GYM FIT FOR MIALE GYM MEMBERSHIP SYSTEM

MWAKAI ALVIN MWAMBALA

A research proposal submitted to the Department of Information and Technology in partial fulfillment to the requirement of the award of Diploma in Information and Technology at Jomo Kenyatta University of Agriculture and Technology

Table of Contents

[DECLARATION 3](#_Toc143521900)

[ACKNOWLEDGEMENT 4](#_Toc143521901)

[DEDICATION 5](#_Toc143521902)

[CHAPTER 1 6](#_Toc143521903)

[INTRODUCTION 6](#_Toc143521904)

[1.0 PREAMBLE 6](#_Toc143521905)

[1.1 PROBLEM DESCRIPTION 6](#_Toc143521906)

[1.1.1 BACKGROUND OF THE PROBLEM 6](#_Toc143521907)

[1.1.2 PROBLEM STATEMENT 6](#_Toc143521908)

[1.2 PROJECT OBJECTIVE 7](#_Toc143521909)

[1.2.1 GENERAL OBJECTIVES 7](#_Toc143521910)

[1.2.2 SPECIFIC OBJECTIVES 7](#_Toc143521911)

[1.3 PROJECT SCOPE 8](#_Toc143521912)

[1.3.1 GEOGRAPHICAL SCOPE 8](#_Toc143521913)

[**1.1.1** **Target User** 8](#_Toc143521914)

[**1.1.2** **Specific Platform** 9](#_Toc143521915)

[**1.2** **CONSTRAINTS** 9](#_Toc143521916)

[1.5 PROJECT STAGES 11](#_Toc143521917)

[1.6 SIGNIFICANCE OF THE PROJECT 12](#_Toc143521918)

[1.7 SUMMARY 14](#_Toc143521919)

[CHAPTER TWO 15](#_Toc143521920)

[LITERATURE REVIEW 15](#_Toc143521921)

[2.1 SYSTEM REVIEW 15](#_Toc143521922)

[2.1.1 EXISTING PROGRAMS 15](#_Toc143521923)

[2.2 SYSTEM ADAPTATION 16](#_Toc143521924)

[CHAPTER 3 17](#_Toc143521925)

[ANALYSIS AND DESIGN 17](#_Toc143521926)

[3.1 DEVELOPMENT APPROACH 17](#_Toc143521927)

[3.2 REQUIREMENT SPECIFICATION 17](#_Toc143521928)

[3.3 LOGICAL DESIGN 18](#_Toc143521929)

[**3.3.1 SEQUENCE DIAGRAM** 18](#_Toc143521930)

[**3.3.2 USE CASE DIAGRAM** 18](#_Toc143521931)

[**3.4 DATABASE DESIGN** 19](#_Toc143521932)

[3.5 INTERFACE DESIGN 20](#_Toc143521933)

[3.6 TIME SCHEDULE 24](#_Toc143521934)

[REFERENCES 25](#_Toc143521935)

[CHAPTER 4 26](#_Toc143521936)

[SYSTEM IMPLIMENTATION 26](#_Toc143521937)

[4.0 INTRODUCTION 26](#_Toc143521938)

[4.1 SYSTEM INTEGRATION 26](#_Toc143521939)

[4.2 SYSTEM OUTPUT 26](#_Toc143521940)

[**4.2.2 User** 29](#_Toc143521941)

[**4.3 SYSTEM TESTING** 30](#_Toc143521942)

[**4.3.1 Test plan** 30](#_Toc143521943)

[**4.3.2 ENHANCEMENT** 31](#_Toc143521944)

[CHAPTER 5 32](#_Toc143521945)

[**CONCLUSION** 32](#_Toc143521946)

[**5.1 SYSTEM CONSTRAINT** 32](#_Toc143521947)

[**5.2 FUTURE ENHANCMENTS** 32](#_Toc143521948)

[**5.3 CONCLUSION** 33](#_Toc143521949)

[**APPENDIX** 34](#_Toc143521950)

[**SAMPLE CODES** 34](#_Toc143521951)

[**TECHNICAL GUIDE AND USER MANUAL** 41](#_Toc143521952)

[**USER MANUAL** 41](#_Toc143521953)

[**TECHNICAL GUIDE** 42](#_Toc143521954)

## DECLARATION

I declare that this report is my original work and has not been presented Kenya National examination Council or any examination body for an award of degree, diploma and certificate.

Name: MWAKAI ALVIN MWAMBALA.

Signature……………………………………………….

Date……………………………………………………….

Supervisor

Mrs. Cecilia Okumu

Signature……………………………………………………

Date………………………………………………………...

## ACKNOWLEDGEMENT

I acknowledge the immense contribution of Mrs. Cecilia Okumu for her support which enabled me to write this report. May God bless her abundantly.

## DEDICATION

I dedicate this report to my family for the support they have given me in the entire academic journey. May God bless them abundantly.

# **CHAPTER 1**

# **INTRODUCTION**

# **1.0 PREAMBLE**

Physical fitness is to the human body what fine tuning is to an engine. Regular exercise and physical activity promote strong muscles and bones. It improves respiratory, cardiovascular health, and overall health. The miale gym membership system is a software-based solution that enables gym owners and staff to manage their gym operations effectively. It will automate the process of gym membership registration, payments, attendance tracking and performance tracking.

# **1.1 PROBLEM DESCRIPTION**

# **1.1.1 BACKGROUND OF THE PROBLEM**

The background of the gym membership system can be tracked back to the early days of gymnasiums and fitness clubs where paper records were used to keep track of member information and payments.

One common problem with gym membership system is managing the sheer amount of data that needs to be processed and stored, including member information, payment details, and attendance records. As the gyms grows and attract more members, it becomes increasingly difficult to keep track of all this data manually which can lead to errors and inconsistencies in billing scheduling.

Another issue is ensuring the security of sensitive member data. With the rise of cyber threats, there is a risk that personal and financial information stored in gym membership systems could be compromised, putting members at risk of identity theft or fraud.

Finally, gym membership systems need to be adaptable to changing business needs and member preferences. As new fitness trends emerge and members demand more personalized experience, gym owners and managers need to be able to modify their systems to meet these evolving needs.

# **1.1.2 PROBLEM STATEMENT**

In the fitness industry, gym owners strive to create a secure and streamlined environment where gym-goers can confidently pursue their fitness goals. However, conventional gym management practices often fall short in meeting the expectations of both gym owners and members. Gym owners face challenges in efficiently managing member registrations and accessing vital information, as they heavily rely on manual record-keeping methods.

For gym members, the process of making payments and booking workout sessions can be tedious and time-consuming. These inefficiencies not only hinder the overall user experience but also lead to a lack of transparency in financial transactions, posing concerns for both parties.

To bridge these gaps and enhance the gym experience for all stakeholders, there is a pressing need for an integrated gym management system. The envisioned system aims to empower gym owners by providing them with a user-friendly dashboard that allows them to view and manage member registrations effortlessly. By digitizing member data, the system ensures data security and simplifies access to crucial information, leading to more effective decision-making and improved gym administration.

Simultaneously, the proposed gym management system caters to the needs of gym members, offering them a convenient platform to make secure payments and book workout sessions with their preferred trainers. By introducing an intuitive user interface and secure payment gateway, gym members can confidently engage in financial transactions and effortlessly manage their workout schedules.

The success of the gym management system lies in its ability to establish trust and credibility between gym owners and members. Data security, transparency in financial transactions, and streamlined administrative processes will be the pillars on which this system stands. Through this comprehensive solution, both gym owners and members will benefit from a seamless and satisfying gym experience, fostering a thriving fitness community.

In conclusion, the development of an integrated gym management system presents a transformative opportunity for gym owners to enhance their operational efficiency and optimize member services. By empowering gym members with a user-friendly platform for payments and bookings, the proposed system creates a harmonious ecosystem that fosters trust, convenience, and excellence within the fitness industry

# **1.2 PROJECT OBJECTIVE**

# **1.2.1 GENERAL OBJECTIVES**

The gym membership system tends to automate the process of registering gym members and showing members monthly or annually subscriptions. This technology will boost efficiency in work and potential gym members. The parties involved are required to have an internet connection since the system will be web based.

# **1.2.2 SPECIFIC OBJECTIVES**

To simplify the Application Process

To automated booking sessions with gym trainers

To enhance transparency in the management of gym funds

To enhance Data Management

# **1.3 PROJECT SCOPE**

The gym membership system tends to automate the process of registering gym members and showing members monthly or annually subscriptions. This technology will boost efficiency in work and potential gym members. The parties involved are required to have an internet connection since the system will be web based.

# **1.3.1 GEOGRAPHICAL SCOPE**

The geographical scope of the gym management system will primarily target the vibrant area known as Pipeline, serving as the hub for fitness enthusiasts seeking a top-notch gym experience. Situated within this dynamic locality, the gym aims to cater to the diverse fitness needs of the community, fostering a sense of well-being and health consciousness among its residents. The gym management system will be tailored to the specific requirements and preferences of Pipeline's residents, ensuring a seamless and localized experience. By concentrating efforts within this area, the system can forge strong connections with the community, effectively engage with members, and adapt to the unique fitness trends prevalent in the Pipeline vicinity. As the go-to fitness solution in Pipeline, the gym management system is set to enhance the fitness journey of its members and elevate the overall health and wellness standards within this bustling urban locality.

### **Target User**

**Gym users:** With the gym management system, gym members are presented with a seamless and user-friendly experience that enhances their fitness journey. The system's intuitive interface enables members to easily sign up and register, eliminating the need for cumbersome paperwork. Empowered with a personalized profile, members can conveniently manage their workout schedules and booking sessions with their preferred trainers. The system's secure payment gateway ensures peace of mind during financial transactions, instilling confidence in members as they engage in secure payments for membership fees and fitness services. Gone are the days of waiting in long queues or dealing with the uncertainty of manual bookings; now, members have the freedom to access the gym's schedule, view trainer availability, and reserve workout sessions at their convenience. The gym management system revolutionizes the way members interact with their fitness facility, creating a dynamic and engaging environment that promotes consistency and commitment to a healthier lifestyle.

**Gym owners:** The gym owners' admin dashboard within the gym management system is a powerful and indispensable tool that puts complete control at their fingertips. With a clean and intuitive user interface, the dashboard becomes the central hub for managing all aspects of the gym's operations. Gym owners can access comprehensive insights into member registrations, financial data, and gym performance metrics, enabling them to make data-driven decisions that drive the gym's success.

**System Administrators:** System administrators will be responsible for managing the system and ensuring its functionality and security. They will also provide technical support and maintenance services to ensure the system runs smoothly.

### **Specific Platform**

The gym management system will be developed as a web-based platform, providing accessibility and convenience to both gym owners and members. Leveraging the robust capabilities of the Laravel framework, the system will offer a scalable and feature-rich solution tailored to the unique needs of the fitness industry. As a PHP-based framework, Laravel ensures the development of a secure, reliable, and efficient platform that adheres to modern web standards.

The web-based platform built using Laravel will enable gym owners to access their admin dashboard from any internet-connected device, providing real-time insights into gym operations. The seamless integration of Laravel's database management features will facilitate efficient member registration, ensuring the secure storage and retrieval of member information. With Laravel's expressive syntax and built-in tools, the development process will be streamlined, resulting in a user-friendly and intuitive user interface for both gym owners and members.

