Health and wellness coaching platform using Django

A Project Submitted to

University of Mumbai for Partial Completion of the degree of

Bachelor in Science (Computer Science)

Under the Faculty of Science

By

MR. Satish Shalikram Varma

Under the Guidance of

Prof. Suchita Revankar

S.I.W.S

N.R. SWAMY COLLEGE OF COMMERCE & ECONOMICS AND

SMT.THIRUMALAI COLLEGE OF SCIENCE

(Affiliated to the University of Mumbai)

Plot No. 337, Sewri – Wadala Estate,

Major R. Parmeshwaran Marg, Wadala, Mumbai – 400 031

2023-2024

Health and wellness coaching platform using Django

A Project Submitted to

University of Mumbai for Partial Completion of the degree of

Bachelor in Science (Computer Science)

Under the Faculty of Science

By

MR. Satish Shalikram Varma

Under the Guidance of

Prof. Suchita Revankar

S.I.W.S

N.R. SWAMY COLLEGE OF COMMERCE & ECONOMICS AND

SMT.THIRUMALAI COLLEGE OF SCIENCE

(Affiliated to the University of Mumbai)

Plot No. 337, Sewri – Wadala Estate,

Major R. Parmeshwaran Marg, Wadala, Mumbai – 400 031

2023-2024

S.I.W.S.

N.R.SWAMY COLLEGE OF MNERCE & ECONOMICS AND

SMT.THIRUMALAI COLLEGE OF SCIENCE

(Affiliated to University of Mumbai)

MAHARASHTRA, MUMBAI – 400031

2023-2024

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that this project report entitled "Health and Wellness Coaching Platform Using Django" is the bonafied work of " MR. SATISH SHALIKRAM VARMA" bearing Seat No.:39015 submitted in partial fulfillment of the requirements for the award of BACHELOR OF SCIENCE in COMPUTER SCIENCE from University of Mumbai for academic year 2023-2024.

Project Guide　　　　　　　　　　　　　　　　Co-Ordinator

Internal Examiner　　　　　　　　　　　　　　External Examiner

Date:　　　　　　　　　　　　　　　　　　　Collage stamp

2023-2024

DECLARATION BY LEARNER

I the undersigned Mr. Satish Shalikram Varma here by declared that the work embodied in this project work titled "Health and wellness coaching platform using Django", forms my own contribution to the research work carried out under the guidance of Prof. Suchit

a Revankar is a result of my own research work and has not previously submitted to any other University for any other degree / Diploma to this or any other University.

Wherever reference has been made to previous works of others has been clearly indicated as such and included in the bibliography.

I, here by further declared that all information of this document has been obtain and presented in accordance with academic rules and ethical conduct.

Name and Signature of the Learner

Certified by

Name and Signature of the Guidance Teacher

ACKNOWLEDGEMENT

First and foremost, I offer my sincerest gratitude to the principal and professors of my college who have supported me throughout my time here, given me valuable knowledge, Moodle and shaped me into the person I am today.

I cannot thank my family enough for bringing me up the way they did. The source behind my excellence is you.

This list would be incomplete without mentioning all of the developers and education institutes around the world that share their knowledge, work, and wisdom over the Internet.

BIBLOGRAPHY

Bibliography: While developing this project internet supports. Following are the websites referred by us which helped us in developing our project:

https://www.djangoproject.com/start/

https://getbootstrap.com/

https://docs.djangoproject.com/en/4.2/topics/db/

https://en.wikipedia.org/wiki/Django_(web_framework)

