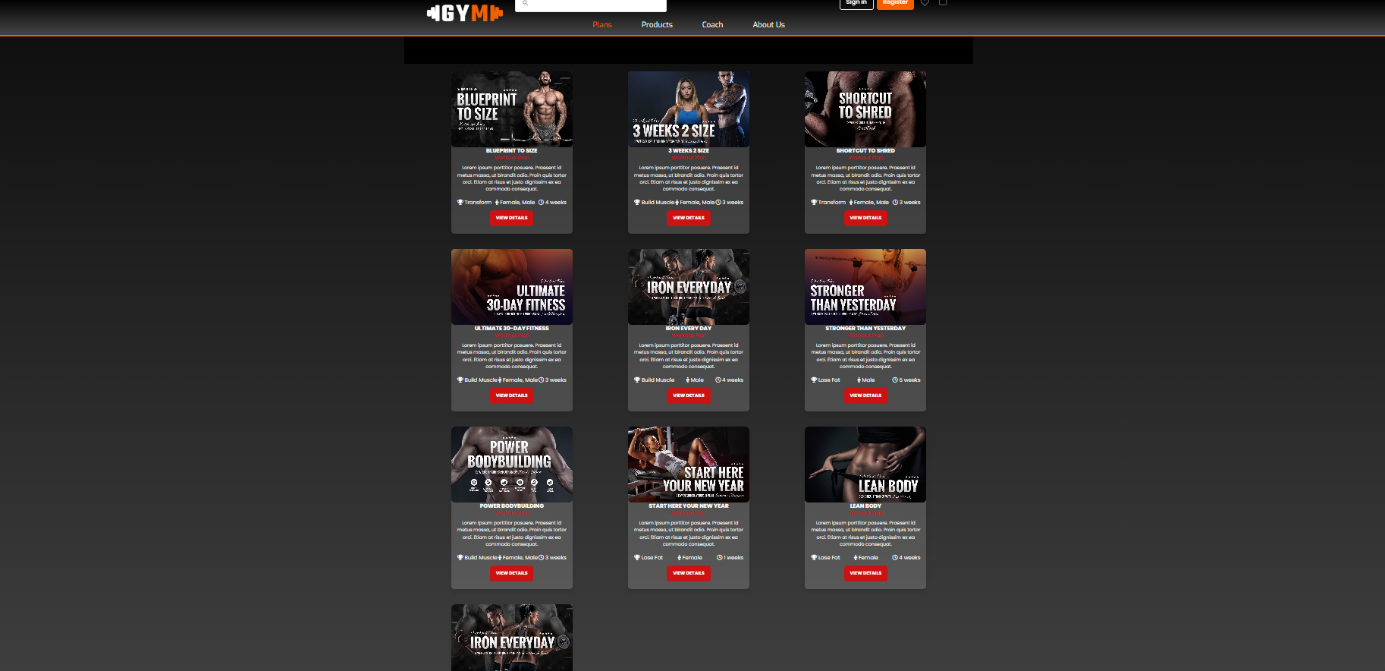
44

ABOUT US

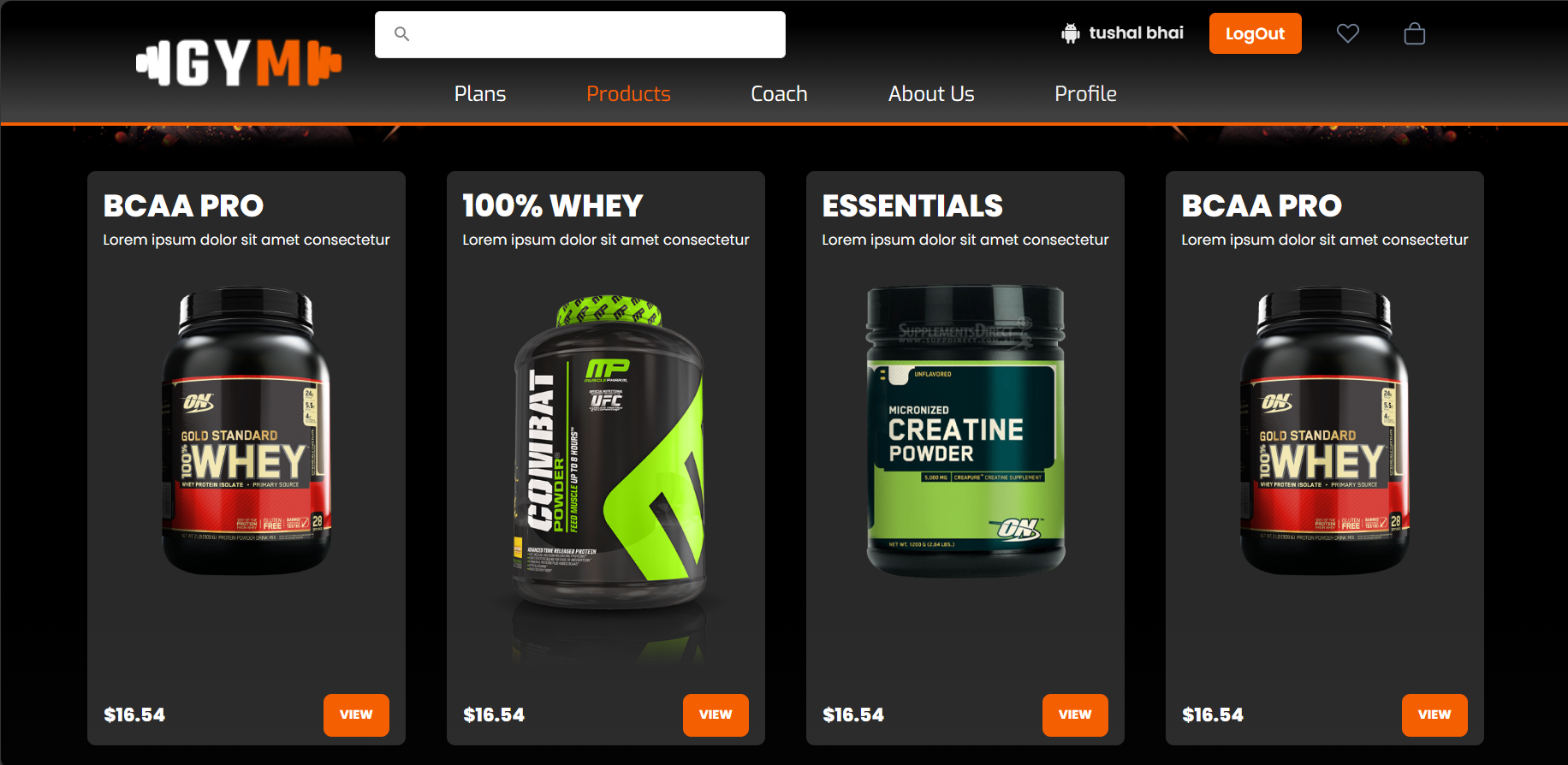




