ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ
VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

A MINI PROJECT REPORT ON

"GYM MANAGEMENT SYSTEM"

Submitted to Visvesvaraya Technological University in partial fulfillment of the requirement for the award of degree of

Bachelor of Engineering

in

Computer Science and Engineering.

Submitted by:

Ms. Pavana R	4JN19CS067
Ms. Sakshi M	4JN19CS086
Ms. Skanda G N	4JN19CS098

Under the Guidance of:

Dr. Jalesh Kumar Ph.D.	**Dr. Chetan K. R** Ph.D.	**Mrs. Ganavi M** M.Tech., (Ph.D.)
Professor, CSE	Associate Prof., CSE	Assistant Prof., CSE

Department of Computer Science & Engineering

Jawaharlal Nehru New College of Engineering

Shivamogga - 577 204

January 2022

National Education Society ®

Jawaharlal Nehru New College of Engineering

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the Mini-project Report entitled

GYM MANAGEMENT SYSTEM

Submitted by:

Pavana R	4JN19CS067
Sakshi M	4JN19CS086
Skanda G N	4JN19CS098

Students of 5th semester B.E under the supervision and guidance towards the partial fulfillment of the requirement for award of degree of the Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2021 – 2022.

Signature of Guides and HOD:

Dr. Jalesh Kumar Ph.D.	**Dr. Chetan K R** Ph.D.	**Mrs. Ganavi M** M.Tech., Ph.D.	**Dr. Poornima K M** Ph.D.,
Professor, CSE	Associate Prof., CSE	Associate Prof., CSE	Professor & Head, CSE

Name of Examiners **Signature with Date**

1.

2.

ABSTRACT

In this modern world, computer becomes more and more popular and important to our society. We can use computer everywhere and they are very useful and helpful to our daily life. Like computers online websites has a crucial role in the daily life. Now we have the facility to know about anything in the world through the various sites in a single click. So here we aimed to develop a site based on Gymnasium for the people who wish to maintain their health and body fitness regularly. Gym Management System allows the user to store trainer details, employee details, the details of person who is in the gym, equipment details etc. This software package allows storing the details of all the data related to a gymnasium. The newly developed site for Gymnasium is more suited than the manual database because it provides the facilities like, large storage capacity, high speed, more accuracy and high security. This project uses Visual Studio Code and MySQL.

ACKNOWLEDGEMENT

First and the foremost, I take this opportunity to express my deep sense of gratitude to my guide Dr. Jalesh Kumar, Dr Chetan K R and Mrs. Ganavi M Dept. of CS&E, J.N.N.C.E Shimoga for his kind support, guidance, support, interest and encouragement throughout the course of this mini-project work.

I would like to express my sincere gratitude to Dr. Poornima K M, Prof and Head, Dept. of CS&E, J.N.N.C.E., Shimoga for her kind support, guidance and encouragement throughout the course of this work.

I express my heartfelt thanks to our principal Dr. K Nagendra Prasad, and the esteemed institution J.N.N.C.E for providing me an opportunity to fulfill my most cherished desire of reaching my goal.

I would like to thank all teaching and non-teaching staff of the Dept. of CS&E for their kind co-operation during the course of the work. The support provided by the college and department library is gratefully acknowledged.

Finally, I am thankful to my parents and friends, who helped me in one way or the other throughout this project work.

4JN19CS067 Pavana R

4JN19CS086 Sakshi M

4JN19CS098 Skanda G N

CONTENTS

Abstract	i
Acknowledgment	ii
List of tables	v
List of figures	vi
Chapter- 1: Introduction----------	**1-4**
1.1 Overview of gym management system	1
1.2 Application	1
1.3 Problem statement	1
1.4 Objectives	2
1.5 Overview of Django	2
1.5.1 Architecture	2
1.5.2 Features	2
1.6 Overview of MYSQL	3
Chapter-2: Design and Implementation----------	**5-19**
2.1 Functional requirements	5
2.1.1 Functionalities of gym management system	5
2.1.2 ER Diagram	6
2.2 Design	8
2.2.1 schema diagram	8

2.2.2 front end mock-up	8
2.3 HTML	12
2.3.1 W3.CSS	12
2.4 Django MVT system	13
2.5 Django admin interfaces	15
2.6 Implementation	16
2.6.1 Flowchart for login	16
2.6.2 Flowchart for client registration	17
2.6.3 Flowchart for booking	18
2.7 APIs used	18
Chapter-3: Results----------	**20-28**
3.1 Front end Results	**21**
3.2 Admin interface	**25**
3.3 Backend (MySQL) snapshots	**25**
Chapter-4: Conclusions and future scope----------	**29**
4.1 Future scope	29
References	30

LIST OF TABLES

Table 2.1	Entities and their attributes	16
Table 2.2	Relationships between entities	17
Table 2.3	APIs used	29
Table 3.1	hello_gym	37
Table 3.2	hello_client	37
Table 3.3	hello_equipment	37
Table 3.4	hello_package	38
Table 3.5	hello_trainer	38
Table 3.6	hello_equipment_gym_id	39
Table 3.7	hello_gym_pk_id	39

LIST OF FIGURES

Fig 2.1 ER Diagram of gym management system 15

Fig 2.2 Schema Diagram of gym management system 17

Fig 2.3 Mock-up of home page 18

Fig 2.4 Mock-up of login page 18

Fig 2.5 Mock-up of client registration page 19

Fig 2.6 Mock-up of booking page 19

Fig 2.7 Mock-up of search page 20

Fig 2.8 Mock-up of search trainer page 20

Fig 2.9 Mock-up of about us page 21

Fig 2.10 Mock-up of contact us page 21

Fig 2.12 Mock-up of my trainer page 22

Fig 2.13 Control flow of MVT 24

Fig 2.14 Flowchart for login 26

Fig 2.15 Flowchart for client registration 27

Fig 2.16 Flowchart for booking 28

Fig 3.1 login 30

Fig 3.2 client registration 30

Fig 3.3 home page 31

Fig 3.4 booking 32

Fig 3.5 alert message 1 32

Fig 3.6 alert message 2 33

Fig 3.7 about us 33

Fig 3.8 contact us 33

Fig 3.9 search 34

Fig 3.10 gym searched 34

Fig 3.11 search trainer 35

Fig 3.12 trainer searched 35

Fig 3.13 my trainer 35

Fig 3.14 admin interface 36

Fig 3.15 tables in the backend 36

CHAPTER-1: INTRODUCTION

1.1 Overview of gym management system

Our Gym Management Software is a gym and health club membership management system. You can keep records on your members, their memberships, and have quick and easy communication between you and your members. Gym Management also includes a booking system. Our Gym Management Software is a complete gym and recreation facility system program which looks after all of gyms, trainers, members, memberships and activities. Our gym management Software provides lots of functions such data entry of customer, keeping records of all the things about customer's fees and plan which help to provide good quality of services to customer from Gym managers. In this proposed system also provide the total information about machinery and data of coaches is also stored in it. Services provided by Gym are also handled by this system.

1.2 Application

The main application of the project is to design and develop a user-friendly system.

- Easy to use and efficient computerized system.
- To develop an accurate and flexible system, it will eliminate data redundancy.
- Computerization can be helpful as means of saving time & money.
- To provide better graphical user interface.
- Less chances of information leakage.
- Provides security to data by using login password.

1.3 Problem statement

As the records are to be manually maintained it consumes a lot of time, lots of paper work is involved as the records are maintained in the files & registers, as files and registers are used the storage space requirement is increased, use of papers for storing valuable data information is not at all reliable. As the system is in manual there are lot many chances of human errors. These can cause errors in calculating mechanism or maintaining customer details. It is difficult for keeping all the new entries of members, their account and transaction details. Also, people need to head out to all the gym's

present and acquire the information about the trainers and cost but the gym management system addresses all these problems under one roof.

1.4 Objectives

The main objective of the project is to develop software that facilitates the data storage, data maintenance and its retrieval for the gym in an igneous way.

- To store the record of the customers, the staff that has the privileges to access, modify and delete any record and finally the service, gym provides to its customers.
- Also, only the staff has the privilege to access any database and make the required changes, if necessary.
- To develop easy-to-use software which handles the customer-staff relationship in an effective manner.
- To develop a user-friendly system that requires minimal user training. Most of features and function are similar to those on any windows platform.

1.5 Overview of Django

1.5.1 Architecture

Django follows a Model-View-Controller (MVC) architecture, which is split up into three different parts:

- The **Model** is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).
- The **View** is the user interface — what you see in your browser when you visit a website. These are represented by HTML/CSS/JavaScript files.
- The **Controller** is the middleman that connects the view and model together, meaning that it is the one passing data from the model to the view.

So, say a user will enter a URL in their browser, that request will go through the internet protocols, to your server, which will call Django. Django will then process the given URL path, and if it matches an URL path you have explicitly stated, it will call the **Controller**, which will then perform a certain action, such as get an entry from your **Model**(database) and then render a **View** (i.e.: JSON text, HTML/CSS/JavaScript Web page).

1.5.2 Features:

- **Portable:** Since Django is a Python-based web framework it runs on almost all platforms. Almost all Linux flavor's, Windows Editions, and Mac Versions support Django
- **Speed:** Frameworks are usually developed to make the application development process easier and faster without having to write code from scratch to even simple things.
- **Security:** Django developers made security a priority while developing the language. Django's application source code doesn't show up in web browser source code and offers various other features like scanning uploaded content and access control security. Django is a versatile framework for the robust security it offers.
- **Scalability:** Django can handle large amounts of traffic and provide API Usage to around 450+ million users. Django makes applications scalable by providing the ability to plug and unplug default components.
- **Versatile:** Typically, frameworks are only used by the publishing industry on the Internet. Django is by far the most versatile framework. Developers can build almost anything, ranging from social networks to financial platforms. In fact, the framework name "Django" literally means – multifaceted nature.
- **Packages:** Django is a full-stack web development framework kit as it offers various packages to build a fully functional application. It restricts the developers from relying on extra technologies by providing packages for almost everything. There are many packages available, Django Developers need to simply use these packages according to their use case.

1.6 Overview of MySQL

MySQL is popular among all databases, and is ranked as the 2nd most popular database, only slightly trailing Oracle Database. Among open-source databases, MySQL is the most popular database in use today, known as one of the most reliable and performative databases out there. MySQL is a relational database that uses structured query language. Relational databases are a type of database that uses a structure that allows us to identify and access the data in relation to another piece of data inside of the database. This format

is often organized as tables. MySQL is a highly scalable product and that scalability can come from several different performance tuning techniques.

Features of MySQL are:

- **Easy to Use:** MySQL is considered easy to use among RDBMS. It works with basic SQL and, given its maturity and adoption, there is abundant documentation available.
- **Secure:** MySQL's maturity also lends itself to security. It's regularly updated and has a vibrant developer community. These factors combine to make MySQL a stable and secure choice among RDBMS.
- **Open Source:** The community edition of MySQL is enterprise ready, and supported by a GNU General Public License. For users who want access to equitable proprietary functionality of MySQL without the added price tag.
- **Scalable:** MySQL is highly scalable for an RDBMS, with a wide range of options not covered in this blog that allow for tuning, customizing and enhancing your MySQL experience.
- **Reliable:** MySQL is reliable — not just from a data perspective, but from a development perspective. It's mature, it has regular releases, patches, and an entrenched developer community that works with it. This makes it a safe choice compared to newer, less mature RDBMS options.

CHAPTER-2: DESIGN AND IMPLEMENTATION

2.1 Functional Requirements

2.1.1 Functionalities of gym management system

Gym Management system is proposed to be an automate database management. This stores employee, member, payroll, trainer and products information. It also provides the facility of search and advanced search for searching the records efficiently and immediately. This system provides data storing & report generation with graphical user interface (GUI).