The web-based nature of the platform ensures that gym members can conveniently access the system from their preferred devices, such as laptops, tablets, or smartphones. By harnessing Laravel's powerful routing and templating capabilities, the platform will deliver a responsive design that adapts to various screen sizes, enhancing the overall user experience. Whether booking workout sessions, making payments, or accessing personalized fitness schedules, gym members can engage effortlessly with the system from anywhere within the Pipeline area.

## **CONSTRAINTS**

**Budget Limitations:** The project may face budget constraints that could limit the resources available for development, testing, and ongoing maintenance. Balancing the desired features and functionality with the available budget will be crucial.

**Time Constraints:** There may be a tight deadline for the project's completion, especially if the gym management system needs to be deployed by a specific date or to coincide with the gym's opening. Time constraints can affect the depth of features and the overall testing and refinement of the system.

**Technical Expertise:** The development team's technical expertise in Laravel or web development, in general, could be a constraint. Complex features or integrations might require additional skills or time to implement.

**Data Privacy and Compliance:** The gym management system will likely handle sensitive personal and financial data of both gym owners and members. Compliance with data protection regulations and ensuring data privacy will be critical, adding complexity to the development process.

**Integration with Existing Systems:** If the gym has already been using certain software or systems (e.g., accounting software, CRM), integrating the new gym management system with these existing systems could pose challenges.

**User Adoption and Training:** Successfully introducing a new system requires user adoption and proper training. Ensuring that both gym owners and members are comfortable using the system may require time and effort.

**Scalability and Performance:** As the number of gym members and data grows, the system must be able to handle increased traffic and maintain optimal performance. Planning for scalability and load testing will be important.

**Mobile Responsiveness:** Ensuring that the web-based platform is responsive and user-friendly across various devices and screen sizes might be a constraint.

**Infrastructure and Hosting:** Choosing the right hosting solution and ensuring reliable infrastructure to host the system may impact the project.

**User Experience:** Creating a user-friendly interface and an intuitive user experience is essential for user satisfaction. Design and usability constraints need to be addressed.

**Gym-specific Requirements:** Addressing gym-specific needs and workflows may add complexity to the system's design and development.

**Security Considerations:** Implementing robust security measures to protect against potential cyber threats and breaches will be a critical constraint.

# **1.5 PROJECT STAGES**

WATERFALL MODEL

**Requirements Gathering:** Work closely with gym owners and stakeholders to gather detailed and comprehensive requirements for the gym management system. Document all the functional and non-functional requirements.

**System Design:** Create a detailed system design based on the gathered requirements. This design phase involves architectural planning, database design, and user interface design.

**Implementation:** With the system design in hand, start implementing the gym management system. Develop the software components, integrate databases, and build the user interface according to the design specifications.

**Testing:** Conduct thorough testing of the gym management system to ensure that it meets the specified requirements and functions as intended. This phase includes unit testing, integration testing, and system testing.

**Documentation:** Generate comprehensive documentation for the gym management system, including user manuals, technical documentation, and system specifications.

**Deployment:** Deploy the fully developed and tested gym management system in the production environment.

**Training:** Train gym owners, staff, and members on how to use the gym management system effectively.

**User Acceptance Testing (UAT):** Involve gym owners and selected members in UAT to validate the system's functionality and usability. Address any issues identified during UAT.

**Final Review:** Conduct a final review of the gym management system to ensure that all requirements have been met and that the system is ready for operational use.

**Maintenance and Support:** Provide ongoing maintenance and support for the gym management system, addressing any issues or bugs that may arise post-implementation.

# **1.6 SIGNIFICANCE OF THE PROJECT**

The gym management system holds significant importance for both gym owners and gym members due to the following reasons.

1. Streamlined Operations: The system brings efficiency to the gym's day-to-day operations by automating tasks such as member registration, class scheduling, and payment processing. This streamlined approach allows gym owners and staff to focus more on providing excellent services and enhancing the overall gym experience.

2. Data Security and Privacy: By digitizing member information and transactions, the system ensures the security and privacy of sensitive data. Gym members can trust that their personal information and financial details are kept safe, instilling confidence in the gym's commitment to data protection.

3. Improved Financial Management: With a centralized system for tracking payments, expenses, and revenue, gym owners can gain better insights into the financial health of the business. This enables more informed decision-making and helps identify areas for cost optimization and revenue growth.

4. Enhanced Member Experience: Gym members benefit from a user-friendly platform that allows them to conveniently book workout sessions, view trainer availability, and make secure payments. The system's ease of use fosters member satisfaction and encourages long-term loyalty to the gym.

5. Resource Optimization: Through data analytics and reports provided by the system, gym owners can optimize resource allocation, ensuring that trainers, equipment, and classes are efficiently utilized, leading to better resource management.

6. Transparent Communication: The system facilitates seamless communication between gym owners, staff, and members through automated notifications and reminders. This improves member engagement and keeps everyone well-informed about class schedules, promotions, and updates.

7. Business Growth and Competitiveness: A well-implemented gym management system can set a gym apart from its competitors. The system's ability to deliver exceptional member experiences and efficient operations can attract new members and retain existing ones, contributing to overall business growth.

8. Adaptable and Scalable Solution: A well-designed gym management system can adapt to the changing needs of the gym and scale alongside its growth. This ensures that the system remains relevant and effective as the gym expands its services and membership base.

9. Long-term Sustainability: By optimizing operations and financial management, the system contributes to the long-term sustainability of the gym. Improved member satisfaction and retention rates translate into a stable and thriving fitness business.

10. Technological Advancement: Implementing a web-based gym management system built using Laravel showcases the gym's embrace of modern technology. This can attract tech-savvy members and position the gym as a forward-thinking fitness center in the community.

# **1.7 SUMMARY**

The gym management system is a project of significant importance that aims to streamline gym operations and enhance the gym experience for both gym owners and members. By providing a secure and user-friendly platform, the system ensures the privacy of member data, automates administrative tasks, and facilitates seamless payment processing and workout session bookings. Gym owners benefit from real-time insights into member registrations and financial data, enabling data-driven decision-making and optimal resource allocation. Additionally, the system fosters transparent communication with members, keeping them informed about class schedules and promotions. The web-based platform built using Laravel offers scalability and adaptability, accommodating the gym's growth and technological advancements. Overall, the gym management system becomes a key tool for achieving long-term sustainability, business growth, and member satisfaction in the dynamic fitness industry.

# **CHAPTER TWO**

# **LITERATURE REVIEW**

A gym management system is a technology-based solution designed to streamline and automate the process of registering and keeping gym members records. Is serves as a centralized platform that assists gym owners and staff to manage and keep track of the gym members status in the gym.

Gym management system typically offer a range if features and functionalities that aid in the various aspects of gym management, these include:

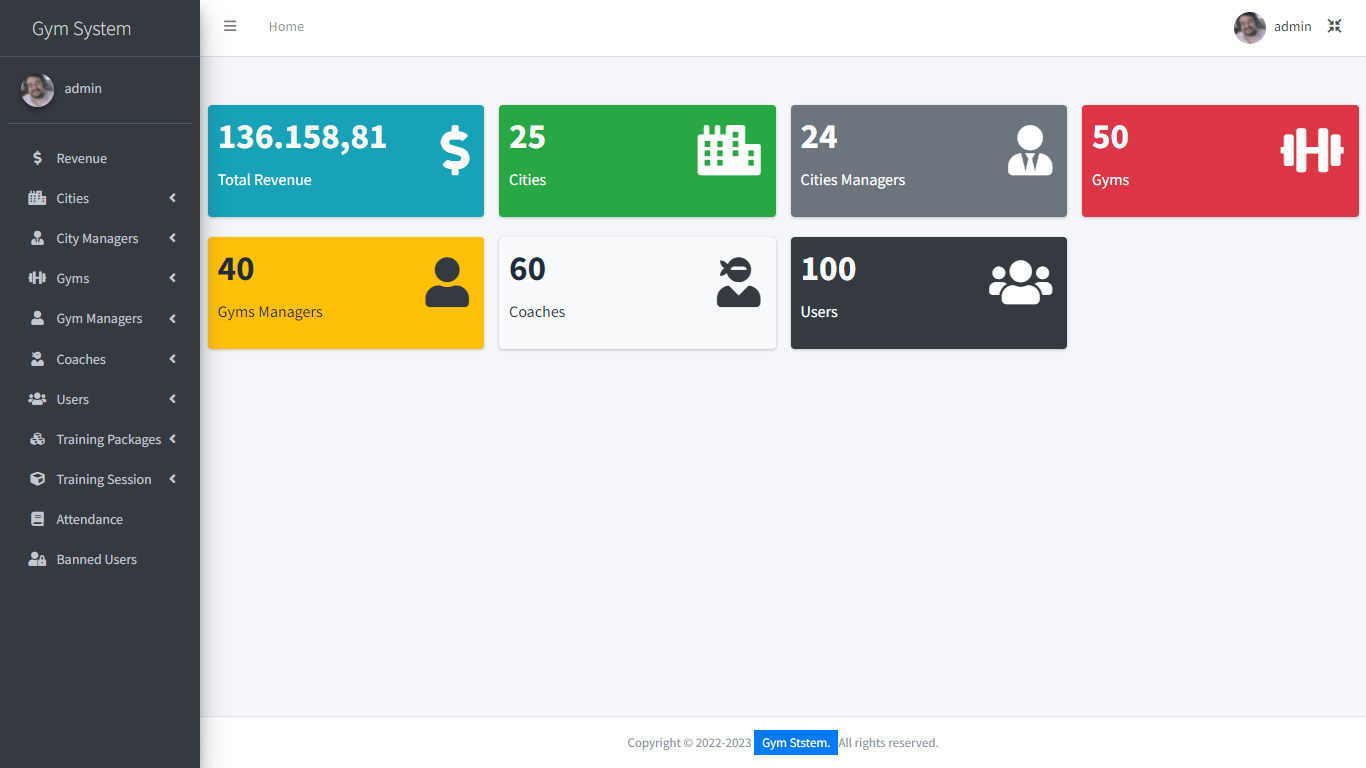
User registration to the gym: The system enables the creation of gym members accounts where they can choose their gym membership subscriptions and make the payment.

Admin registration: The system enables the creation of administrators of the system where they will be able to verify user details in the admin dashboard.

Sessions and payment tracking: The system tracks and manages the gym members gym sessions and also their payment statuses.

# **2.1 SYSTEM REVIEW**

# **EXISTING PROGRAMS**



**2.1.2 Advantages**

1. Includes a city manager who sees all the registered gyms in the city.
2. Project is can be used in a wide geographical area
3. Includes different currencies of payment
   * 1. **Disadvantages**
4. Can only be used in Northern America and Europe

# **2.2 SYSTEM ADAPTATION**

A functionality was added to allow members to be added to different gym schedules. This was achieved by creating a feature in the system that enabled gym administrators to create different workout schedules and allowed members to be assigned to specific schedules based on their fitness goals and preferences. Members were also allowed to switch between different schedules as needed.

A feature was implemented to enable assigning appropriate trainers to members. This was achieved by creating a feature in the system that enabled gym administrators to assign trainers to members based on their fitness goals, skill level, and workout preferences. Trainers were also allowed to create customized training programs for members and track their progress over time.