https://getbootstrap.com/docs/5.3/getting-started/introduction/

https://bootstrapmade.com/bootstrap-business-templates/

PLAGIARISM

Models.py

```
from Django. dB import models

# Create your models here.
class Contact(models.Model):
   name=models.CharField(max_length=25)
   email=models.EmailField()
   phonenumber=models.CharField(max_length=12)
   description=models.TextField()


   def _str_(self):
      return self.email


class Enrollment(models.Model):
   FullName=models.CharField(max_length=25)
   Email=models.EmailField()
   Gender=models.CharField(max_length=25)
   PhoneNumber=models.CharField(max_length=12)
   DOB=models.CharField(max_length=50)
   SelectMembershipplan=models.CharField(max_length=200)
   SelectTrainer=models.CharField(max_length=55)
   Reference=models.CharField(max_length=55)
   Address=models.TextField()
   paymentStatus=models.CharField(max_length=55,blank=True,null=True)
   Price=models.IntegerField(max_length=55,blank=True,null=True)
   DueDate=models.DateTimeField(blank=True,null=True)
 timeStamp=models.DateTimeField(auto_now_add=True,blank=True,)


   def __str__(self):
      return self.FullName
```

```
class Trainer(models.Model):
   name=models.CharField(max_length=55)
   gender=models.CharField(max_length=25)
   phone=models.CharField(max_length=25)
   salary=models.IntegerField(max_length=25)
   timeStamp=models.DateTimeField(auto_now_add=True,blank=True)
   def __str__(self):
      return self.name

class MembershipPlan(models.Model):
   plan=models.CharField(max_length=185)
   price=models.IntegerField(max_length=55)

   def __int__(self):
      return self.id



class Attendance(models.Model):
   Selectdate=models.DateTimeField(auto_now_add=True)
   phonenumber=models.CharField(max_length=15)
   Login=models.CharField(max_length=200)
   Logout=models.CharField(max_length=200)
   SelectWorkout=models.CharField(max_length=200)
   TrainedBy=models.CharField(max_length=200)
   def __int__(self):
      return self.id
```

Admin.py

```
from django.contrib import admin

from authapp.models import Contact,MembershipPlan,Enrollment,Trainer,Attendance

# Register your models here.


admin.site.register(Contact)

admin.site.register(MembershipPlan)

admin.site.register(Enrollment)

admin.site.register(Trainer)

admin.site.register(Attendance)
```

Url.py

```
from django.urls import path

from authapp import views



urlpatterns = [

    path('', views.Home,name="Home"),

    path('signup',views.signup, name="signup"),

    path('login',views.handlelogin, name="handlelogin"),

    path('logout',views.handleLogout,name="handleLogout"),

    path('contact',views.contact,name="contact"),

    path('join',views.enroll,name="enroll"),

    path('profile',views.profile,name="profile"),

    path('gallery',views.gallery,name="gallery"),

    path('attendance',views.attendance,name="attendance"),

    path('blog',views.blog,name="blog"),

    path('service',views.service,name="service"),

     path('about',views.about,name="about"),

 ]
```

ROLL NO 38035

SATISH KUMAR

AIM ;- Health and wellness coaching platform using Django

OBJECTIVE AND AIM

- The aim of creating this project is to bring every manual activity for good health at online platform.
- This website can helps the member of a gym, through this website the members can track their attendance manage their schedules.
- Trainers of the gym also can track their attendance and exercise and yoga details of members via this website

Social Integration: Enable social media integration within the software, allowing users to share their achievements, workout routines, and progress on popular platforms.
Personalized Nutrition Plans: Develop a nutrition module that generates personalized meal plans based on an individual's fitness goals, dietary restrictions, and preferences.
Interactive Exercise Library: Build an extensive exercise library with instructional videos, proper form demonstrations, and variations of exercises.

Project profile

ADMIN*:* Admin is the one who manages the whole website and has every access right to the website.

MEMBER: Members are like clients of the Gym. Member can also access many things on a website like purchase products, view attendance contact as etc.

TRAINER: Trainers are like employees of the gym. Trainers will do things like managing the workout schedule and diet chart of members.

Conclusion

1) Building a platform where people can apply and start their workout activities even at Home.
2) website has made it easier for the GYM Owner to manage the information regarding gym.

INDEX

1. Introduction

Introduction

Gym Management Software is a comprehensive gym and health club membership management system. It allows you to efficiently manage your members, their memberships, and facilitates seamless communication between you and your members. This software also encompasses a booking system, point of sale capabilities, and offers a wide range of reports to streamline the management of your club.

Our Gym Management Software is a complete program designed to cater to the needs of gyms, recreation centres, and health clubs. It effectively manages all aspects of your facility, including members, memberships, and activities.