Some of the services provided by this system to the end users are:

- Storing information of members according to their id.
- To reduce the burden of paper work.
- To save time management for recording details of each and every member and employee.
- The proposed system is highly secured, because for login the system it requires the username and password which is different for each department therefore providing each department a different view of the customer information.
- It provides wide range of certain criteria in each window the client is working for better and quicker solution.
- It maintains report for all criteria and transactions.
- Manages trainer information separately for all gyms and employee information separately for considering the requirements of gym.
- Stores information about gym equipment's.
- This system can run on any operating system

2.1.2 ER Diagram

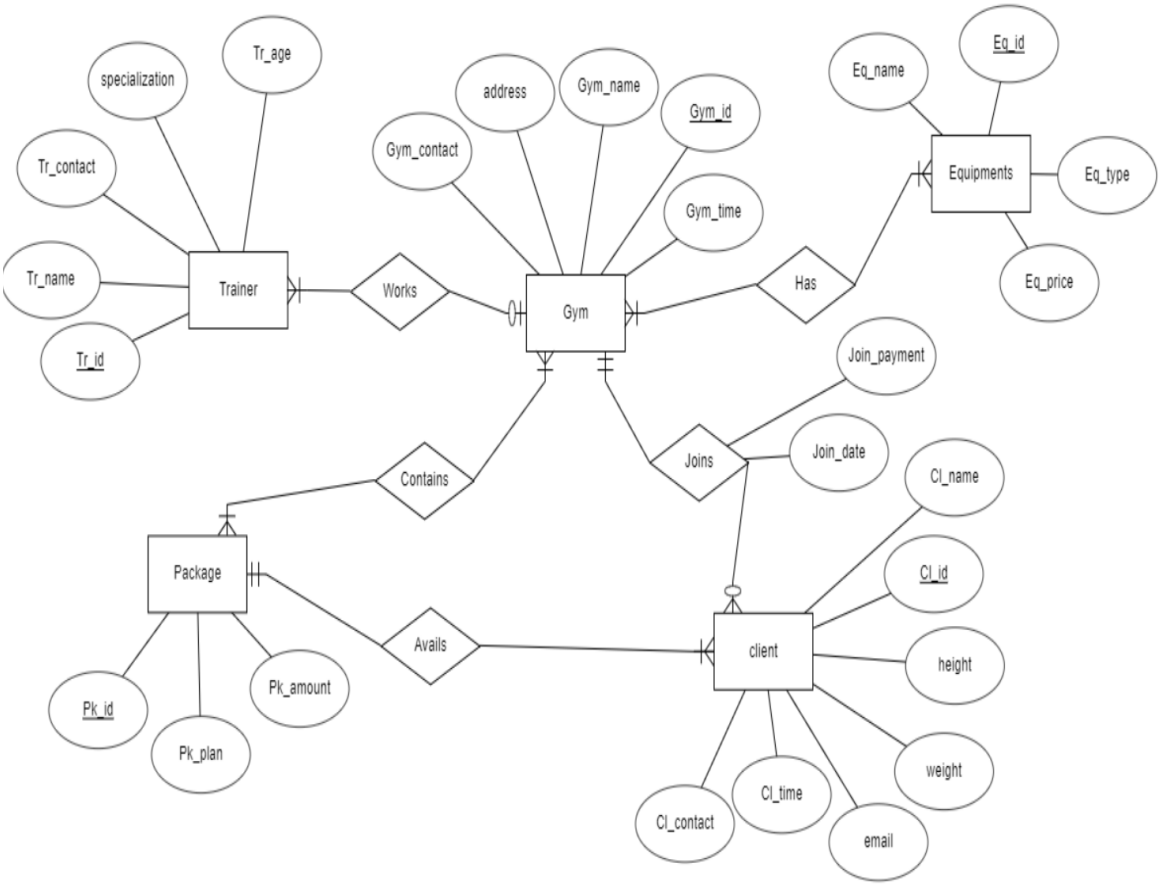


Fig 2.1 ER Diagram of gym management system

Figure 2.1 represents the ER Diagram of gym management system. There are 5 entities along with their respective attributes and a relationship between entities.

The entities and attributes in the ER Diagram are as follows:

1. **Trainer:** the attributes in this entity are trainer age **(Tr_age**), field in which the trainer is specialized(**specialization**), contact number of the trainer (**Tr_contact**), name of the trainer **(Tr_name**) and trainer ID(**Tr_id**).
2. **Gym:** the attributes of the entity gym are as follows, gym id (**Gym_id**) this a unique number representing the gym's, name of the gym (**Gym_name**), gym timings that is the opening and closing timings of the gym (**Gym_time**), gym contact number (**Gym_contact**) and address of the gym (**Gym_address**).
3. **Client:** the attributes of the client entity are, client id (**Cl_id**) which is unique to all clients, client name (**Cl_name**), height of the client(height), weight of the client(**weight**), email id of the client(**email**), contact info of the client (**Cl_contact**) and the timing in which the client attends the gym (**Cl_time**).

4. **Package:** package had 3 attributes that is package id (**Pk_id**), package plan that is the duration of the plan (**Pk_plan**) and the cost of the package (**Pk_amount**).
5. **Equipment's:** the attributes on this entity are, equipment id (**Eq_id**), name of the equipment (**Eq_name**), type of the equipment that is if it is used for cardio, abs, functional training etc. (**Eq_type**) and price of the equipment **(Eq_price**).

Table.2.1 Entities and their attributes

Entity name	Attributes
Gym	Gym_id, Gym_name, address, Gym_contact, Gym_time
Trainer	Tr_id, Tr_name, Tr_contact, Tr_age, specialization
Client	Cl_id, Cl_name, Cl_contact, Email, Cl_time, weight, height
Package	Pk_id, Pk_plan, Pk_amount
Equipments	Eq_id, Eq_name, Eq_type, Eq_price

The relationship between the entities are as follows:

Table.2.2 Relationships between entities

Entities involved	Relationship	Participation constraint	Cardinality ratio
Trainer - Gym	Works	Gym – mandatory Trainer - optional	Gym – many Trainer - one
Client - Gym	Joins (Join_date, Join_payment)	Gym – optional Client -mandatory	Gym – many Client - one
client - Package	Avails	Client –mandatory Package – mandatory	Client – one Package - many
Gym - Equipments	Has	Gym – mandatory Equipments -mandatory	Gym – many Equipments - many
Gym - package	contains	Gym – mandatory Package - mandatory	Gym – many Package - many

2.2 Design

2.2.1 schema diagram

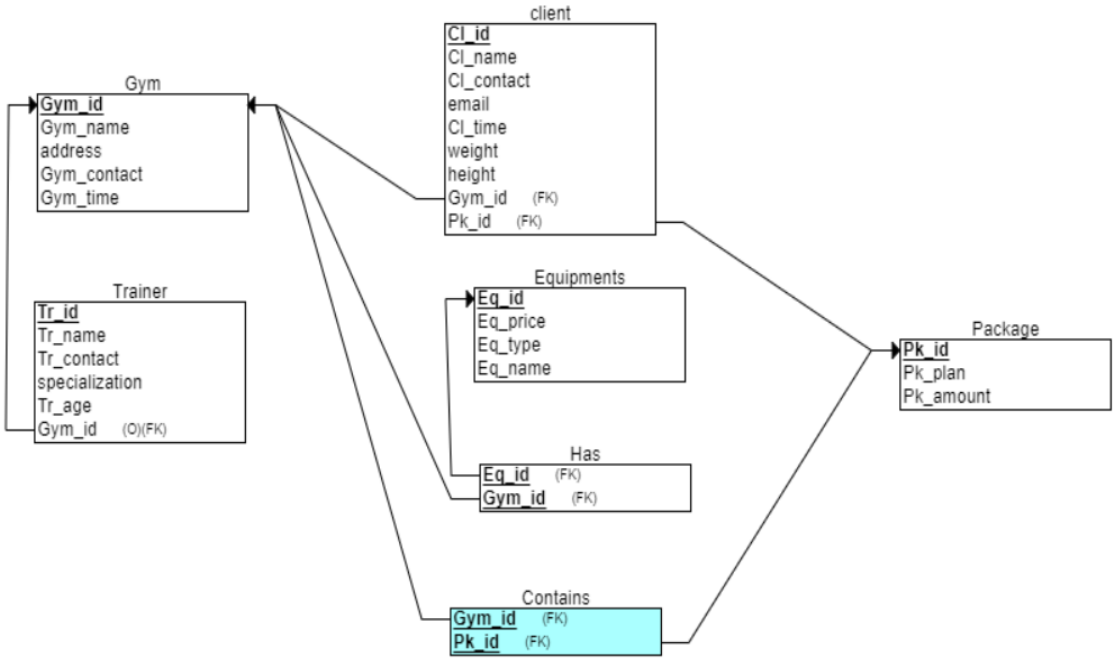


Fig 2.2 Schema Diagram of gym management system

2.2.2 Front end mock-up

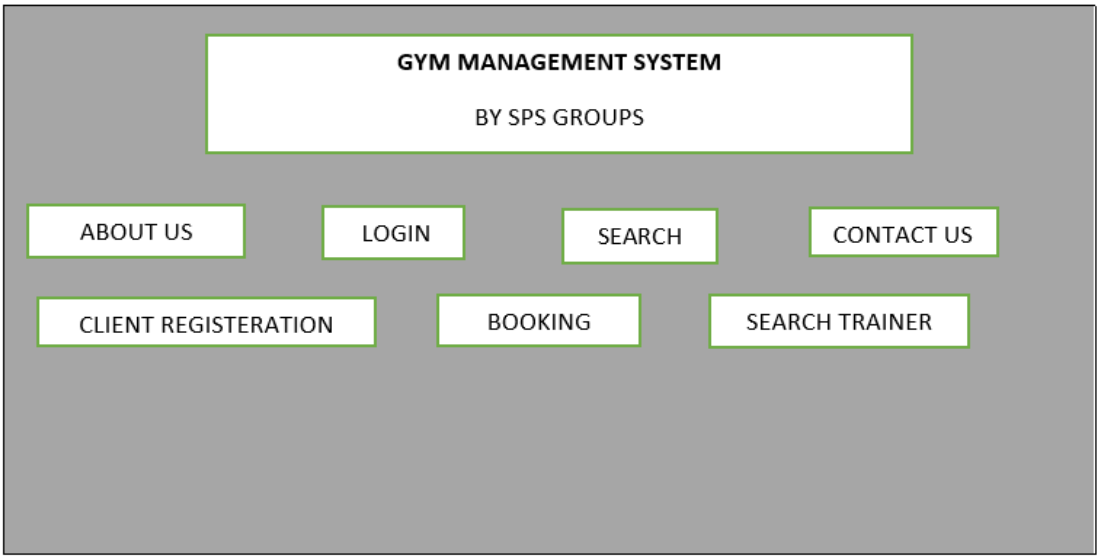


Fig 2.3 Mock up home page

Home page menu consists of the following: login, about us, search, contact us, client registration, booking and search trainer.

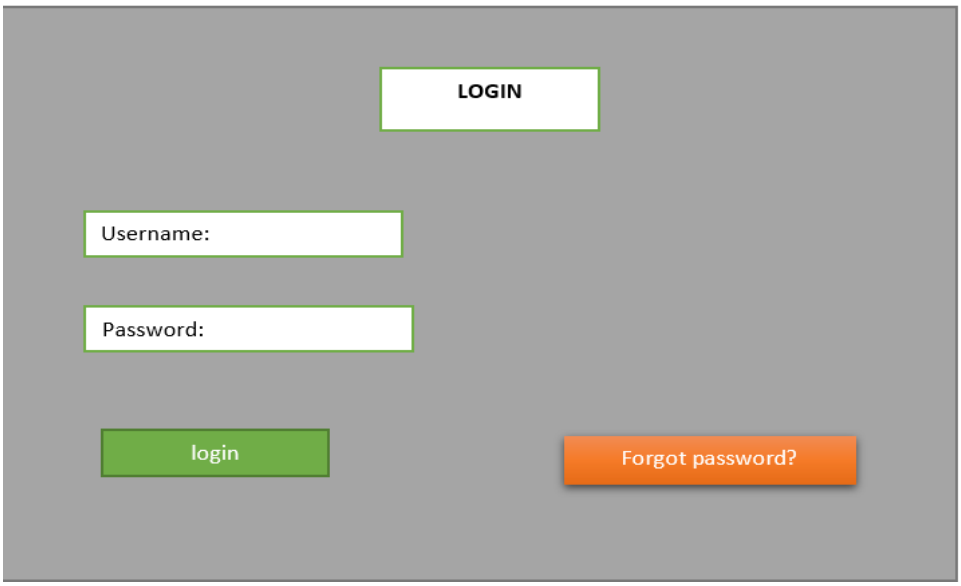


Fig 2.4 Mock up login page

Enter your user name and password and click on the login button, if it is a valid user name and password it will take you to the home page.