A feature was added to provide recommended diet plans for members. This was achieved by creating a feature in the system that enabled gym administrators or nutritionists to create recommended diet plans for members based on their fitness goals and dietary preferences. Members were also allowed to track their dietary intake and receive feedback on their progress towards their fitness goals.

# **CHAPTER 3**

# **ANALYSIS AND DESIGN**

In this chapter, system requirement of the gym management system is identified

System requirement of the gym management system are identified. These include user requirements such as admins being able to view a variety of clients data such as monthly

# **3.1 DEVELOPMENT APPROACH**

The gym membership system is developed using the modified waterfall methodology. The modified waterfall methodology provides a structured approach to develop application projects. Waterfall model is a software development life cycle approach that assumes the various phases of a project can be completed sequentially. It also ensures that the development process is efficiently managed. Due to its structured technique, the developers put equal emphasis on all modules to come up with a project that is of high standards to the consumer.

Advantages of waterfall model

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.

Disadvantages of waterfall model

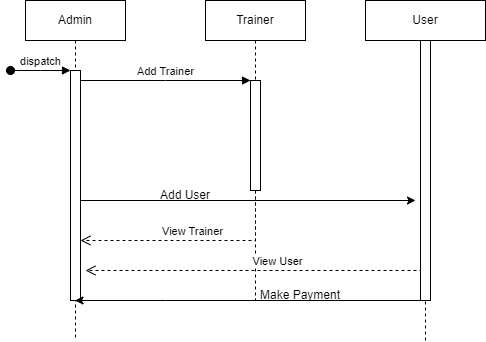
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.

# **3.2 REQUIREMENT SPECIFICATION**

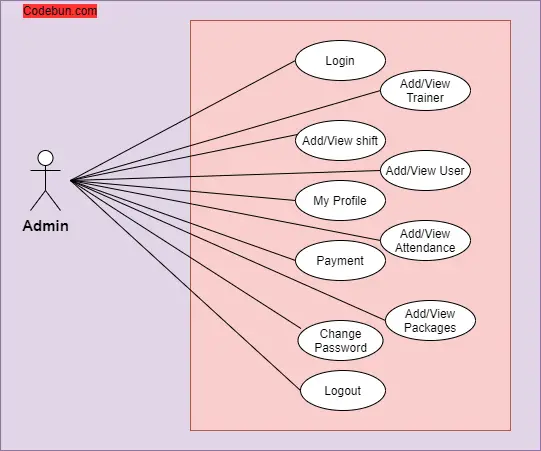
This section covers what the application does. It also dictates how the application responds to inputs from users and the capabilities and functions that are included in the system. This is a web-based application that requires the user to have a mobile phone or a computer that has access to the internet.