OUR PLAN



OUR PRODUCT



45



FOOTER



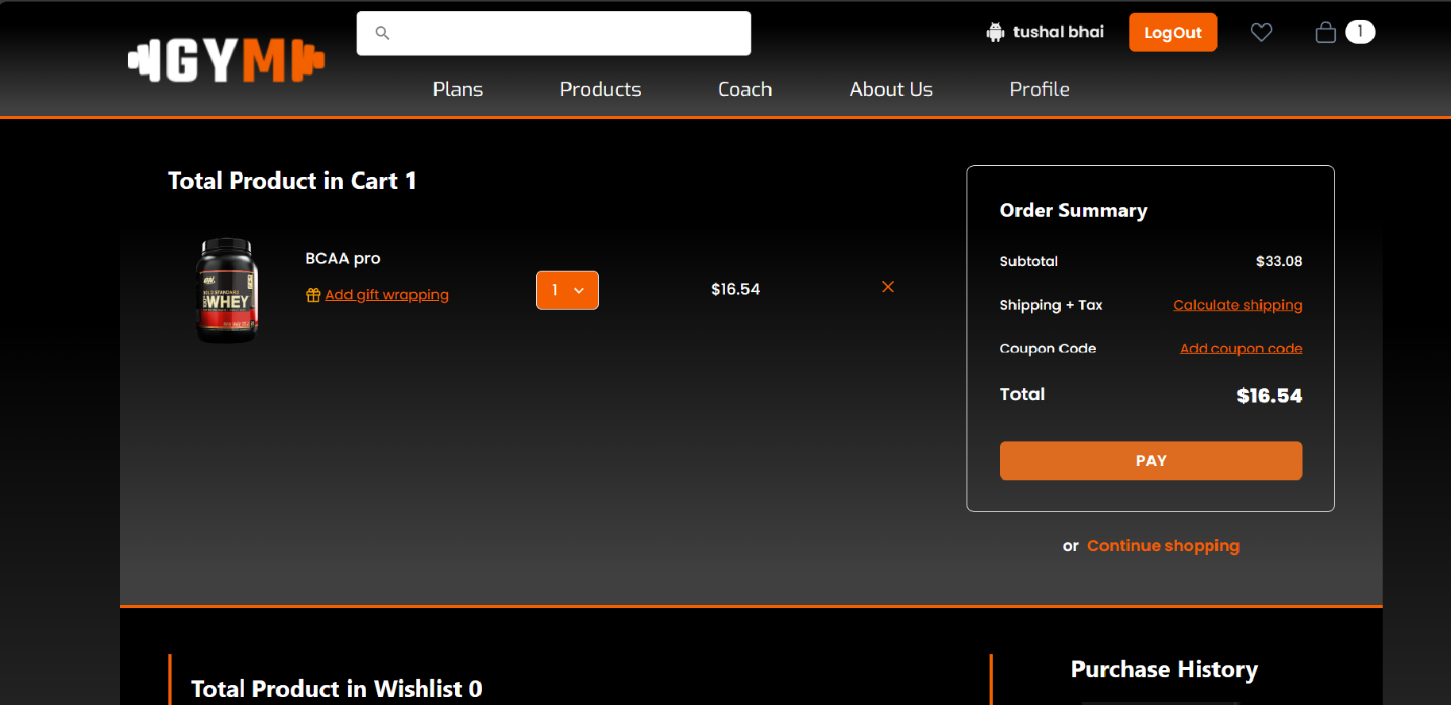