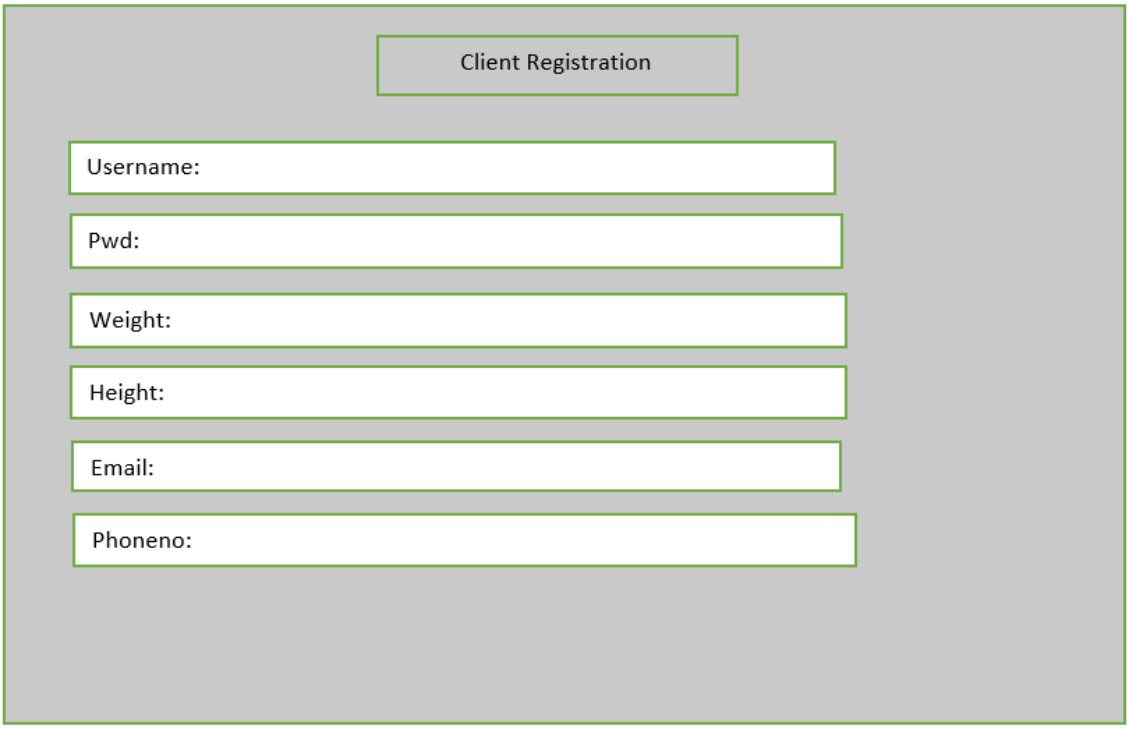


Fig 2.5 Mock up client registration page

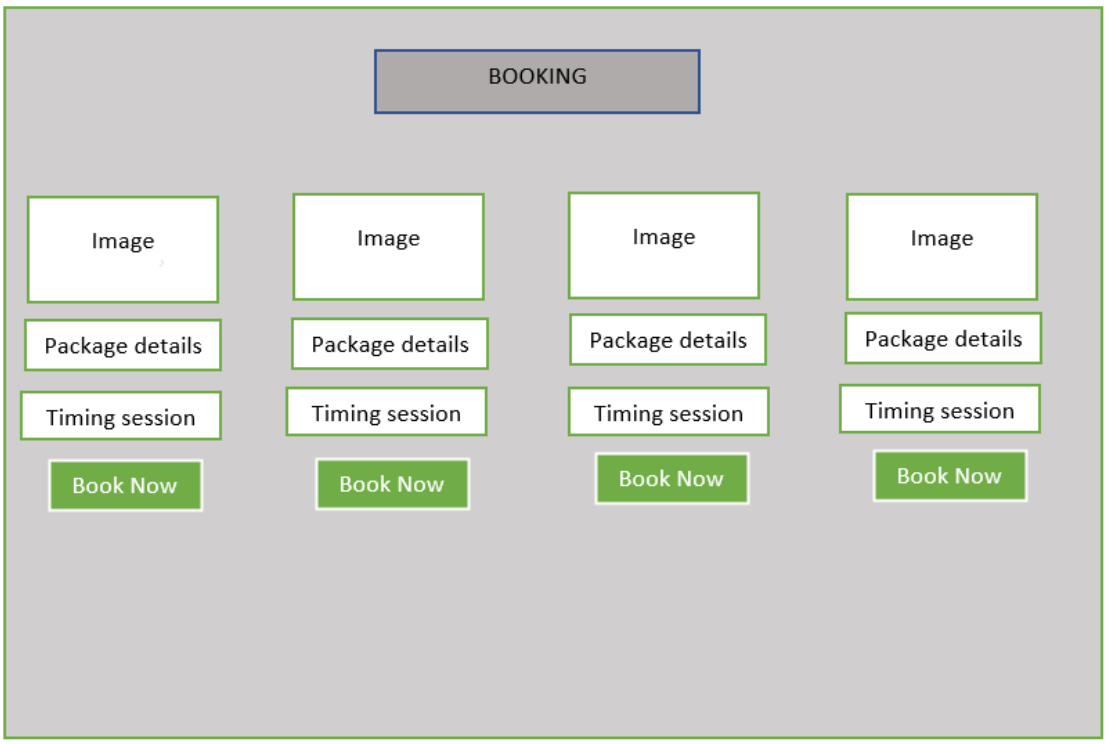
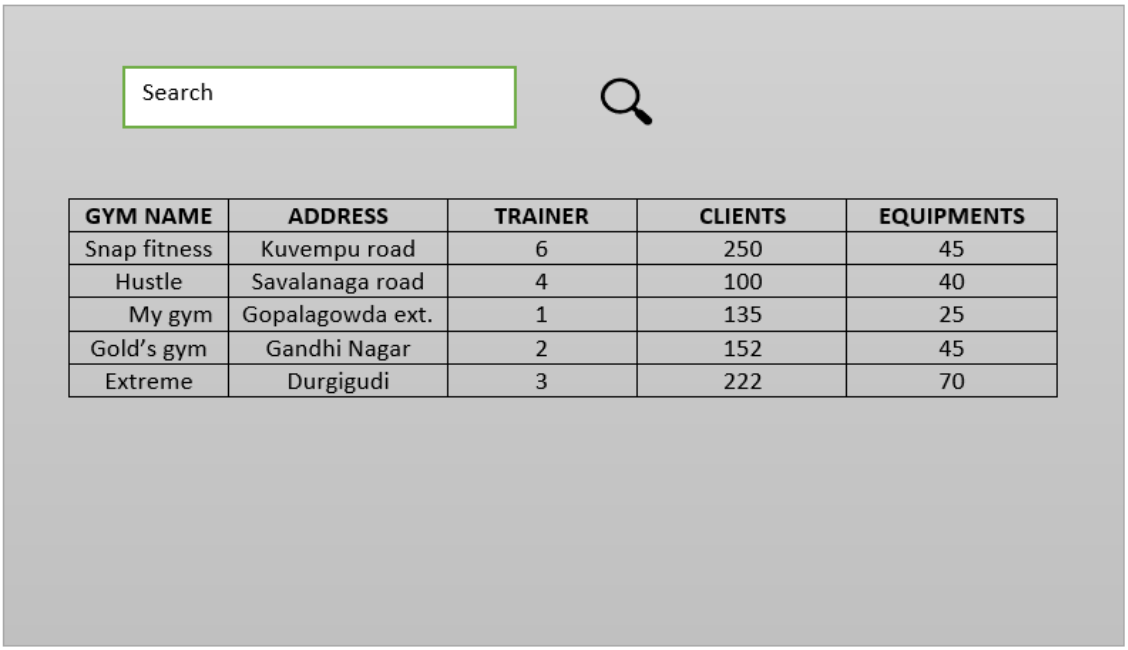
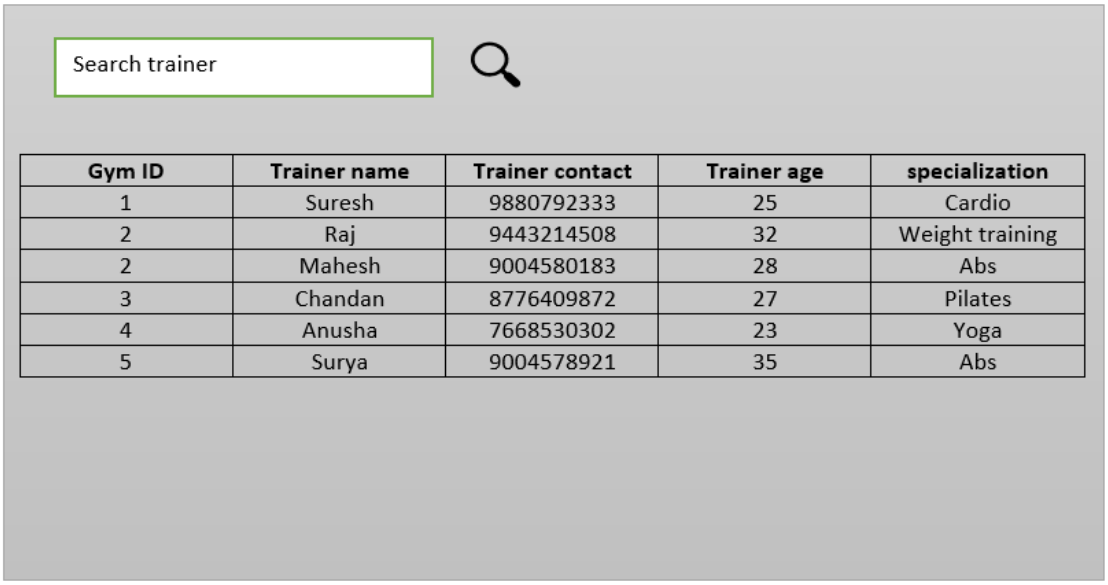


Fig 2.6 Mock up booking page



GYM NAME	ADDRESS	TRAINER	CLIENTS	EQUIPMENTS
Snap fitness	Kuvempu road	6	250	45
Hustle	Savalanaga road	4	100	40
My gym	Gopalagowda ext.	1	135	25
Gold's gym	Gandhi Nagar	2	152	45
Extreme	Durgigudi	3	222	70

Fig 2.7 Mock up search page



Gym ID	Trainer name	Trainer contact	Trainer age	specialization
1	Suresh	9880792333	25	Cardio
2	Raj	9443214508	32	Weight training
2	Mahesh	9004580183	28	Abs
3	Chandan	8776409872	27	Pilates
4	Anusha	7668530302	23	Yoga
5	Surya	9004578921	35	Abs