# **3.3 LOGICAL DESIGN**

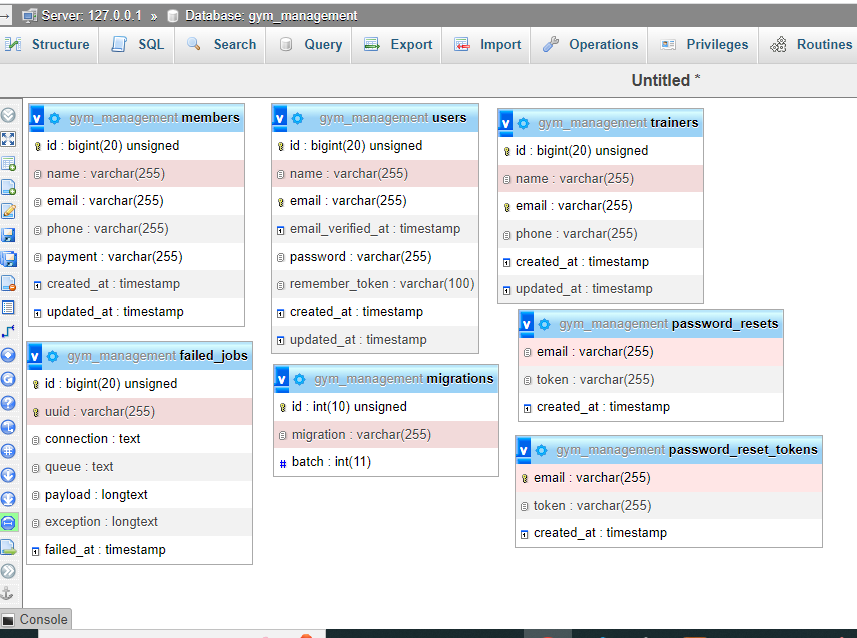
### **3.3.1 SEQUENCE DIAGRAM**



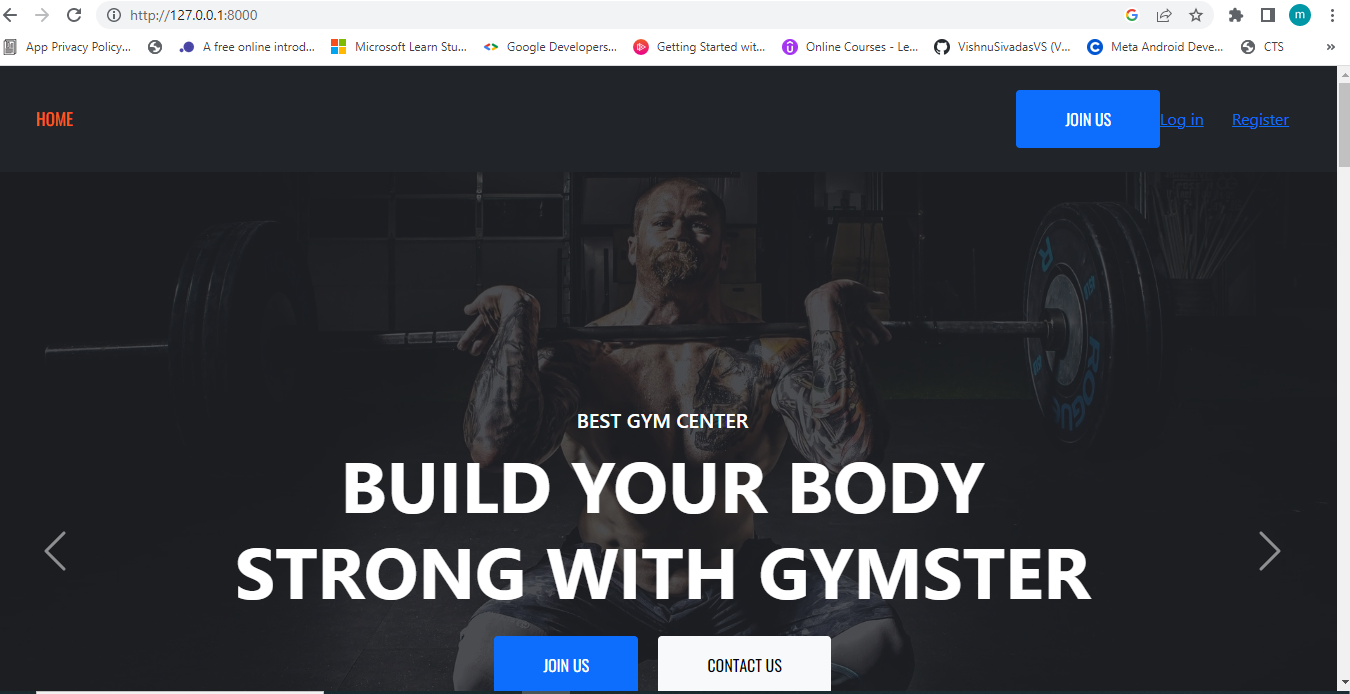
### **3.3.2 USE CASE DIAGRAM**



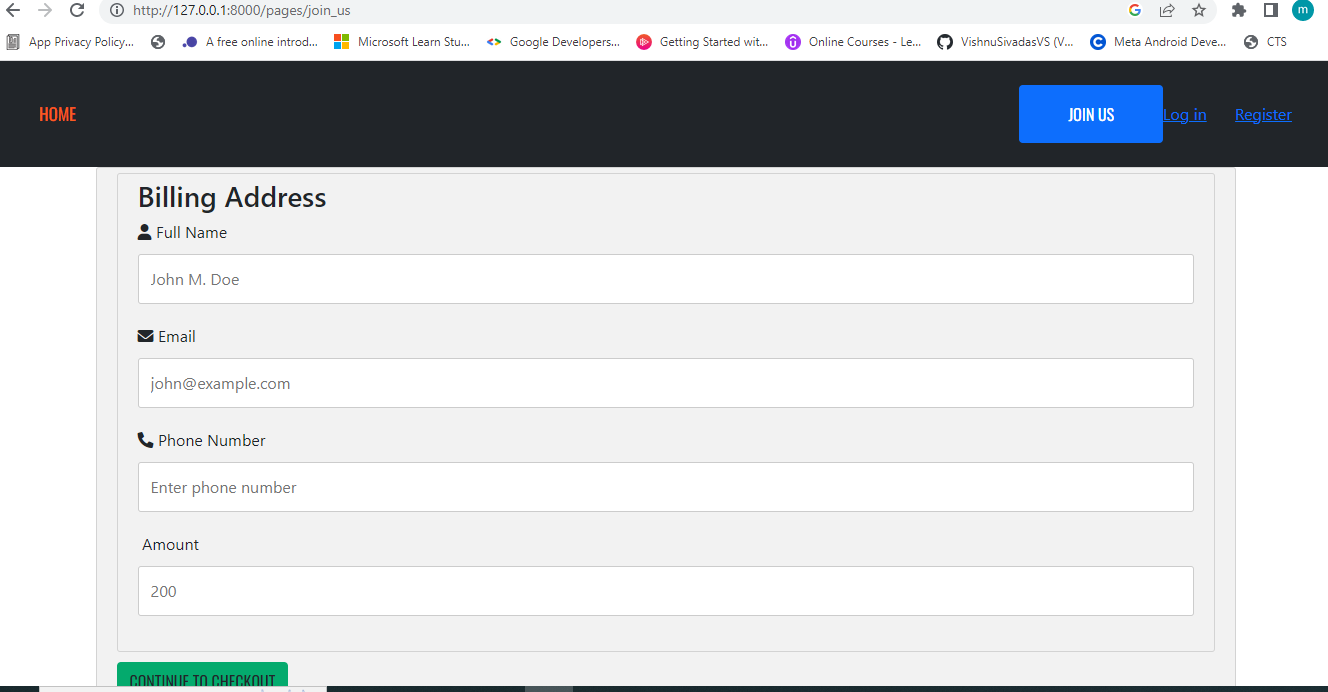
## **3.4 DATABASE DESIGN**



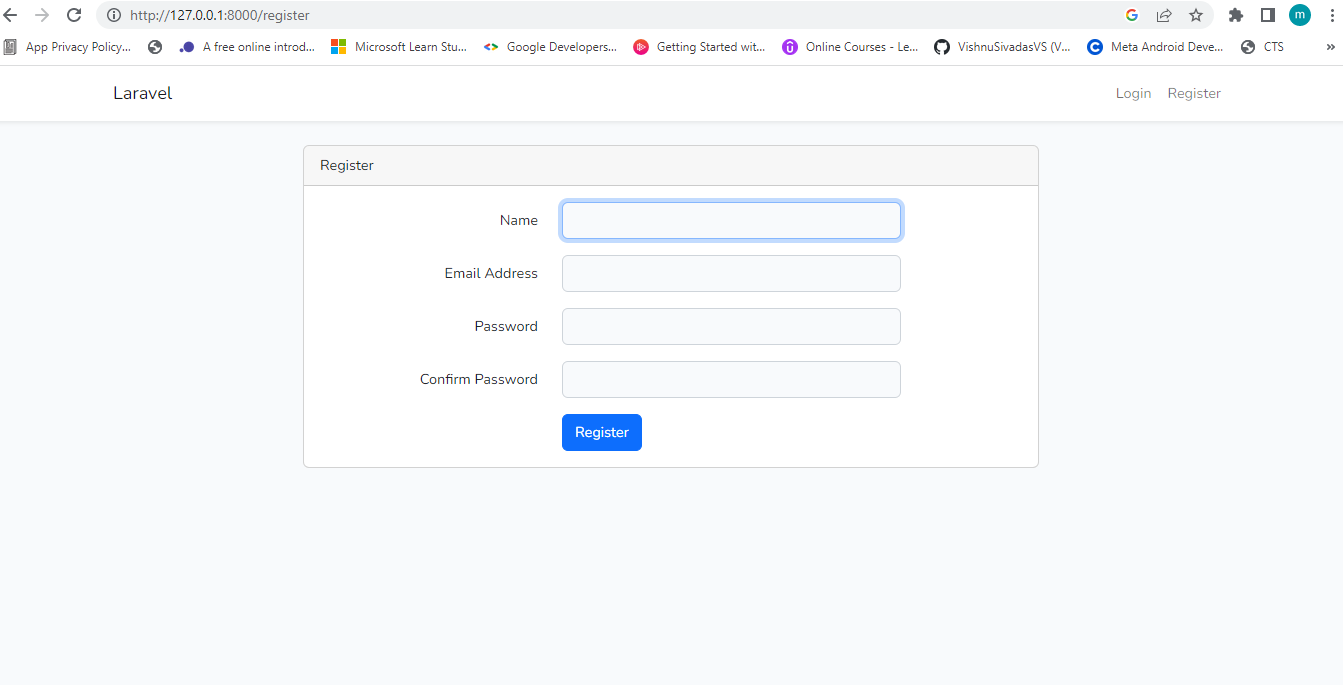
## 3.5 INTERFACE DESIGN

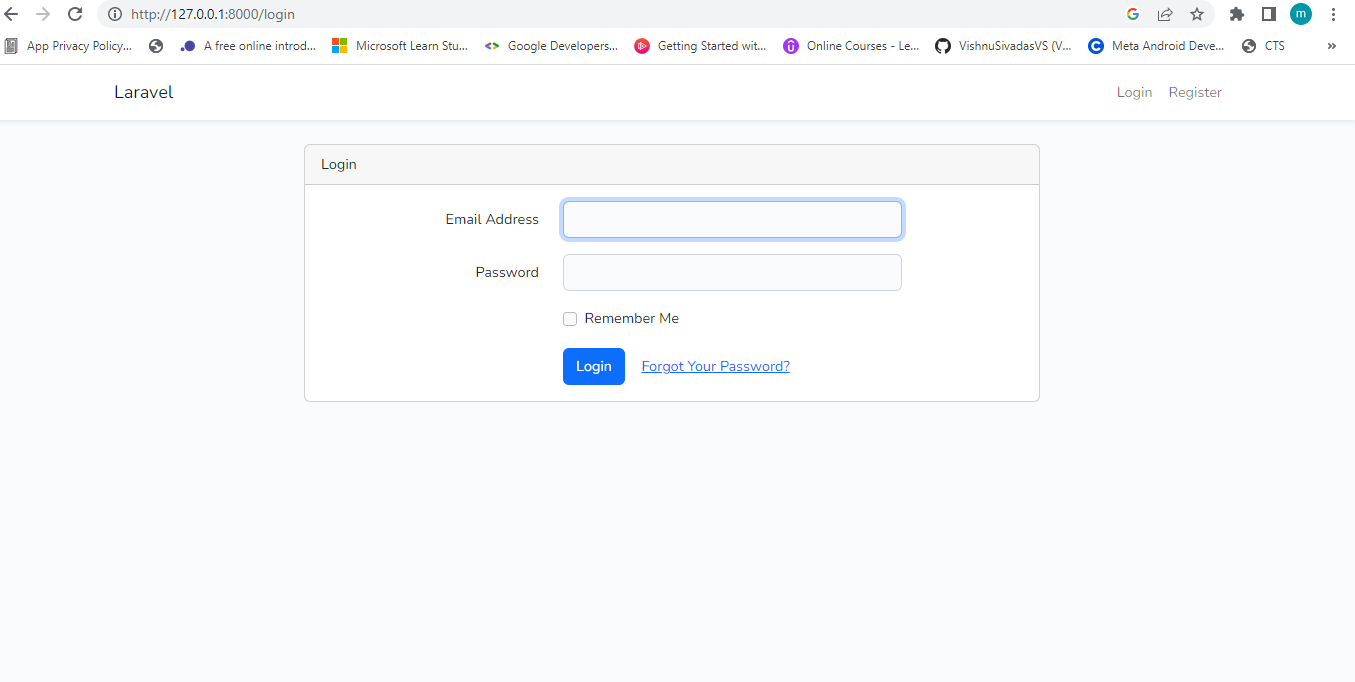
This is the users home page where they will be able to view the gyms membership. 

This is the users billing and registration form, this page is where a user adds their details and also makes a payment to the gym, the payment is through M-pesa, once the user inputs their details and clicks on “continue to checkout”, the M-pesa express API is initiated and asked to make a payment through their phone number.

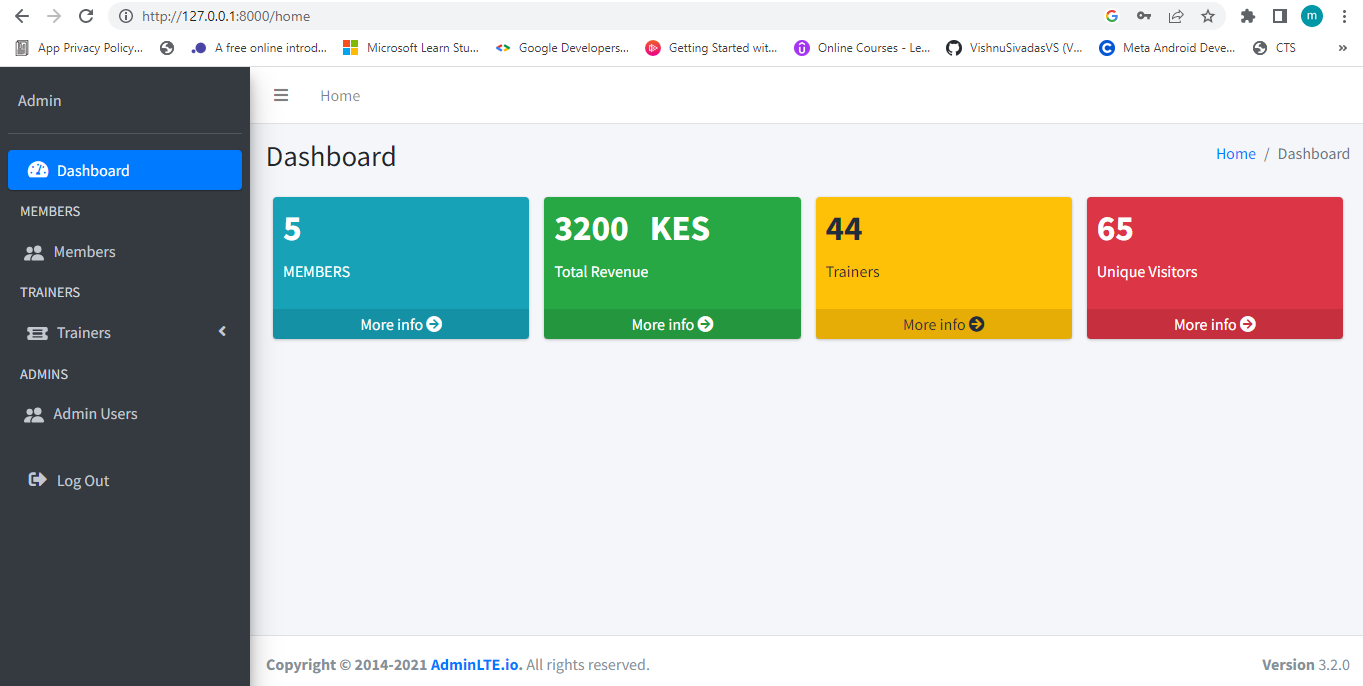


This are the admin pages for the gym membership system.