PRODUCT ADD TO CARG

46





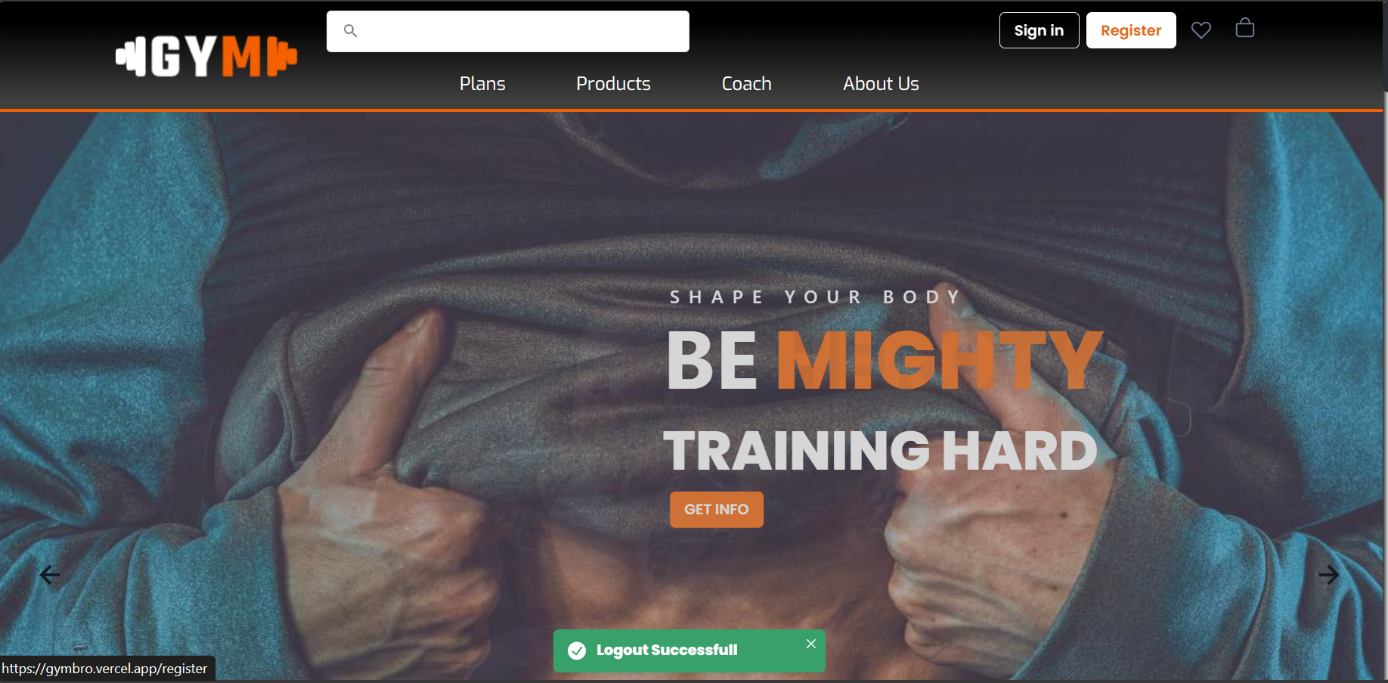
CART





FOOTER

LOGOUT



47