Fig 2.8 Mock up search trainer page

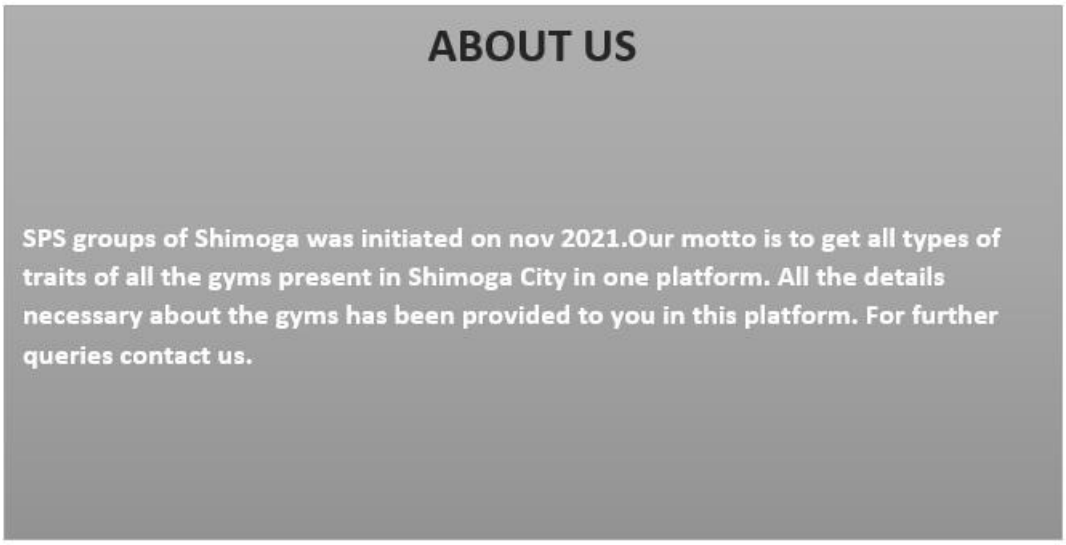


Fig 2.9 Mock up about us page

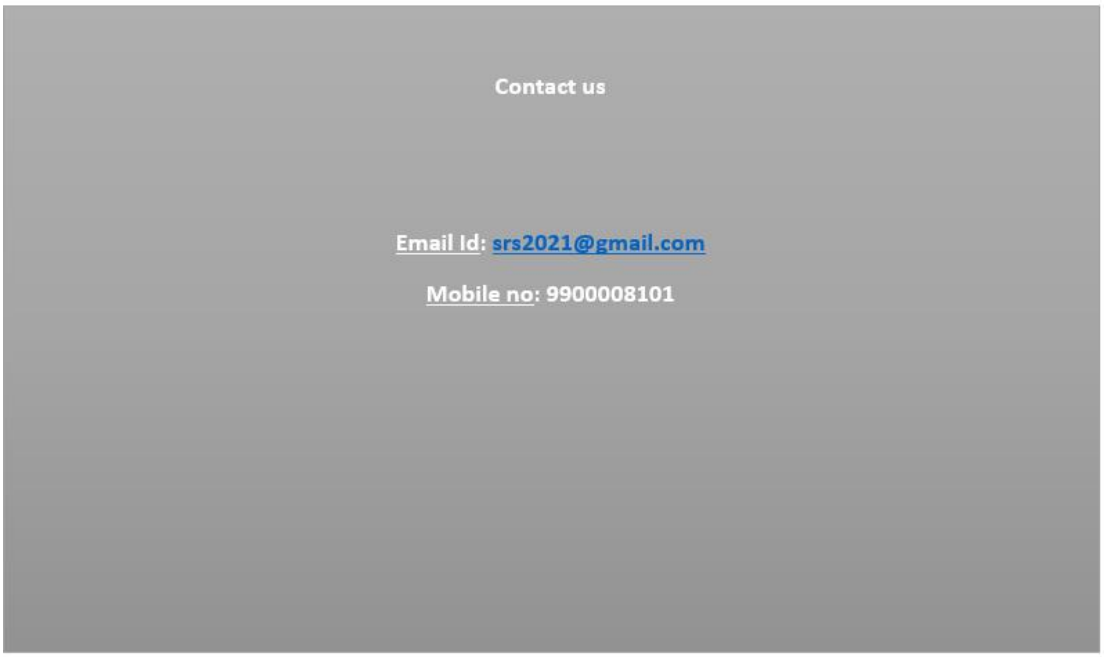


Fig 2.10 Mock up contact us page

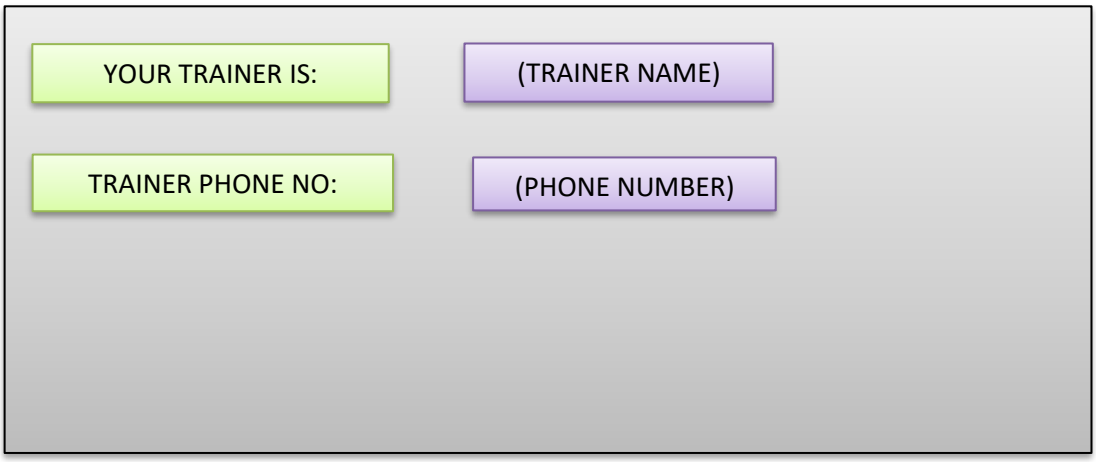


Fig 2.11 Mock up my trainer page

2.3 HTML

HTML stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. HTML 5 is the fifth and current version of HTML. It has improved the markup available for documents and has introduced application programming interfaces (API) and Document Object Model (DOM).

HTML contains lots of predefined tag. Most commonly used tags in HTML are:

- **HTML tag**: It is the root of the html document which is used to specify that the document is html.
  Syntax: <html> Statements... </html>
- **Head tag:** Head tag is used to contain all the head element in the html file. It contains the title, style, meta, … etc tag.
  Syntax**:** <head> Statements... </head>
- **Body tag:** It is used to define the body of html document. It contains image, tables, lists, … etc.
  Syntax: <body> Statements... </body>
- **Title tag:** It is used to define the title of html document.
  Syntax: <title> Statements... </title>
- **Heading tag:** It is used to define the heading of html document.
  Syntax: <h1> Statements... </h>
  <h2> Statements... </h2>
  <h3> Statements... </h3>
- **Paragraph tag:** It is used to define paragraph content in html document.
  Syntax: Statements...
- **Bold tag:** It is used to specify bold content in html document.
  Syntax: Statements...

2.3.1 W3.CSS

W3.CSS is a Cascading Style Sheet (CSS) developed by w3schools.com. It helps in creating faster, beautiful, and responsive websites. It is inspired from Google Material Design. Some of its salient features are it has in-built responsive designing, it's inspired by Google Material Design and it's free to use.W3.CSS is a modern, responsive, mobile first CSS framework. Provides equality for all browsers: Chrome, Firefox, Edge, IE, Safari, Opera and for all devices: Desktop, Laptop, Tablet, Mobile. W3.CSS is standard CSS only (No jQuery or JavaScript library).

2.4 Django MVT system

MVT architecture is the software design pattern used by the Django web framework. Although Django at its core is based on MVC architecture, it actually is implementing a

variation of MVC, called MTV architecture. There are some drawbacks of MVC architecture and it has certain areas where Django is offering a better-quality feature when using the MTV architecture. In this architecture, the model remains the same, it provides the interface for storing the data in the database. Just like View in the MVC model, Django replaces it with a Template in its framework, the controller part in the MVC model is taken care of by View in the MVT architecture.

MVT stands for **Model – View – Template.**

- **Model:** Just like the Model in MVC, here as well it has the same functionality of providing the interface for the data stored in the database**.**
- **Template:** Just like View in MVC, Django uses templates in its framework. Templates are responsible for the entire User Interface completely. It handles all the static parts of the webpage along with the HTML, which the users visiting the webpage will perceive.
- **Views:** In Django, Views act as a link between the Model data and the Templates**.**

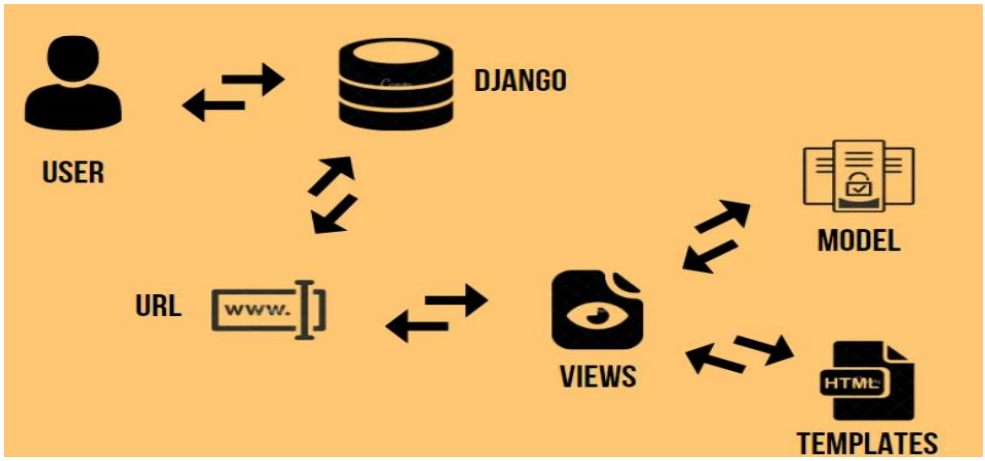


Fig. 2.12 Control flow of MVT

As shown in the Fig. 2.12

1. The user sends a URL request for a resource to Django.
2. Django framework then searches for the URL resource.
3. If the URL path links up to a View, then that particular View is called.
4. The View will then interact with the Model and retrieve the appropriate data from the database.
5. The View then renders back an appropriate template along with the retrieved data to the user.

2.5 Django Admin Interfaces

Django provides a ready-to-use user interface for administrative activities. We all know how an admin interface is important for a web project. Django automatically generates admin UI based on your project models. That interface will let you administrate Django groups and users, and all registered models in your app. Django provides a default admin interface which can be used to perform create, read, update and delete operations on the model directly. It reads set of data that explain and gives information about data from the model, to provide an instant interface where the user can adjust contents of the application. This is an in-built module and design to execute admin related work to the user.

When you ran startproject, Django created and configured the default admin site for you. All that you need to do now is create an admin user (superuser) to log into the admin site. To create an admin user, run the following command from inside your virtual environment:

```
python manage.py createsuperuser
```

Enter your desired username and press enter.

```
Username: admin
```

You will then be prompted for your email address:

```
Email address: admin@example.com
```

The final step is to enter your password. You will be asked to enter your password twice, the second time as a confirmation of the first.

```
Password: **********
Password (again): *********
Superuser created successfully.
```

First, make sure the development server is running, then open a web browser to http://127.0.0.1:8000/admin/. You should see the admin's login screen.