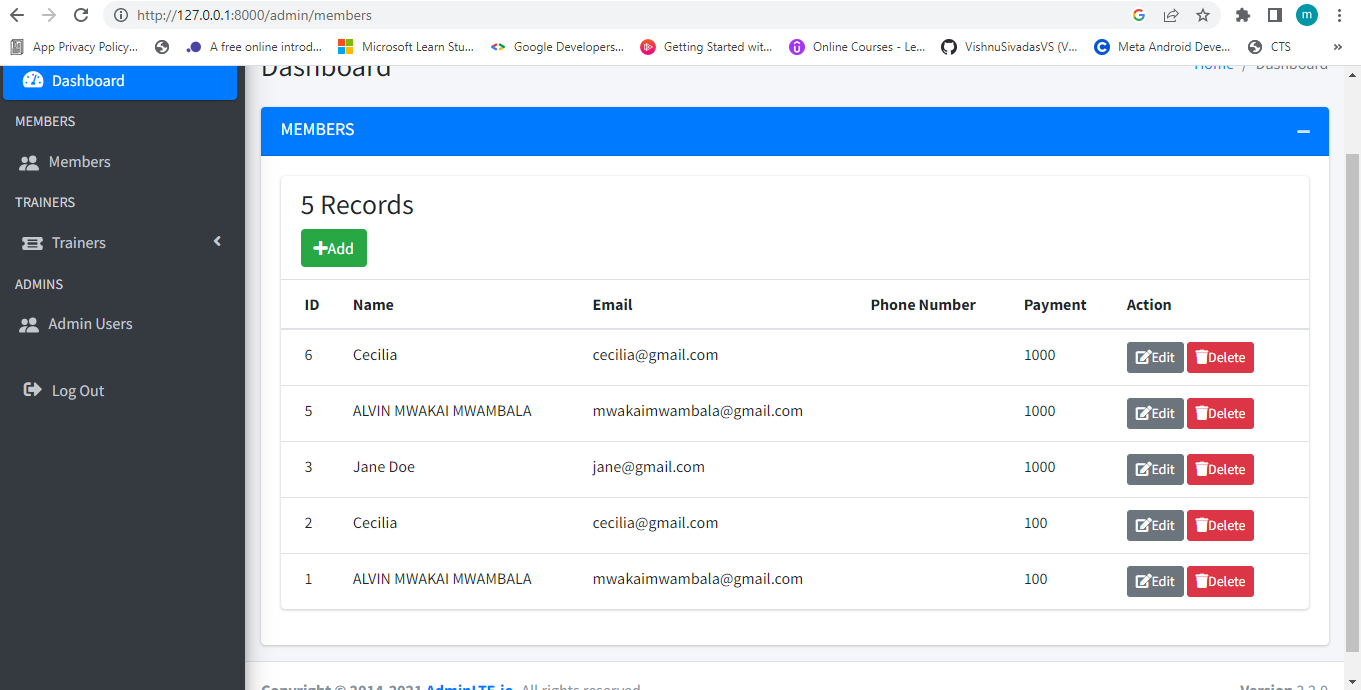


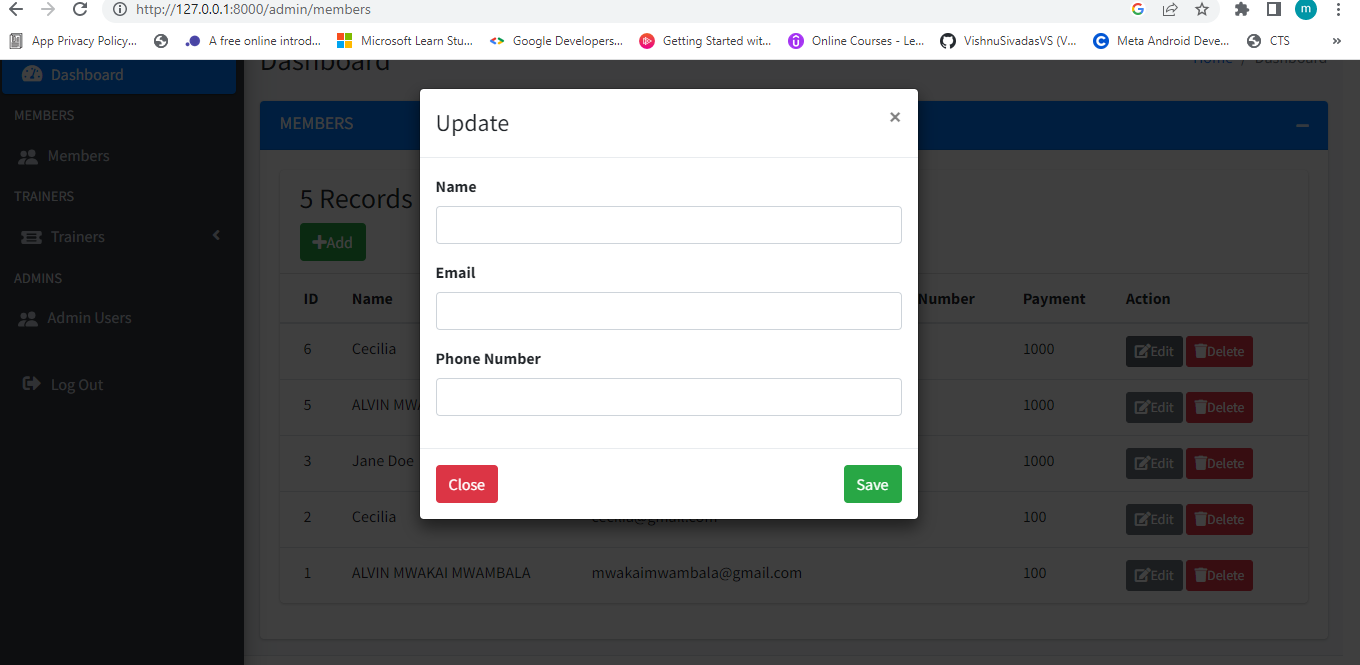


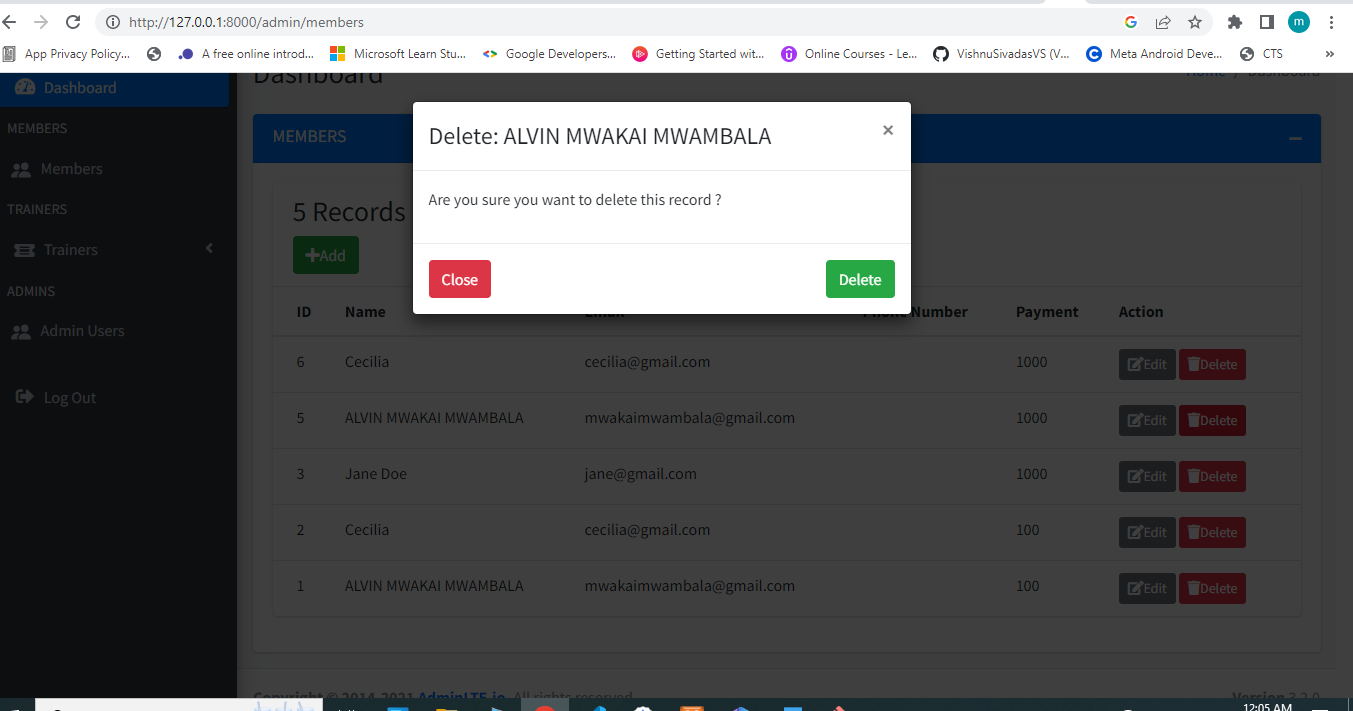
This is the admin dashboard.



This are the admin pages to check and manage members.







## 3.6 TIME SCHEDULE



# **REFERENCES**

Taylor Otwell**:** <https://laravel.com/docs/6.x/installation>

<https://github.com/MohamedAlabasy/Gym-laravel-Project-ITI>

<https://codebun.com/uml-diagram-for-gym-management-system/>

This is the GitHub URL for the project, it is open for contribution <https://github.com/Mwakai/gym-management>

# **CHAPTER 4**

# **SYSTEM IMPLIMENTATION**

## 4.0 INTRODUCTION

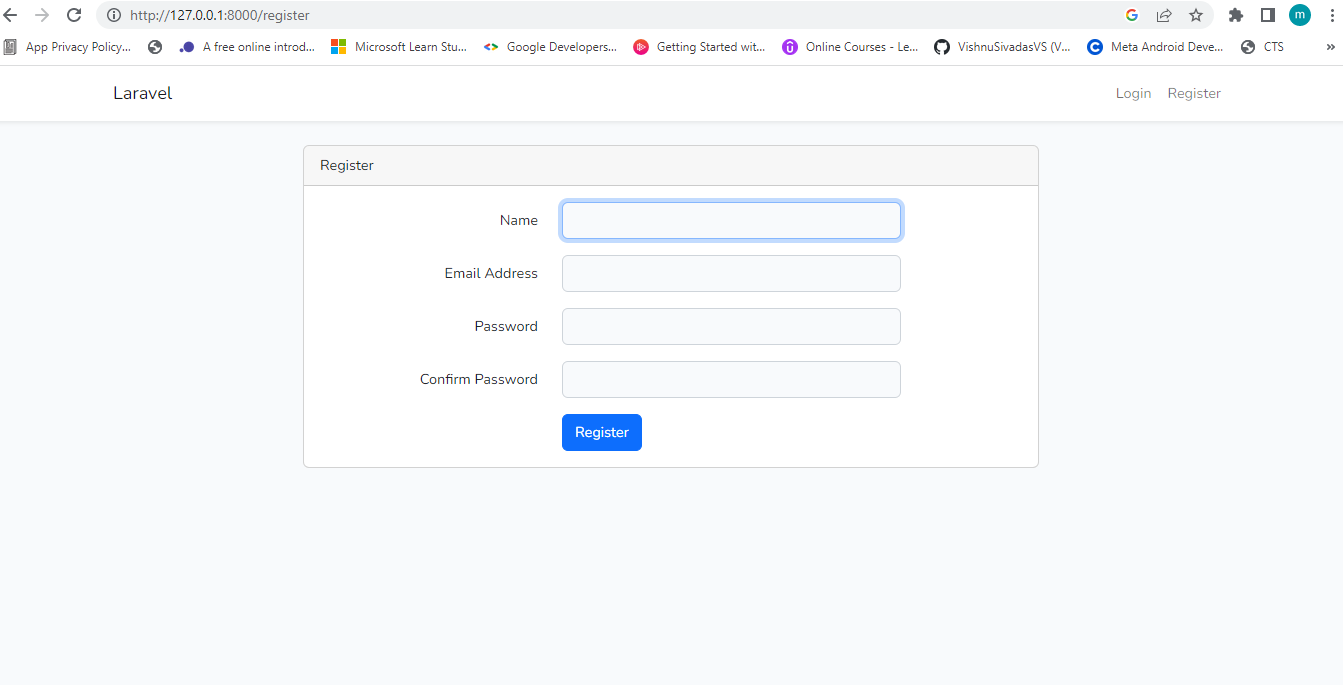
The final product of the project is full functional web-based gym management system that enables gym members to signup for gym classes and gym owners to manage the gym members and gym classes. The system incorporates a user-friendly interface, secure login and registration features and secure payments through M-pesa express. The system also includes a database that stores user information and various APIs and technologies that facilitates communication between the frontend and back-end components of the system