The Gym Management Software provides numerous functions, including customer data entry, keeping records of customer fees, membership plans, and physical fitness progress. These features are aimed at delivering high-quality services to customers by gym managers. Additionally, this system also includes comprehensive information about the gym's machinery and stores data related to coaches.

Services offered by the gym are seamlessly handled by this software. The system's structure is designed to be easily understood, thanks to the Data Flow Diagram provided. Moreover, we include a Context Level Diagram and various charts in this case study to enhance clarity. For those interested in using the software, we offer a demo that includes customer detail forms and a detailed overview of the software's database.

With Gym Management Software, you can efficiently and effectively manage your gym or health club, providing excellent services to your members while simplifying your administrative tasks.

2. OBJECTIVES

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.

3.PRELIMINARY SYSTEM ANALYSIS

Preliminary Investigation

Preliminary System Analysis:

is a crucial process that involves gathering and interpreting data, identifying problems, and recommending improvements for a system? It is a problem-solving activity that demands intensive communication between system users and developers. System analysis, or study, stands as a pivotal phase in any system development process. During this phase, the system is examined in intricate detail, and the system analyst assumes the role of an interrogator, delving deep into the workings of the existing system. The system is considered holistically, and its inputs are meticulously identified.

Various techniques, such as interviews and questionnaires, are employed to conduct an exhaustive examination of the system's processes. Data collected from these sources is subject to thorough scrutiny to form a comprehensive understanding of how the system currently operates. This system, in its present state, is referred to as the "existing system." Subsequently, the existing system undergoes an in-depth analysis, where problematic areas are identified. At this point, the designer transitions into a problem solver, working to resolve the challenges that the organization faces.

The solutions to these issues are presented in the form of proposals. Each proposal is then systematically assessed against the existing system, and the most optimal one is selected. This proposal is then presented to the user for endorsement. Upon user request, the proposal is reviewed, and necessary adjustments are made. This iterative process continues until the user is completely satisfied with the proposal. In essence, preliminary study serves as the foundation for gathering and interpreting facts, providing essential information for further system studies and improvements.

Existing System:

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application.

- Lack of security of data.
- More man powers.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials

Proposed System

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

- Security of data.
- Ensure data accuracy's
- Proper control of the higher officials
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required

Proposed System:

The proposed gym management system is a software solution designed to streamline and automate various aspects of managing a fitness club. Its primary goal is to enhance the efficiency, organization, and overall experience for both gym staff and members. Here are some of the key purposes and benefits of a gym management system.

Member Management: The system allows for easy member registration, data storage, and profile management. It can track member information such as contact details, membership type and attendance.

Membership Plans and Billing: The system helps in creating and managing different membership plans with varying durations and features. It automates billing processes, sending payment reminders, and tracking dues and payments.

Attendance Tracking: The system can record and monitor member attendance using gym management system This helps in tracking member engagement and analyzing attendance patterns.

The proposed system will include following features:

Creating a database for the gym contains the information present with them on the paper in the existing system.

Advantages of Proposed System

Time saving since all the details information is stored in this system enables in significant reductions in the number of man hours takes up for, he storage and relative information as it removes the larger registration from the scene.

Accuracy and reliability

1. The online registration process eliminates manual data entry errors, ensuring accurate member information from the beginning.
2. The system's allocation of trainers is reliable, giving members confidence in their class experience and maintaining a consistent instructor schedule
3. Users Friendly, accurate and robust system
4. Security of data integrations of all functions in to one system

Remove redundancy and inconsistency

Feasibility study

It is the step-in requirement investigation whether out system feasible to organization

gym management system is essential to determine whether implementing such a system is practical, viable, and beneficial for the gym.

A. Economic feasibility
B. Technical feasibility
C. Operational feasibility

Economic feasibility:

The proposed system would be economically beneficial for the GYM center because of less of manual power and no time could be consumed and less use of stationary less people the entries.

Technical feasibility:

My Project is internally feasible as work present equipment current procedure and existing software technology. Hardware and software requirements is already available in present system. Then my project is technical feasible

Operational feasibility:

our system save time it also Provide data security there is no any loss of data.