2.6 Implementation

2.6.1 Flowchart for login

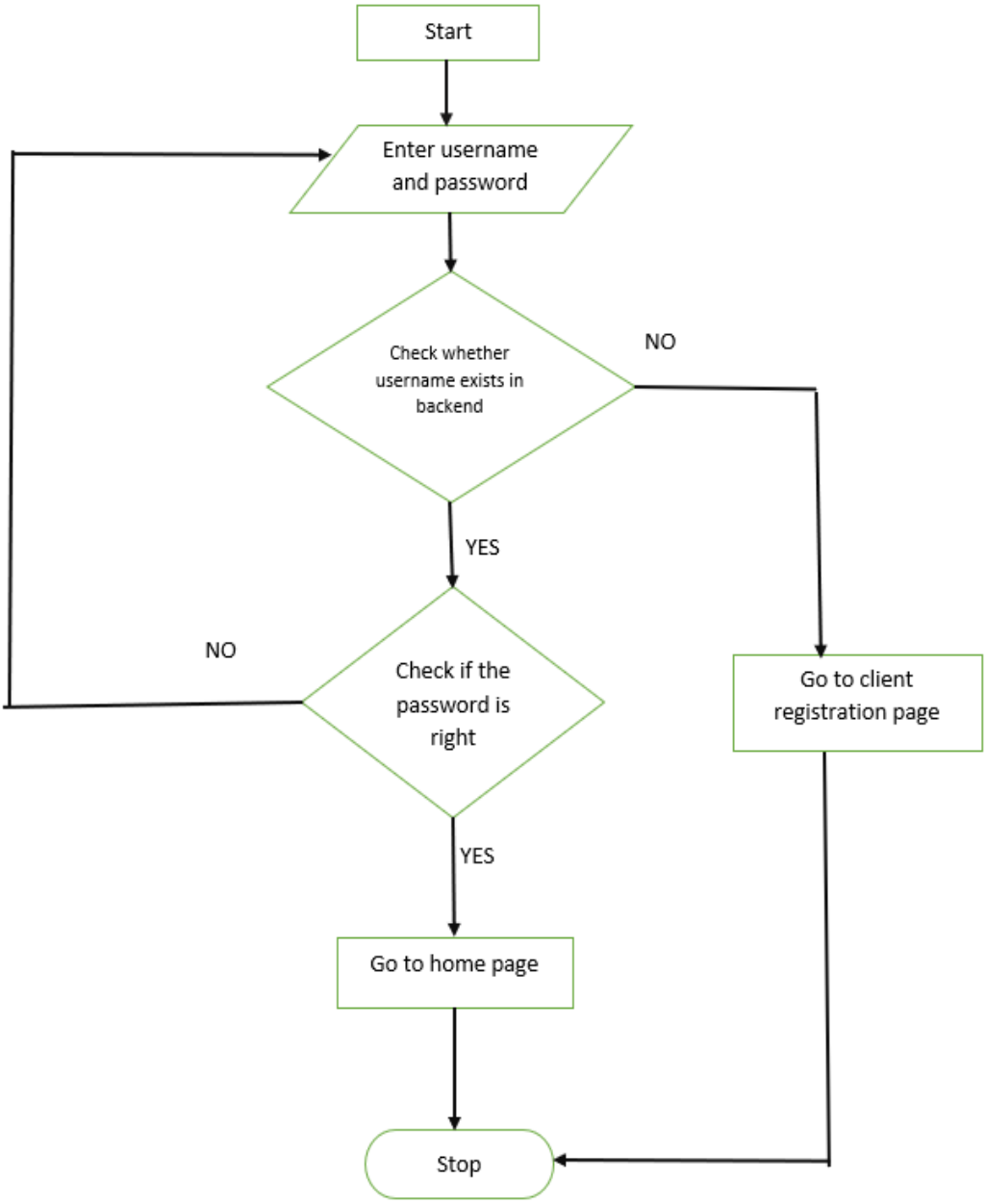


Fig. 2.13 Flowchart for login

2.6.2 Flowchart for client registration

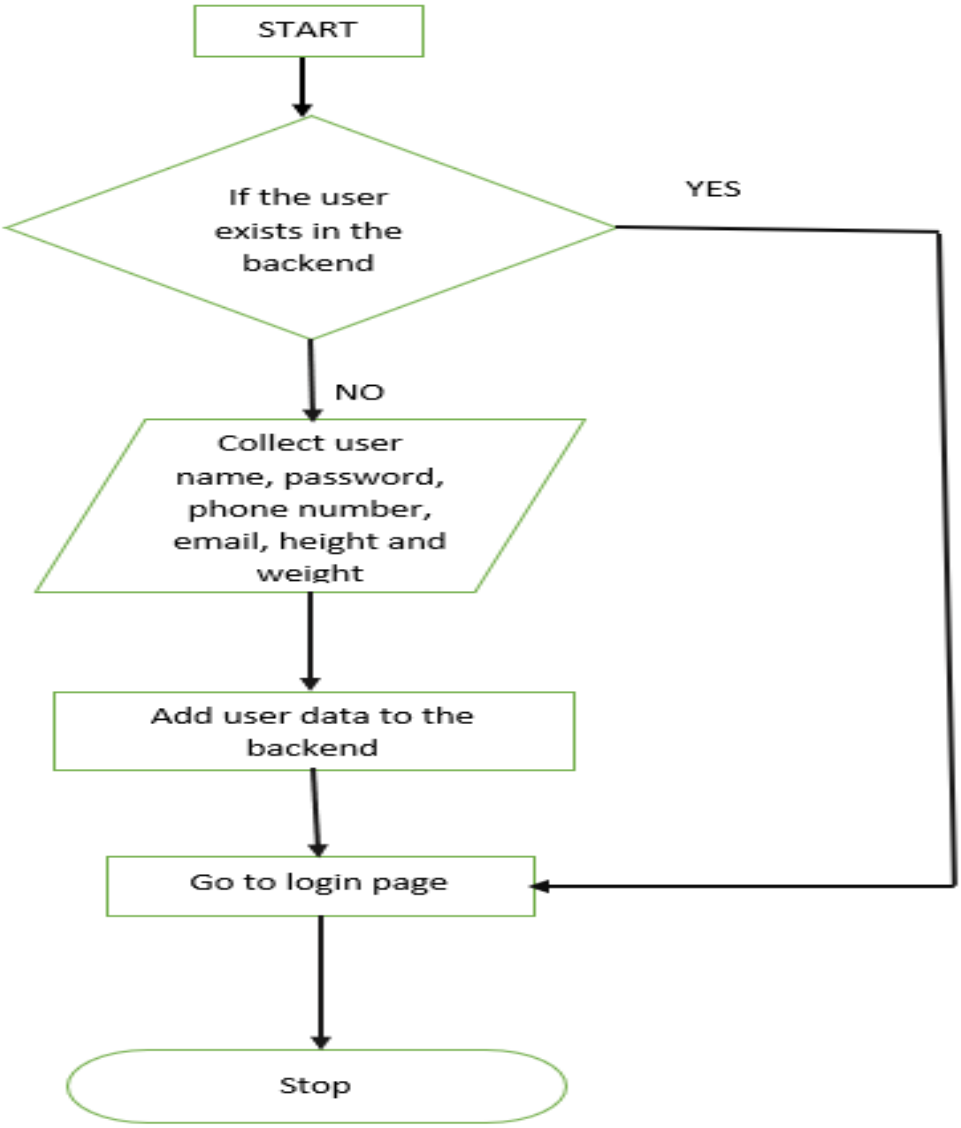


Fig 2.14 Flowchart for client registration

2.6.3 Flowchart for booking

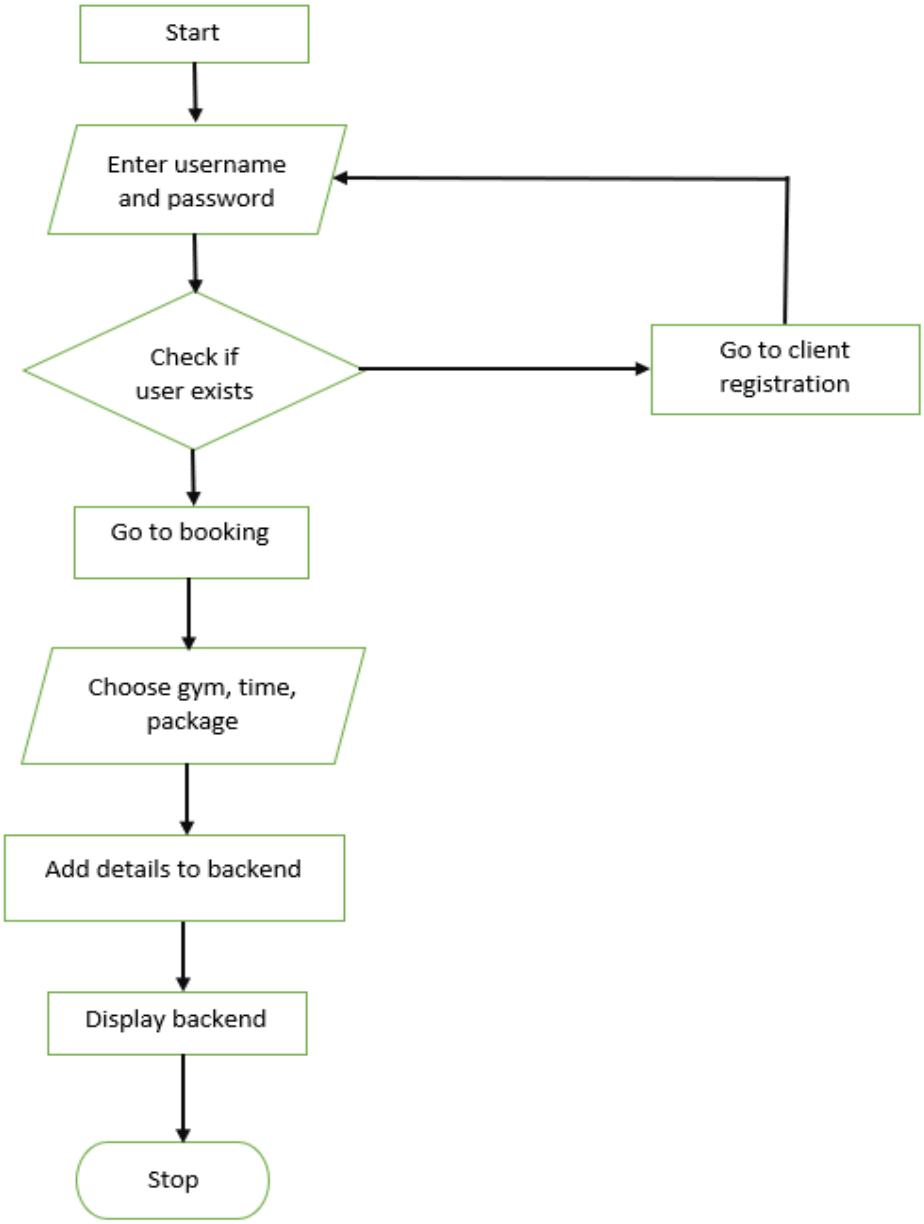


Fig. 2.15 Flowchart for booking

2.7 APIs used

Table 2.3 API's used

Django APIs	
APIs used	**Explanation**

Request.GET.get	request.GET contains the **GET variables**. These are what you see in your browser's address bar. The.get () method is a method used for dictionaries.
objects.filter	objects.filter is to get a matching result from the database, return a list of objects.
objects.get	It is a method of accessing objects in python. It returns the value associated with that object
response.set_cookie	The set_cookie() method in DjangoHttpResponse has a name: Name of the cookie.vale: Value you want to store int or string but it will return string**.**
objects.all	Return a copy of the current QuerySet.
request.COOKIES.get	Using request.COOKIES.get () Django also provides a method to get the desired value from the cookie. You can directly access that value using get method over request object.

Chapter-3: Results

3.1 Front end Results

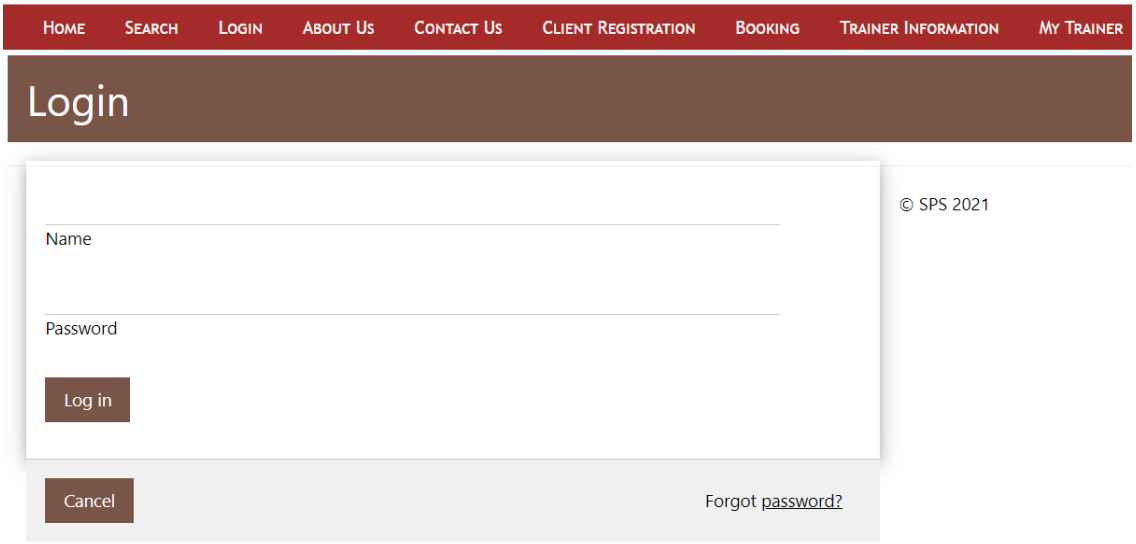


Fig 3.1 login

Figure 3.1 shows the login page. Enter your username and password, if the user is already registered in the database and if you click on the login button it will take you to the home page. If it is not registered or if you have entered a wrong password, it will refresh and remain in the same page. Go to client registration page and register yourself first.