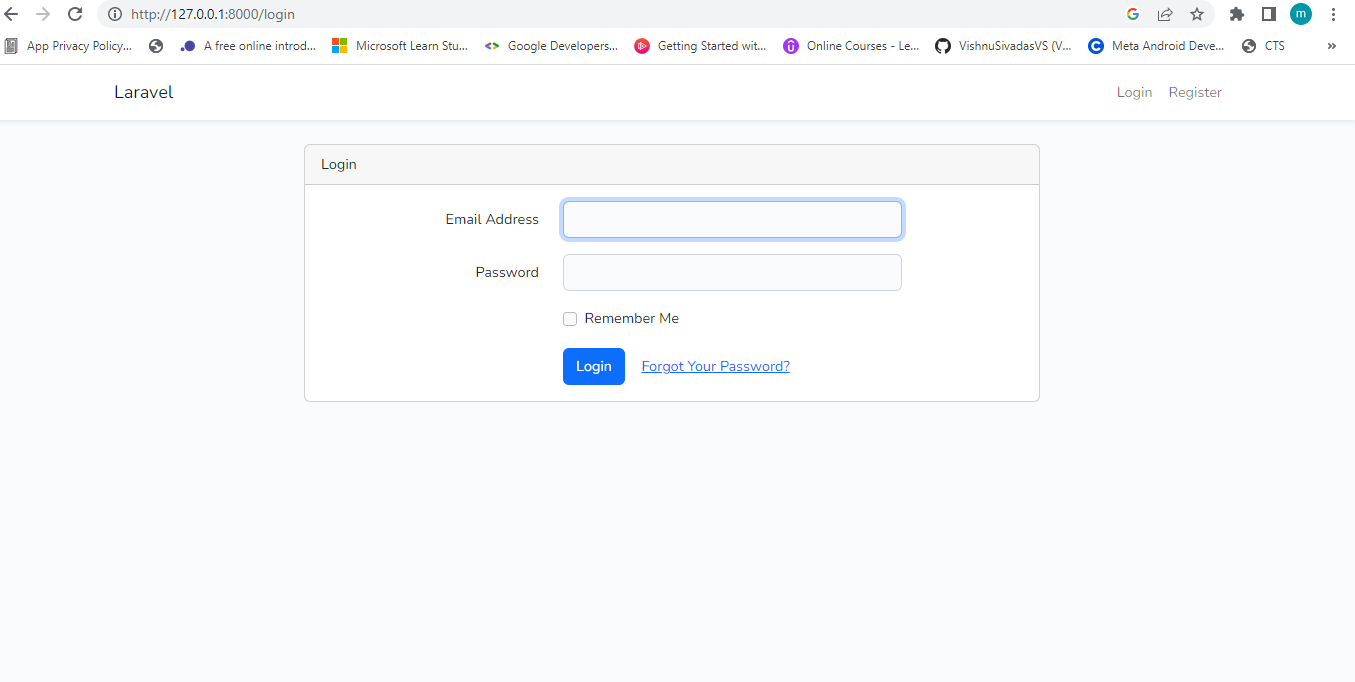
Overall, the project has been successful in achieving its objectives and the final product represents a significant contribution to the field of gym management systems. The system offers a range of features and benefits to both gym members and gym owners, and its user-friendly interface and intuitive design make it easy and give the users a great user experience. The project demonstrates the value of effective project management, agile development methodologies and the waterfall methodologies in achieving the project goals and delivering high-quality software solutions.