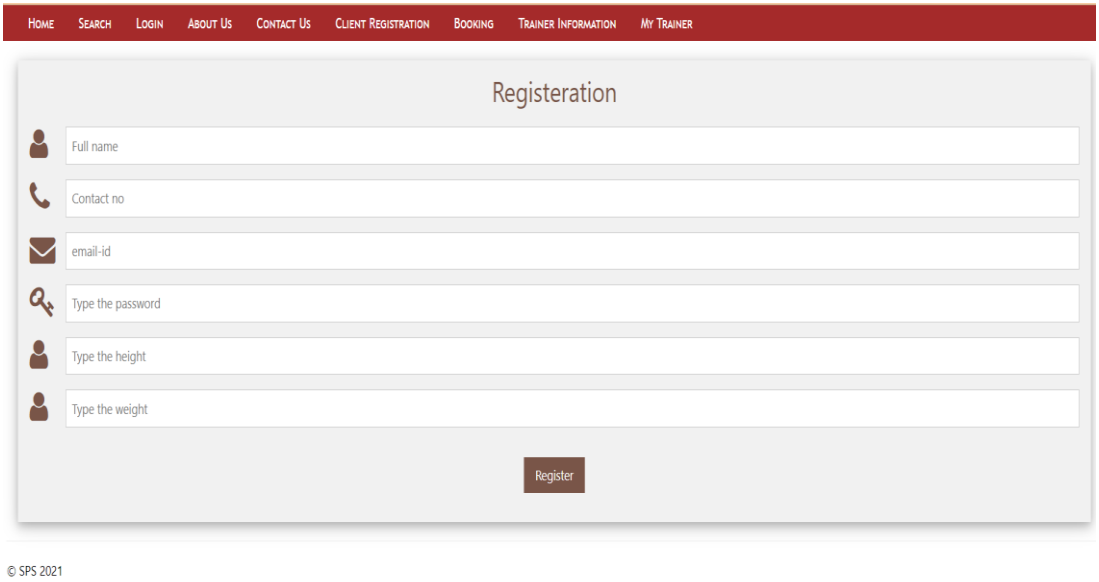


Fig 3.2 Client registration

Fig.3.2 shoes the client registration page, register yourself by filling the details once you completed filling and click on the register button it will automatically take you to the

login page. Now login with your newly created user name and password and when you click on the login button it will take you to the home page.

Fig 3.3 Home page

Fig.3.3 shows the layout of the home page in the menu we have home, search, login, about us, contact us, client registration, booking, trainer information and my trainer.

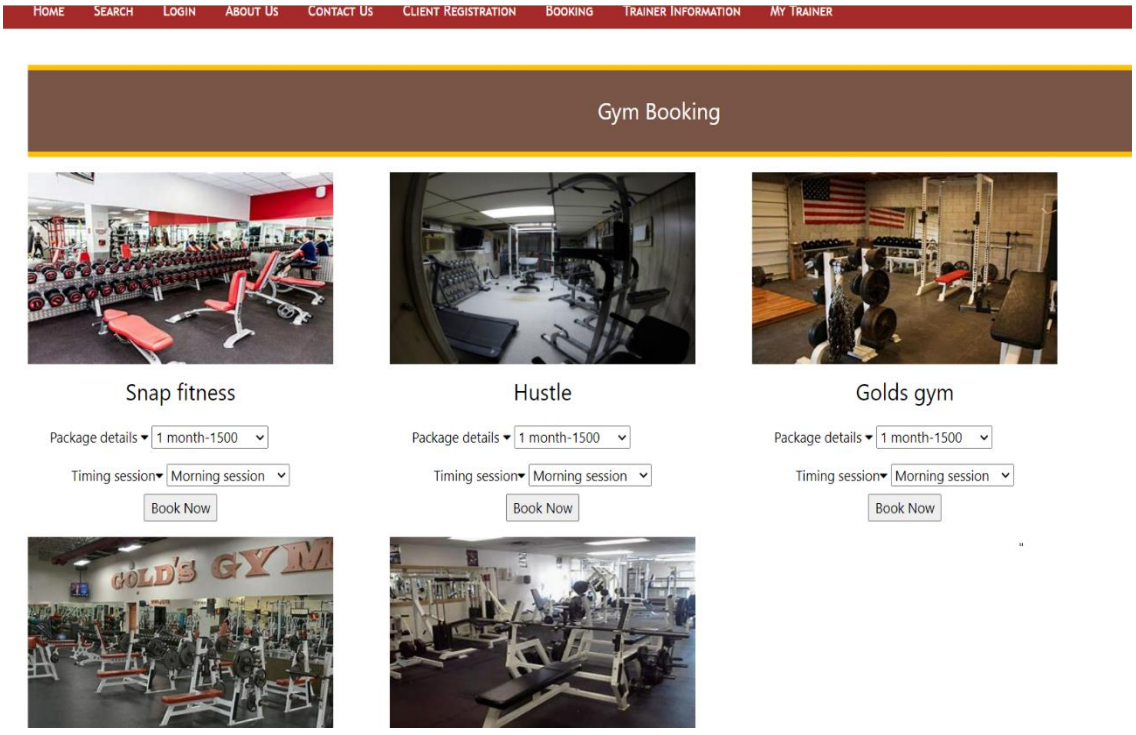


Fig 3.4 Booking

Once you have logged in you can book the gym of your choice. Go to booking, Fig.3.4 shows the booking page select the choice of your gym, select package details, different gym offers different packages and also select timing session finally when you are done choosing click on the book button.

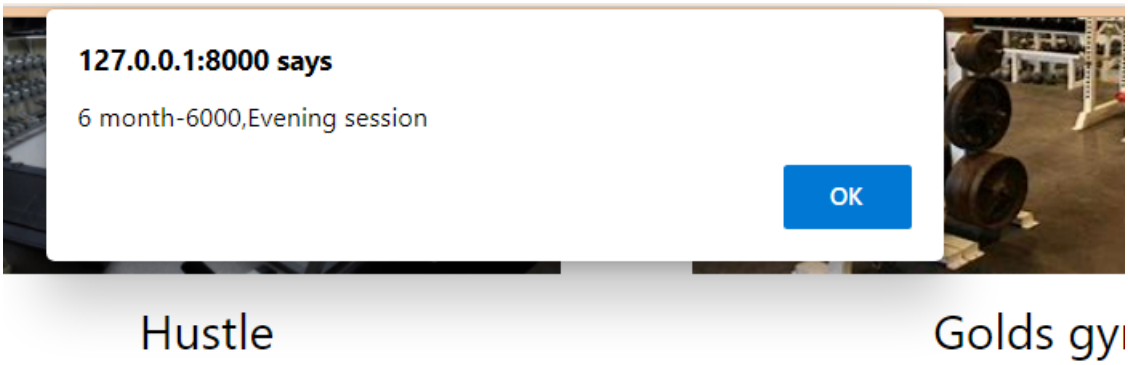


Fig 3.5 alert message 1

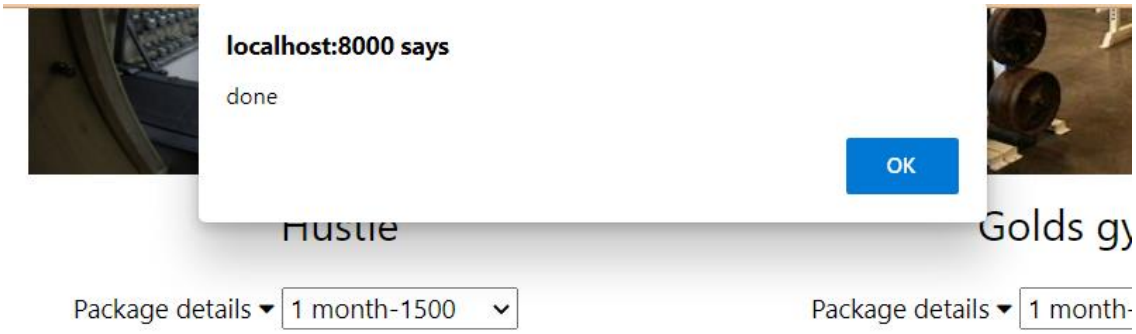


Fig 3.6 alert message 2

Once you click on the book button you get two alert messages. Fig.3.5 and Fig.3.6 shows the alert message which pops up and confirms our booking.

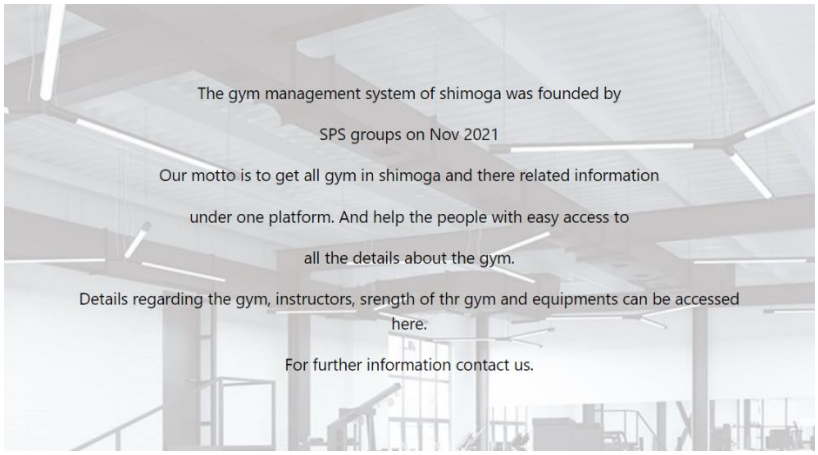


Fig 3.7 about us

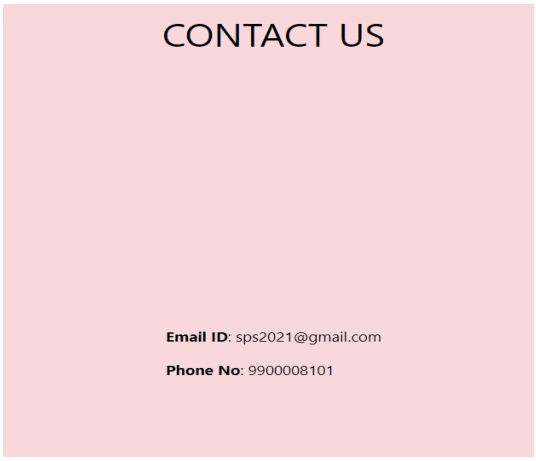


Fig. 3.8 contact us

Fig.3.7 and Fig.3.8 shows the about us and the contact us page.

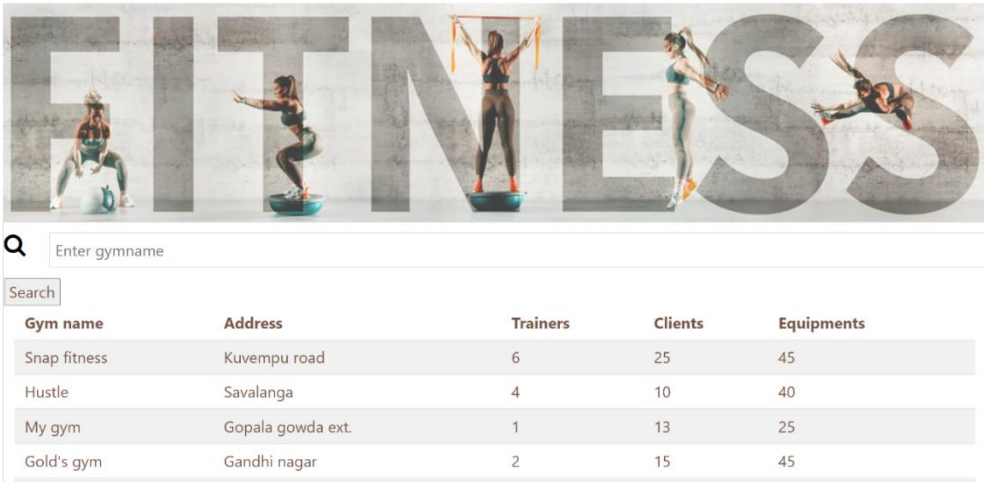


Fig 3.9 search

Fig. 3.9 shows the search page. Here you can search the information by entering the gym name. once searched address, total number of trainers, clients and equipment's present in that gym will be displayed.

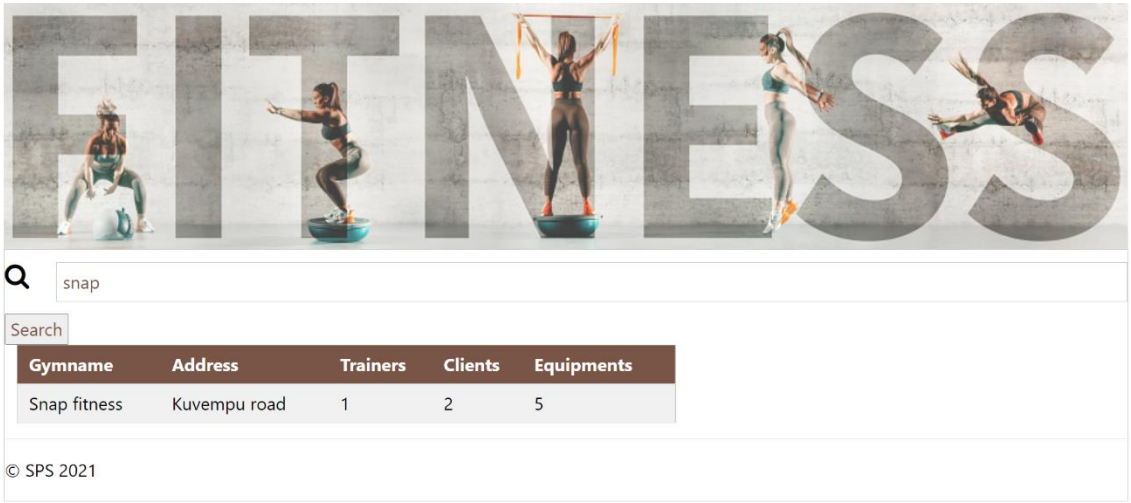


Fig 3.10 gym searched

For example, the gym searched is snap, the details of that gym will be displayed this is shown in Fig. 3.10.

Fig 3.11 search trainer

In the same way you can search the trainers by gym name it will display the details of the trainer.

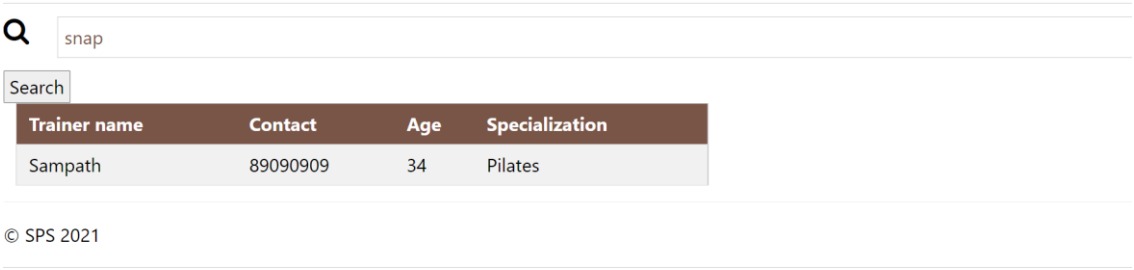


Fig 3.12 trainer searched

In this example we have searched for trainers in the gym snap, name of the trainers presents and their details will be displayed.

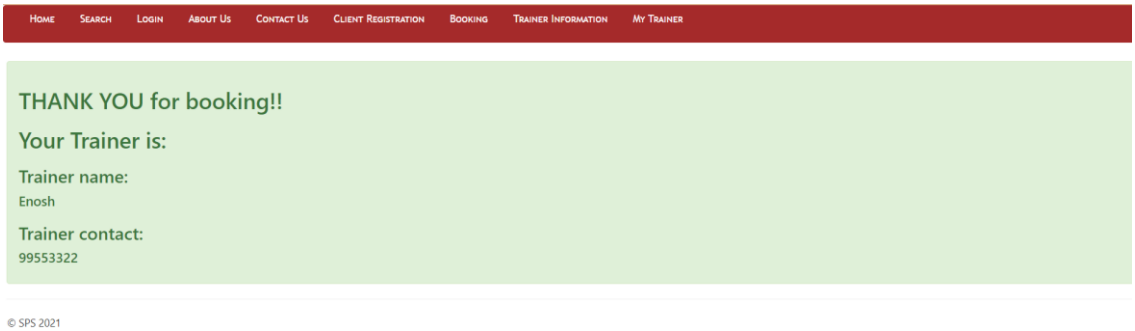


Fig 3.13 My trainer

Once you have completed your booking, the admin will allot you a trainer. Your trainer information can be viewed in my trainer.

3.2 Admin interface

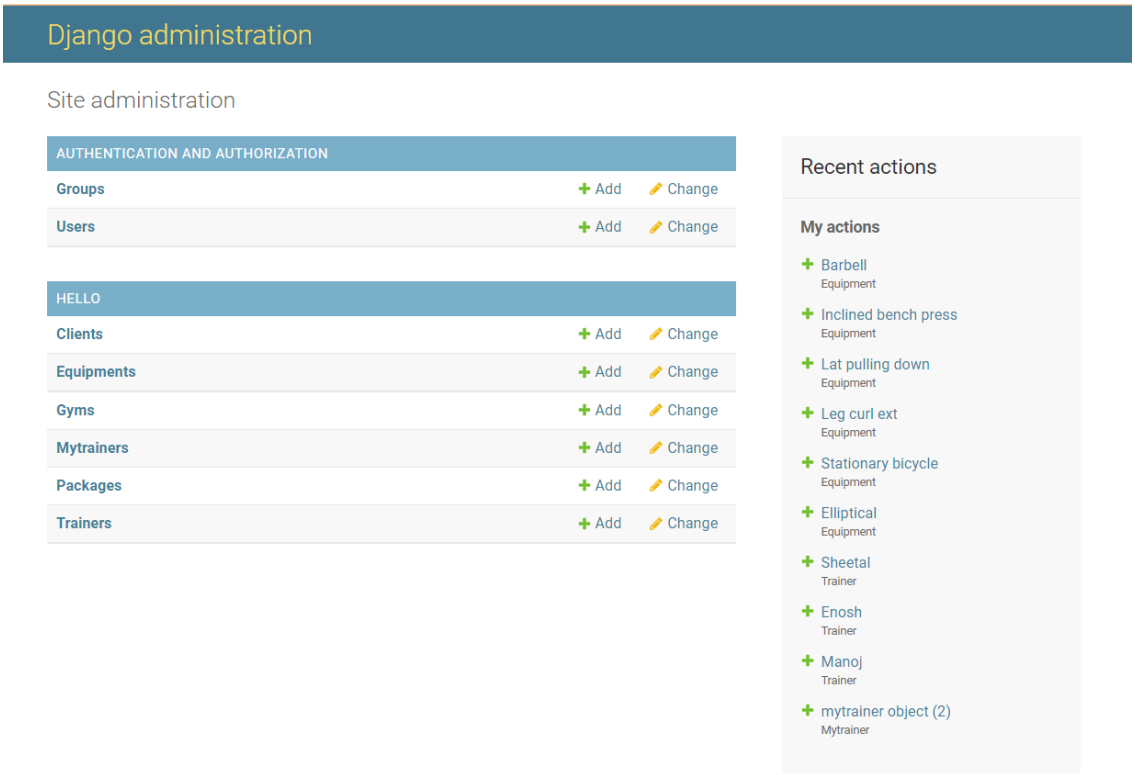


Fig 3.14 admin interface

Fig.3.14 shows the outlook of admin interface.

3.3 Backend (MySQL) snapshots

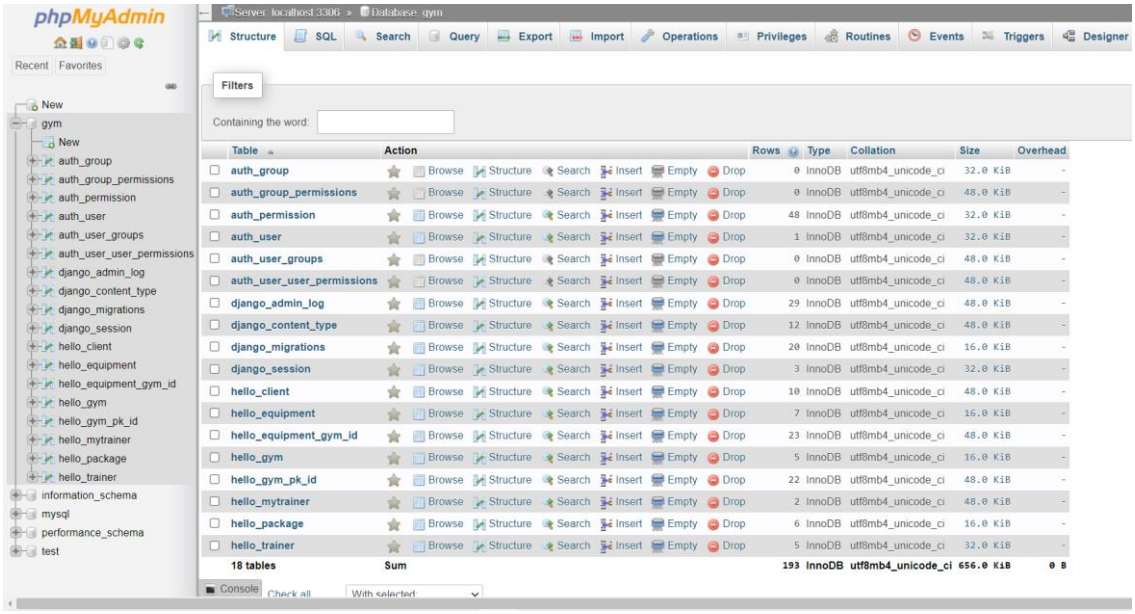
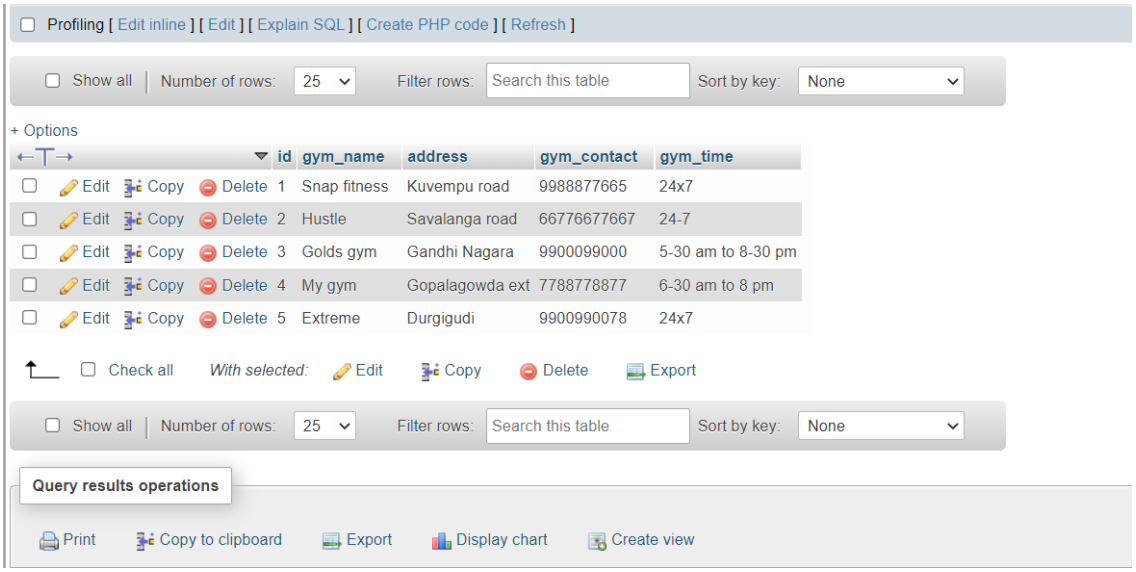