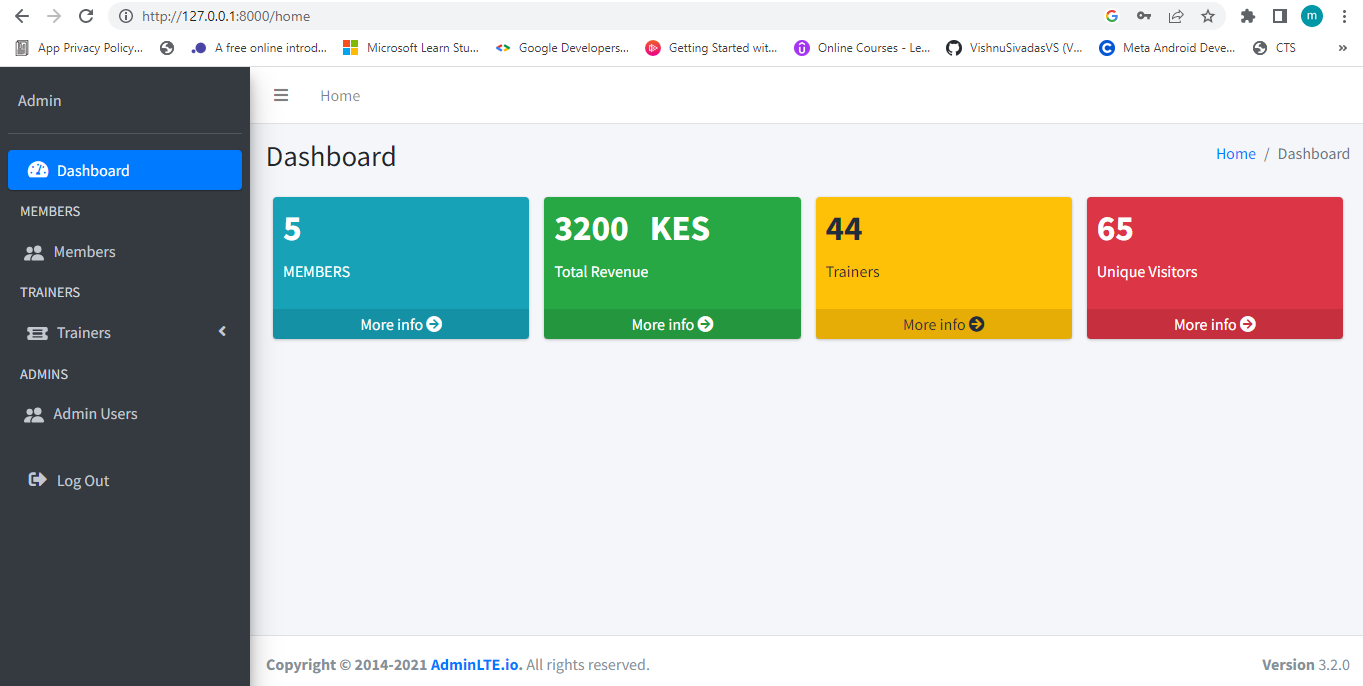
## 4.1 SYSTEM INTEGRATION

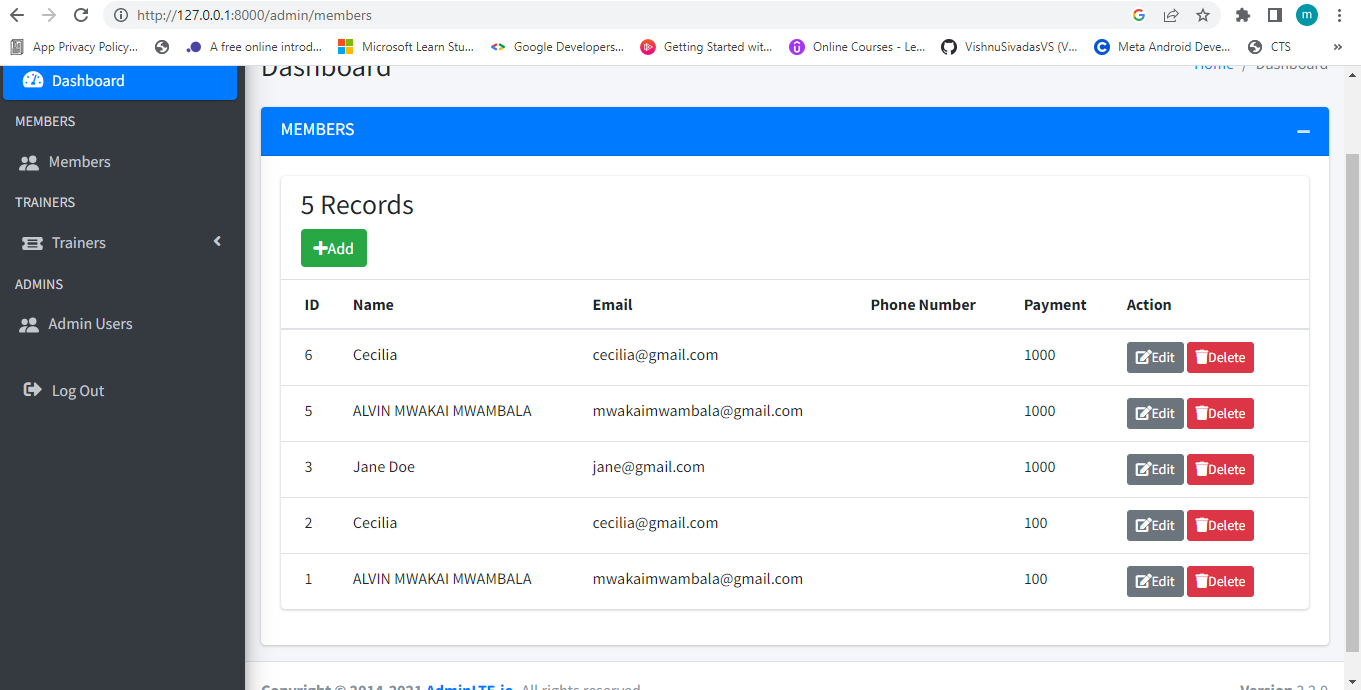
## 4.2 SYSTEM OUTPUT

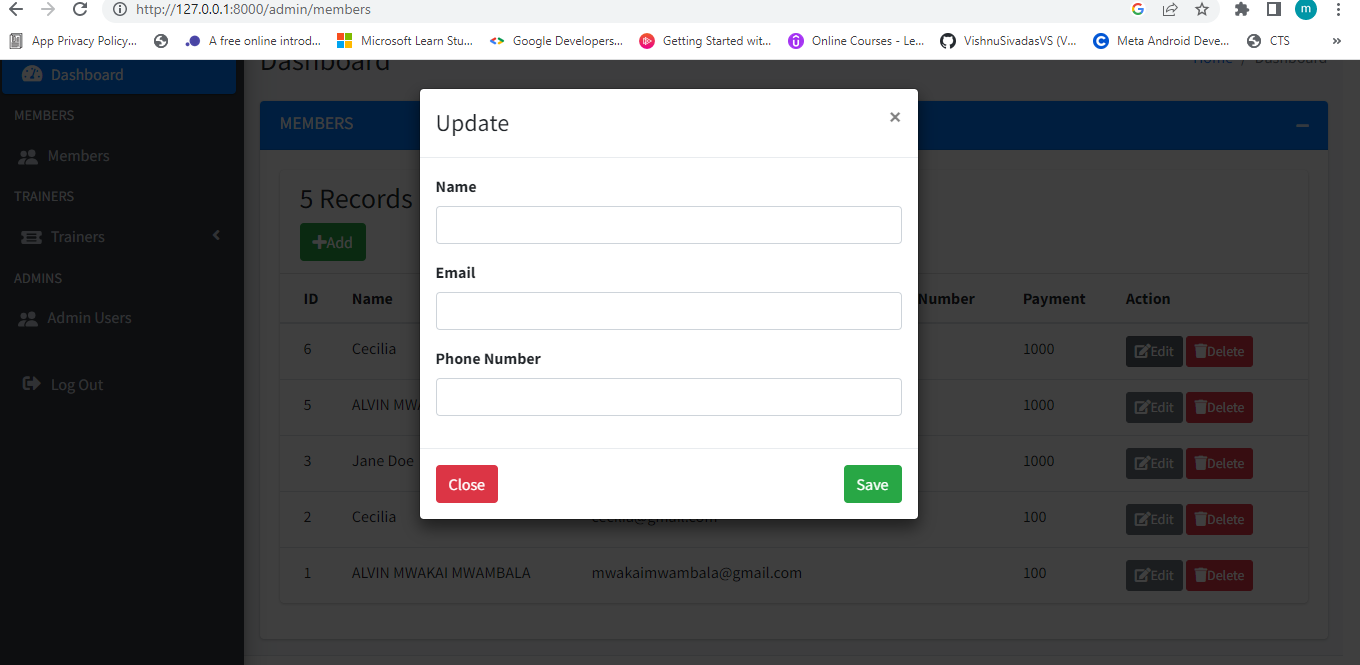
**4.2.1 ADMINISTRATOR**

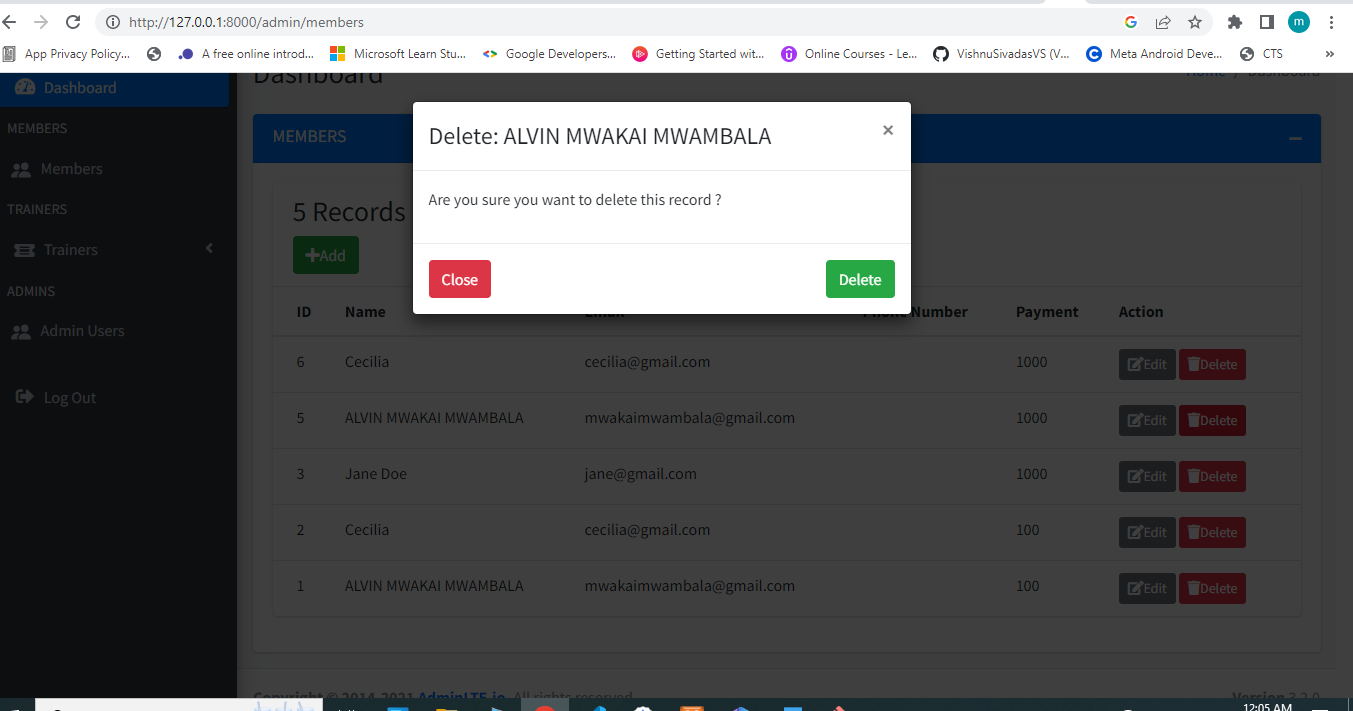




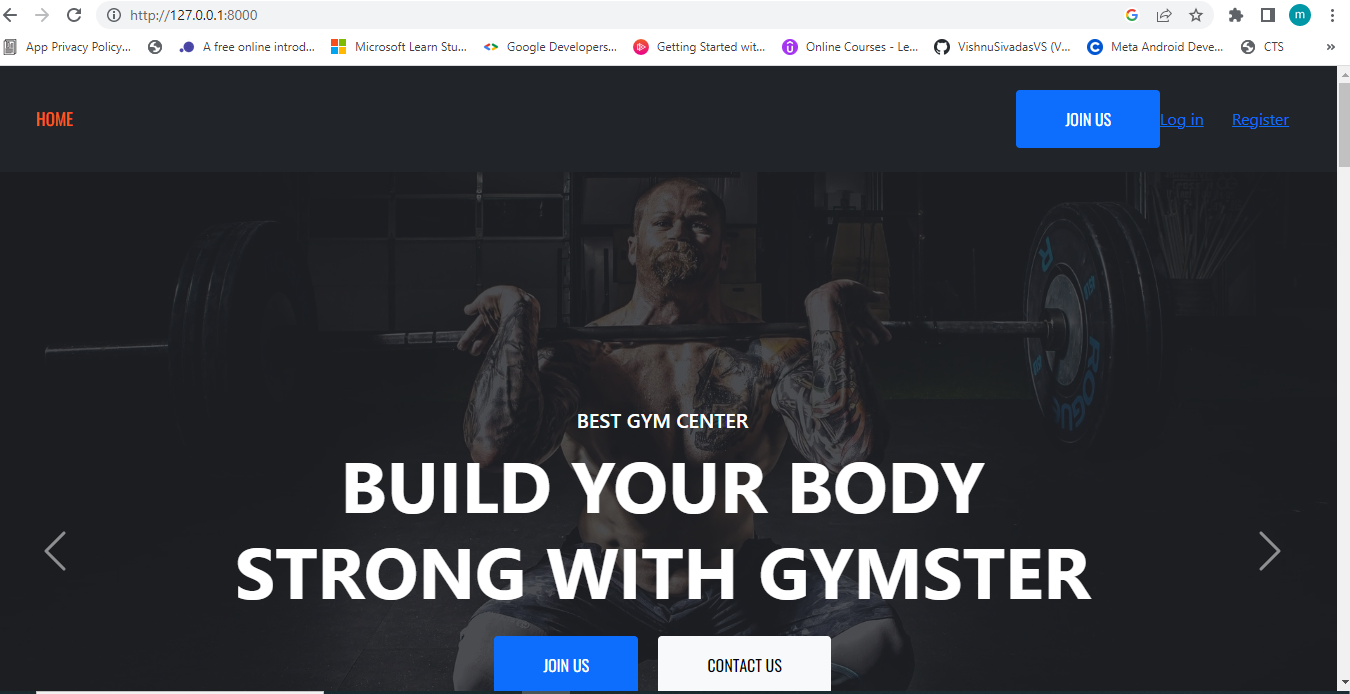


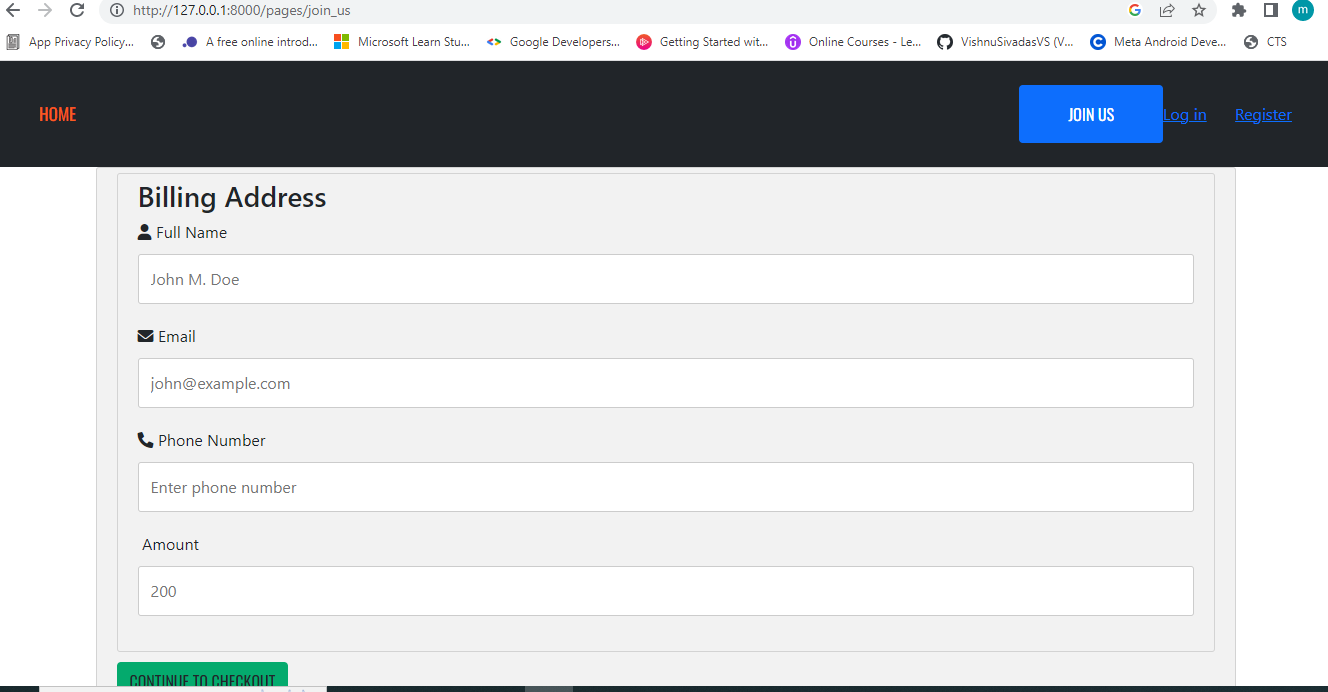






### **4.2.2 User**





## **4.3 SYSTEM TESTING**

The system is tested using unit test

### **4.3.1 Test plan**

Activities involved in a test plan typically include:

Test design - This involves creating test cases that cover all the functionalities of the software application.

Test environment setup - This involves setting up the necessary hardware and software environment for testing.

Test execution - This involves running the test cases to identify defects in the software application.

Test reporting - This involves documenting the results of the tests, including any defects found.

Test closure - This involves evaluating the test results and determining whether the software application is ready for release.

Once the test plan is executed, the testers can draw conclusions based on the results. The conclusion should state whether the software application meets the requirements, functions as intended, and is ready for release.

Users are often asked to provide comments on the software application during the testing process. These comments can be used to identify areas of improvement or to verify that the software application meets their needs. The comments provided by the users should be analyzed and used to improve the software application.

### **4.3.2 ENHANCEMENT**

During system testing the system had a slow network request while trying to implement the payment API, this slow network request can be reduced by reducing the amount of network being called by the API.

Some potential areas for improvement could include enhancing the user interface and user experience, improving the efficiency and speed of the system, adding new features or functionality to meet user needs, fixing any bugs or errors, optimizing the database structure.

# **CHAPTER 5**

## **CONCLUSION**

The system has achieved its main objectives of getting users details and payment digitally.

One of the features that the system in currently unable to process is the use of mail services to send workout sessions and diet timetable to its members.

## **5.1 SYSTEM CONSTRAINT**

The purpose of this document is to outline the limitations of the current fitness management system and identify areas that require future improvements. The system needs implementation such as adding members to different gym schedules and giving them the appropriate trainer and the recommended diet plan to improve the functionality of the system and meet the needs of members.

Limitations of the Current System:

* Lack of functionality to add members to different gym schedules: The current fitness management system does not have the capability to add members to different gym schedules. As a result, the system cannot provide customized workout plans and schedules for members.
* Inability to assign appropriate trainers: The current system does not have a feature that enables assigning the appropriate trainers to members based on their fitness goals, skill level, or workout preferences. This can limit the effectiveness of training programs and reduce member satisfaction.
* Absence of recommended diet plans: The current system does not have the capability to provide recommended diet plans for members. As a result, the system cannot provide a comprehensive fitness program that addresses both workout and diet.