Fig. 3.15 tables in the backend

Table 3.1 hello_gym



Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	gym_name	address	gym_contact	gym_time
Edit Copy Delete	1	Snap fitness	Kuvempu road	9988877665	24x7
Edit Copy Delete	2	Hustle	Savalanga road	66776677667	24-7
Edit Copy Delete	3	Golds gym	Gandhi Nagara	9900099000	5-30 am to 8-30 pm
Edit Copy Delete	4	My gym	Gopalagowda ext	7788778877	6-30 am to 8 pm
Edit Copy Delete	5	Extreme	Durgigudi	9900990078	24x7

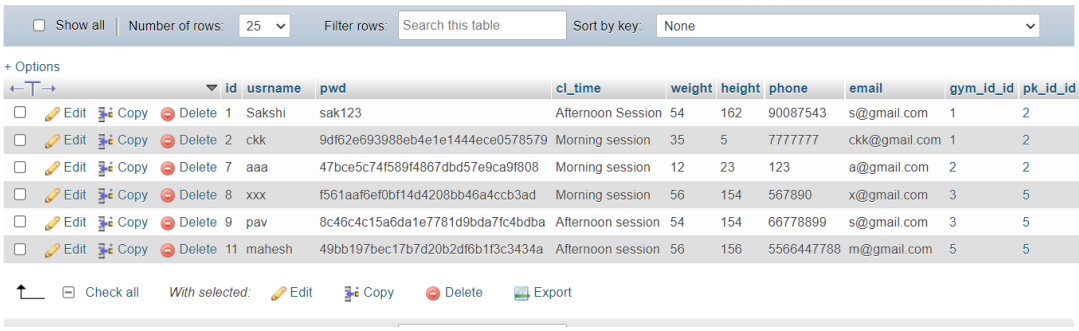
Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Table 3.2 hello_client



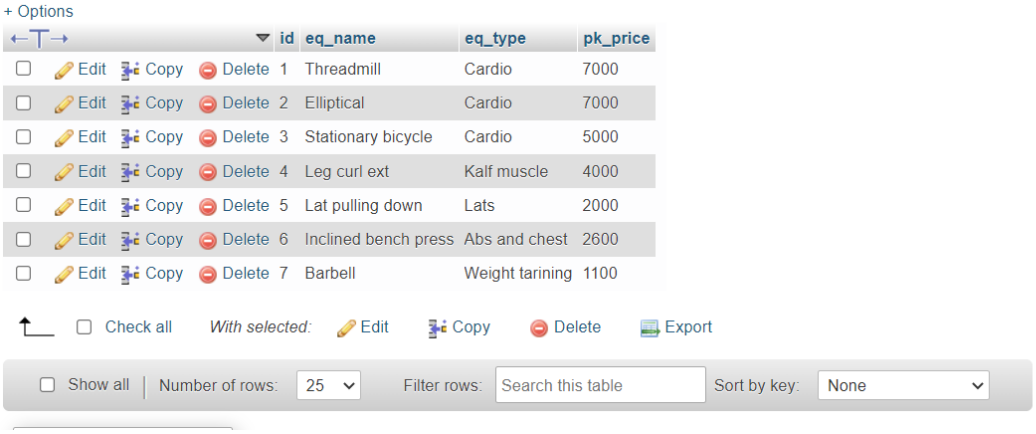
Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	id	usrname	pwd	cl_time	weight	height	phone	email	gym_id_id	pk_id_id
Edit Copy Delete	1	Sakshi	sak123	Afternoon Session	54	162	90087543	s@gmail.com	1	2
Edit Copy Delete	2	ckk	9df62e693988eb4e1e1444ece0578579	Morning session	35	5	7777777	ckk@gmail.com	1	2
Edit Copy Delete	7	aaa	47bce5c74f589f4867dbd57e9ca9f808	Morning session	12	23	123	a@gmail.com	2	2
Edit Copy Delete	8	xxx	f561aaf6ef0bf14d4208bb46a4ccb3ad	Morning session	56	154	567890	x@gmail.com	3	5
Edit Copy Delete	9	pav	8c46c4c15a6da1e7781d9bda7fc4bdba	Afternoon session	54	154	66778899	s@gmail.com	3	5
Edit Copy Delete	11	mahesh	49bb197bec17b7d20b2df6b1f3c3434a	Afternoon session	56	156	5566447788	m@gmail.com	5	5

Check all With selected: Edit Copy Delete Export

Table 3.3 hello_equipment



+ Options

	id	eq_name	eq_type	pk_price
Edit Copy Delete	1	Threadmill	Cardio	7000
Edit Copy Delete	2	Elliptical	Cardio	7000
Edit Copy Delete	3	Stationary bicycle	Cardio	5000
Edit Copy Delete	4	Leg curl ext	Kalf muscle	4000
Edit Copy Delete	5	Lat pulling down	Lats	2000
Edit Copy Delete	6	Inclined bench press	Abs and chest	2600
Edit Copy Delete	7	Barbell	Weight tarining	1100

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Table 3.4 hello_package

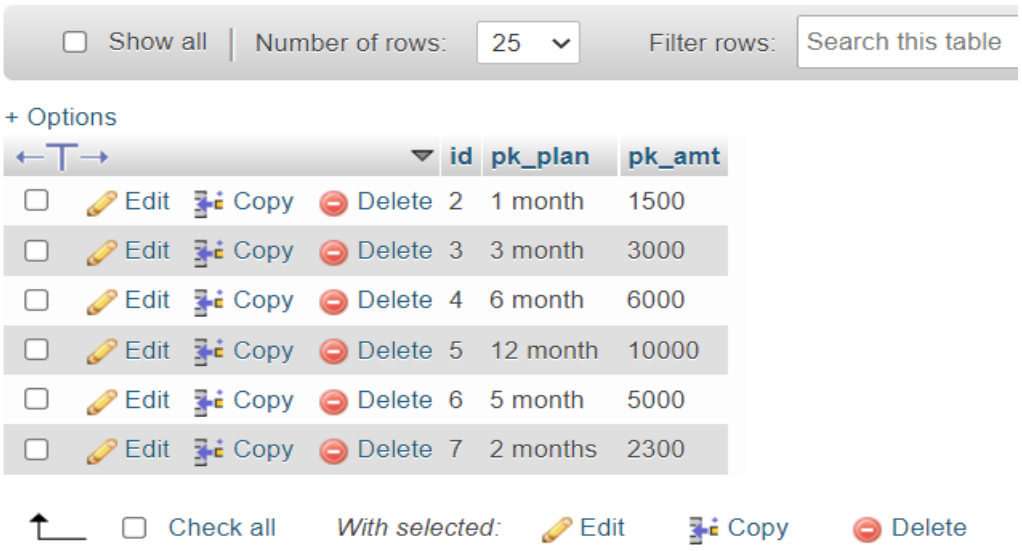


Table 3.5 hello_trainer

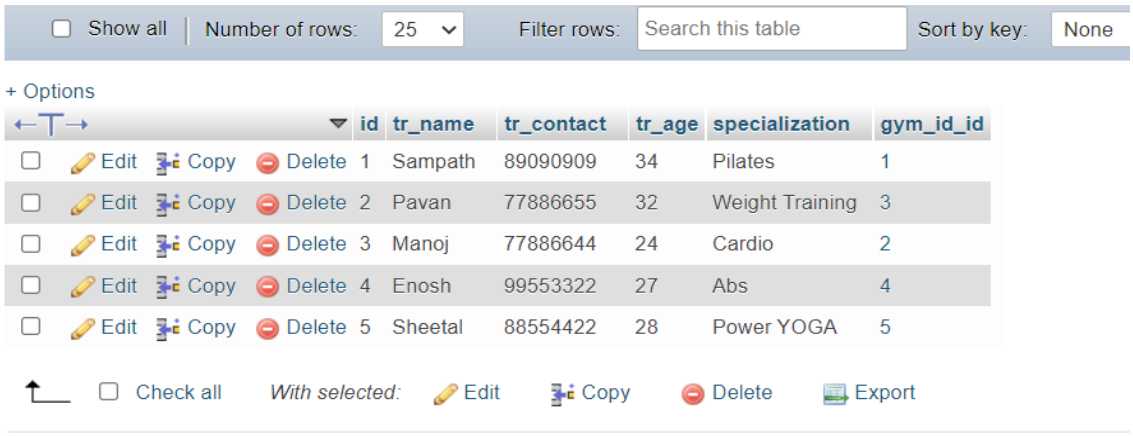


Table 3.6 hello_equipment_gym_id

+ Options

	id	equipment_id	gym_id
Edit Copy Delete	1	1	1
Edit Copy Delete	2	2	1
Edit Copy Delete	3	2	2
Edit Copy Delete	4	2	3
Edit Copy Delete	5	3	1
Edit Copy Delete	6	3	2
Edit Copy Delete	7	3	3
Edit Copy Delete	8	3	4
Edit Copy Delete	9	3	5
Edit Copy Delete	10	4	3
Edit Copy Delete	11	4	4
Edit Copy Delete	12	5	1
Edit Copy Delete	13	5	2
Edit Copy Delete	14	5	3
Edit Copy Delete	15	5	4
Edit Copy Delete	16	6	1
Edit Copy Delete	17	6	2

Table 3.7 hello_gym_pk_id

+ Options

	id	gym_id	package_id
Edit Copy Delete	1	1	2
Edit Copy Delete	2	1	3
Edit Copy Delete	3	1	4
Edit Copy Delete	4	1	5
Edit Copy Delete	5	2	2
Edit Copy Delete	6	2	4
Edit Copy Delete	7	2	5
Edit Copy Delete	8	2	7
Edit Copy Delete	9	3	2
Edit Copy Delete	10	3	5
Edit Copy Delete	11	3	6
Edit Copy Delete	12	4	2
Edit Copy Delete	13	4	3
Edit Copy Delete	14	4	4
Edit Copy Delete	15	4	5
Edit Copy Delete	16	4	6

Chapter-4: Conclusions and Future Scope

The “GYM MANAGEMENT SYSTEM” is successfully designed and developed to fulfilling the necessary requirements, as identified in the requirements analysis phase, such as the system is very much user friendly, form level validation and field level validation are performing very efficiently. The new computerized system was found to be much faster and reliable and user friendly then the existing system, the system has been designed and developed step by step and tested successfully. It eliminates the human error that are likely to creep in the kind of working in which a bulk quantity of data and calculations as to be processed. The system results in quick retrieval of information that is very vital for the progress any organization. Cost is minimized in case of stationary. Burden of manual work is reduced as whenever transaction takes place, there is a no need to record it in many places manually.

4.1 Future scope

The software has been developed in such a way that it can accept modifications and further changes. The software is very user friendly and future any changes can be done easily. Every system should allow scope for further development or enhancement. The system can be adapted for any further development. The system is so flexible to allow any modification need for the further functioning of programs. Since the objectives may be brought broad in future, the system can be easily modified accordingly, as the system has been modularized. The future expansion can be done in a concise manner in order to improve the efficiently of the system.

References

1. Antonio Mele, Django3 by Example, 3rd Edition, Pack Publishers, 2020.
2. Daniel Rubio, Beginning Django: Web Application Development and Deployment with Python, APress, 1stEdition, 2017.
3. Arun Ravindran, Django Design Patterns and Best Practices, 2nd Edition, Pack Publishers, 2020.
4. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Publications, 2016
5. https://erdplus.com/
6. https://www.w3schools.com/css/
7. Vikram Vaswani, MYSQL The Complete Reference, Tata McgrawHill, 1st Edition, 2017