## **5.2 FUTURE ENHANCMENTS**

To address the limitations of the current fitness management system, the following improvements need implementation:

* Adding functionality to allow members to be added to different gym schedules: This feature will enable customized workout plans and schedules for members, based on their fitness goals and preferences.
* Implementing a feature that enables assigning appropriate trainers to members: This feature will enable the system to assign the appropriate trainers to members based on their fitness goals, skill level, or workout preferences. This will improve the effectiveness of training programs and increase member satisfaction.
* Adding a feature to provide recommended diet plans for members: This feature will enable the system to provide recommended diet plans for members based on their fitness goals and preferences. This will help to provide a comprehensive fitness program that addresses both workout and diet.

## **5.3 CONCLUSION**

In conclusion, the current fitness management system has limitations that need to be addressed to improve its functionality and meet the needs of members. The implementation of features such as adding members to different gym schedules, assigning appropriate trainers, and providing recommended diet plans will enhance the system's capabilities and provide a more comprehensive fitness program. These improvements can increase member satisfaction, improve the effectiveness of training programs, and ultimately lead to better fitness outcomes for members. Therefore, it is recommended that the future improvements outlined in this document be considered and implemented in the fitness management system.

## **APPENDIX**

The system is implemented through the model-view-controller architect

## **SAMPLE CODES**

The following is the controller:

<?php

namespace App\Http\Controllers;

use App\Models\Member;

use Illuminate\Http\Request;

use Safaricom\Mpesa\Mpesa;

use Safaricom\Mpesa\Facade\Mpesa as FacadeMpesa;

class MembersController extends Controller

{

    public function index() {

        $query = Member::all();

        $total = count($query);

        $members = Member::latest()->paginate(5);

        return view('admin.members', compact('members', 'total'))->with('i', (request()->input('page', 1) - 1) \* 5);

    }

    public function join\_us()

    {

        return view('pages.join\_us');

    }

    //ADD MEMBER

    public function addMember(Request $request) {

        $request->validate([

            'name' => 'required',

            'email' => 'required',

            'phone' => 'required',

            'payment' => 'required',

        ]);

        Member::create([

            'name' => $request->name,

            'email' => $request->email,

            'phone' => $request->phone,

            'payment' => $request->payment,

        ]);

        /\*\*

         \* DARAJA API TO MAKE MPESA

         \* PAYMENT

         \*/

        $BusinessShortCode = '174379';

        $LipaNaMpesaPasskey = '';

        $TransactionType = 'CustomerPayBillOnline';

        $Amount = $request->payment;

        $PartyA = '2547'; // replace this with your phone number

        $PartyB = '174379';

        $PhoneNumber = '254728408484';

        $CallBackURL = 'https://safaricom.co.ke/mpesa\_online/lnmo\_checkout\_server.php?wsdl';

        $AccountReference = 'Mpesa';

        $TransactionDesc = 'Mpesa';

        $Remarks = 'Mpesa';

        $mpesa= new Mpesa();

        $stkPushSimulation = $mpesa->STKPushSimulation(

            $BusinessShortCode, // This is the paybill number

            $LipaNaMpesaPasskey, // Lipa Na Mpesa Online Passkey

            $TransactionType, // Transaction type CustomerPayBillOnline

            $Amount, // The amount the customer is paying

            $PartyA, // The phone number of the customer

            $PartyB, // The organization shortcode used to receive the transaction.

            $PhoneNumber, // Phone number same as $PartyA

            $CallBackURL, // The url to where responses from M-Pesa will be sent to

            $AccountReference, // Used with M-Pesa PayBills.

            $TransactionDesc, // A description of the transaction.

            $Remarks, // Comments that are sent along with the transaction.

        );

        dd($stkPushSimulation);

        return redirect()->back()->with('success', 'Member Added Successfully');

    }

    //DELETE MEMBER

    public function deleteMember(Request $request) {

        $id = $request->id;

        Member::where('id', $id)->delete();

        return redirect()->back()->with('success', 'Member Deleted Successfully');

    }

}

**The following is the model:**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Member extends Model

{

    use HasFactory;

    protected $table = 'members';

    protected $fillable = [

        'name',

        'email',

        'password',

        'payment',

    ];

}

The following is the view:

@extends('layouts.frontend')

@section('content')

<div class="container">

    <script>

      (function(d, s, id){

        var js, mpesa = d.getElementsByTagName(s)[0];

        if (d.getElementById(id)) {return;}

        js = d.createElement(s); js.id = id;

        js.src = "https://cdn.jsdelivr.net/gh/muaad/mpesa\_button@master/src/button.min.js";

        mpesa.parentNode.insertBefore(js, mpesa);

      }(document, 'script', 'mpesa\_btn\_js'));

    </script>

    <div class="row">

    <div class="col-75">

        <div class="container">

        <form action="{{route('admin.memebers')}}" method="POST">

          @csrf

            <div class="row">

            <div class="col-50">

                <h3>Billing Address</h3>

                <label for="fname"><i class="fa fa-user"></i> Full Name</label>

                <input type="text" id="fname" name="name" placeholder="John M. Doe">

                <label for="email"><i class="fa fa-envelope"></i> Email</label>

                <input type="text" id="email" name="email" placeholder="john@example.com">

                <label for="phone"><i class="fa fa-phone"></i> Phone Number</label>

                <input type="text" id="phone" name="phone" placeholder="Enter phone number" required>

                <label for="adr"><i class="fa fa-address-card-o"></i> Amount</label>

                <input type="text" id="payment" name="payment" placeholder="200">

            </div>

            </div>

            </div>

            <label>

            </label>

            <input type="submit" value="Continue to checkout" class="btn" style="background-color: #04AA6D;">

        </form>

        </div>

    </div>

</div>

@endsection

The following is the database schema:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('members', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('email')->nullable();

            $table->string('phone')->nullable();

            $table->string('payment');

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('members');

    }

};

The following is the setup of the project

APP\_NAME=Laravel

APP\_ENV=local

APP\_KEY=base64:Gufk/{$App\_KEY} =

APP\_DEBUG=true

APP\_URL=http://localhost

LOG\_CHANNEL=stack

LOG\_DEPRECATIONS\_CHANNEL=null

LOG\_LEVEL=debug

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=gym\_management

DB\_USERNAME=root

DB\_PASSWORD=

BROADCAST\_DRIVER=log

CACHE\_DRIVER=file

FILESYSTEM\_DISK=local

QUEUE\_CONNECTION=sync

SESSION\_DRIVER=file

SESSION\_LIFETIME=120

MEMCACHED\_HOST=127.0.0.1

REDIS\_HOST=127.0.0.1

REDIS\_PASSWORD=null

REDIS\_PORT=6379

MAIL\_MAILER=smtp

MAIL\_HOST=mailpit

MAIL\_PORT=1025

MAIL\_USERNAME=null

MAIL\_PASSWORD=null

MAIL\_ENCRYPTION=null

MAIL\_FROM\_ADDRESS="hello@example.com"

MAIL\_FROM\_NAME="${APP\_NAME}"

AWS\_ACCESS\_KEY\_ID=

AWS\_SECRET\_ACCESS\_KEY=

AWS\_DEFAULT\_REGION=us-east-1

AWS\_BUCKET=

AWS\_USE\_PATH\_STYLE\_ENDPOINT=false

PUSHER\_APP\_ID=

PUSHER\_APP\_KEY=

PUSHER\_APP\_SECRET=

PUSHER\_HOST=

PUSHER\_PORT=443

PUSHER\_SCHEME=https

PUSHER\_APP\_CLUSTER=mt1

VITE\_PUSHER\_APP\_KEY="${PUSHER\_APP\_KEY}"

VITE\_PUSHER\_HOST="${PUSHER\_HOST}"

VITE\_PUSHER\_PORT="${PUSHER\_PORT}"

VITE\_PUSHER\_SCHEME="${PUSHER\_SCHEME}"

VITE\_PUSHER\_APP\_CLUSTER="${PUSHER\_APP\_CLUSTER}"

MPESA\_CONSUMER\_KEY=

MPESA\_CONSUMER\_SECRET=

MPESA\_ENV= sandbox

This is the Unit test:

<?php

namespace Tests;

use Illuminate\Contracts\Console\Kernel;

use Illuminate\Foundation\Application;

trait CreatesApplication

{

    /\*\*

     \* Creates the application.

     \*/

    public function createApplication(): Application

    {

        $app = require \_\_DIR\_\_.'/../bootstrap/app.php';

        $app->make(Kernel::class)->bootstrap();

        return $app;

    }

}

## **TECHNICAL GUIDE AND USER MANUAL**

### **USER MANUAL**

**Introduction**

Welcome to the Gym Membership System user manual. This guide will walk you through using our platform to manage your gym membership and make the most of the features we offer.

**Getting Started**

To begin, access the Gym Membership System by opening your web browser and navigating to the provided URL. If you're a new user, click the "Sign Up" button to create your account. Provide your name, email, and password, then click "Register." For existing users, simply click "Log In" and enter your credentials to access your account.

**Using the System**

Once you're logged in, you can explore the various sections of the system. In the "Memberships" section, you can choose and purchase the membership plan that suits your needs. Head to the "Classes" section to view available classes and enroll in the ones that fit your schedule. Your profile information can be managed in the "Profile" section, where you can update your contact details, profile picture, and password. If you ever encounter any issues or have questions, don't hesitate to reach out to our support team at support@gmail.com.

This concludes the Gym Membership System user manual. Thank you for choosing our platform to manage your gym experience. If you have any further questions or need assistance, feel free to refer to the contact information provided above.

### **TECHNICAL GUIDE**

**Introduction**

Welcome to the Gym Membership System's technical guide. This document provides an overview of the system's architecture, technologies, and deployment process, focusing on its implementation using the Laravel framework.

**System Architecture**

The Gym Membership System follows a client-server architecture, with the frontend and backend components interacting to provide a seamless user experience. The frontend is built using HTML, CSS, and JavaScript, incorporating Laravel's Blade templating engine for dynamic content rendering. On the backend, the Laravel framework handles routing, request processing, and database interactions. The Model-View-Controller (MVC) design pattern is central to the system's architecture, ensuring clear separation of concerns and maintainability. The system's data is stored in a MySQL database, with Eloquent ORM providing an intuitive way to interact with the database using PHP classes.

**Technologies and Deployment**

The system leverages Laravel's comprehensive feature set to deliver robust functionality. Laravel's authentication scaffolding is employed for user registration and login processes, enhancing security. Eloquent ORM simplifies database interactions, allowing developers to work with database records as PHP objects. Payment integration is facilitated through Laravel's support for various payment gateways. To deploy the system, follow these steps: set up a web server (e.g., Apache, Nginx), configure the server to point to the project's public directory, and set environment variables for database connection, payment gateway credentials, and other sensitive information. Version control using Git is recommended to track changes, and continuous integration tools can automate testing and deployment processes.

In conclusion, the Gym Membership System's technical foundation relies on the Laravel framework's capabilities. This architecture ensures scalability, maintainability, and security, while also enabling smooth integration of features like user authentication, database management, and payment processing. Developers familiar with Laravel's conventions will find it straightforward to extend and customize the system's functionality to meet specific requirements.