4.Software and hardware requirements specifications

Software Requirements:

Operating System Platforms:

- Microsoft

- Linux (Ubuntu)

Front-End Technologies:

- Python3

- HTML5

- CSS

- JavaScript

Back-End Technologies:

- MySQL-based Django database

- Python3 Django framework

Web Browsers:

- Chrome (version 31 or above)

- Opera

These requirements indicate the technology stack you intend to use for your project. You have chosen Python3 and the Django framework for the backend, along with MySQL for the database. For the front-end, you are using HTML5, CSS, and JavaScript. Additionally, you have specified the compatible web browsers, including Chrome and Opera.

Hardware Requirements:

Server-Side Hardware Requirements:

1**. Server**: least one dedicated server to host your Gym Management System.

- multi-core processor (e.g., Intel Xeon or AMD Ryzen)
- 16GB to 32GB or more of RAM
- Sufficient storage space, preferably using SSDs (Solid-State Drives) for faster data access
- Network interface card (NIC) for reliable network connectivity

2**. Operating System:** server-grade operating system such as Windows Server or a Linux distribution (e.g., Ubuntu Server, CentOS) based on your software's.

3**. Database Server:** database management system like MySQL.

Client-Side Hardware Requirements:

1. Front Desk Workstations: For staff members at the front desk who manage memberships and bookings, you will need desktop or laptop workstations with:

- Dual-core or higher processor
- 4GB to 8GB or more of RAM
- Standard hard drive or SSD for storage
- Monitor, keyboard, and mouse

2. **Network Connectivity:** Ensure a stable network connection at the front desk for client workstations, preferably wired Ethernet connections for reliability. If you offer Wi-Fi access to members, ensure proper Wi-Fi infrastructure.

Limitation of Current System

1. Single-User Access:

- The system allows only one user to access it at a time, which can lead to delays and inefficiencies when multiple users need simultaneous access.

2. No Mobile App Support:

- The system lacks mobile app compatibility, restricting users from accessing it on mobile devices for added convenience.

3. Inflexible User Permissions:

- User permissions within the system are inflexible, making it challenging to customize access levels and roles based on specific organizational needs.

4. Data Import/Export Limitations:

- The system has limitations in importing and exporting data, making it cumbersome to transfer data to and from other applications or systems.

5. Lack of Multi-Language Support:

- The system does not support multiple languages, limiting its usability for non-English-speaking users or a global audience.

6. Security Vulnerabilities:

- The system may have security vulnerabilities that could expose it to potential data breaches or unauthorized access.

7. Limited User Training Resources:

- Users may struggle with the system due to a lack of comprehensive training materials or documentation.

8. No Automated Backup System:

- The system does not have an automated backup system in place, putting data at risk in case of system failures or data loss incidents.

Proposed System

The proposed gym management system is a software solution designed to streamline and automate various aspects of managing a fitness club. Its primary goal is to enhance the efficiency, organization, and overall experience for both gym staff and members. Here are some of the key purposes and benefits of a gym management system:

Member Management: The system allows for easy member registration, data storage, and profile management. It can track member information such as contact details, membership type, payment history, and attendance.

Membership Plans and Billing: The system helps in creating and managing different membership plans with varying durations and features. It automates billing processes, sending payment reminders, and tracking dues and payments.

Attendance Tracking: The system can record and monitor member attendance using gym management system This helps in tracking member engagement and analyzing attendance patterns.

The proposed system will include following features

Creating a database for the gym contains the information present with them on the paper in existing system

5.Detailed system analysis

STACK HOLDER

1. Gym Owners/Managers:

- Gym owners or managers are primary stakeholders responsible for overseeing the entire project. They define the system's goals, objectives, and budget. They play a pivotal role in decision-making, ensuring the system aligns with the gym's business strategy.

2. Front Desk Staff:

- Front desk staff members are directly involved in daily gym operations. They use the system for member check-ins, class scheduling, and managing member information. Their feedback and training are essential for a successful system implementation.

3. Fitness Instructors and Coaches:

- Instructors and coaches use the system for scheduling classes, booking appointments, and tracking member progress. They need tools that streamline their interactions with members and optimize their schedules.

4. Members:

- Gym members are end-users of the system. They use it to book classes, check-in, track their fitness progress, and communicate with gym staff. Member satisfaction is critical for system success.

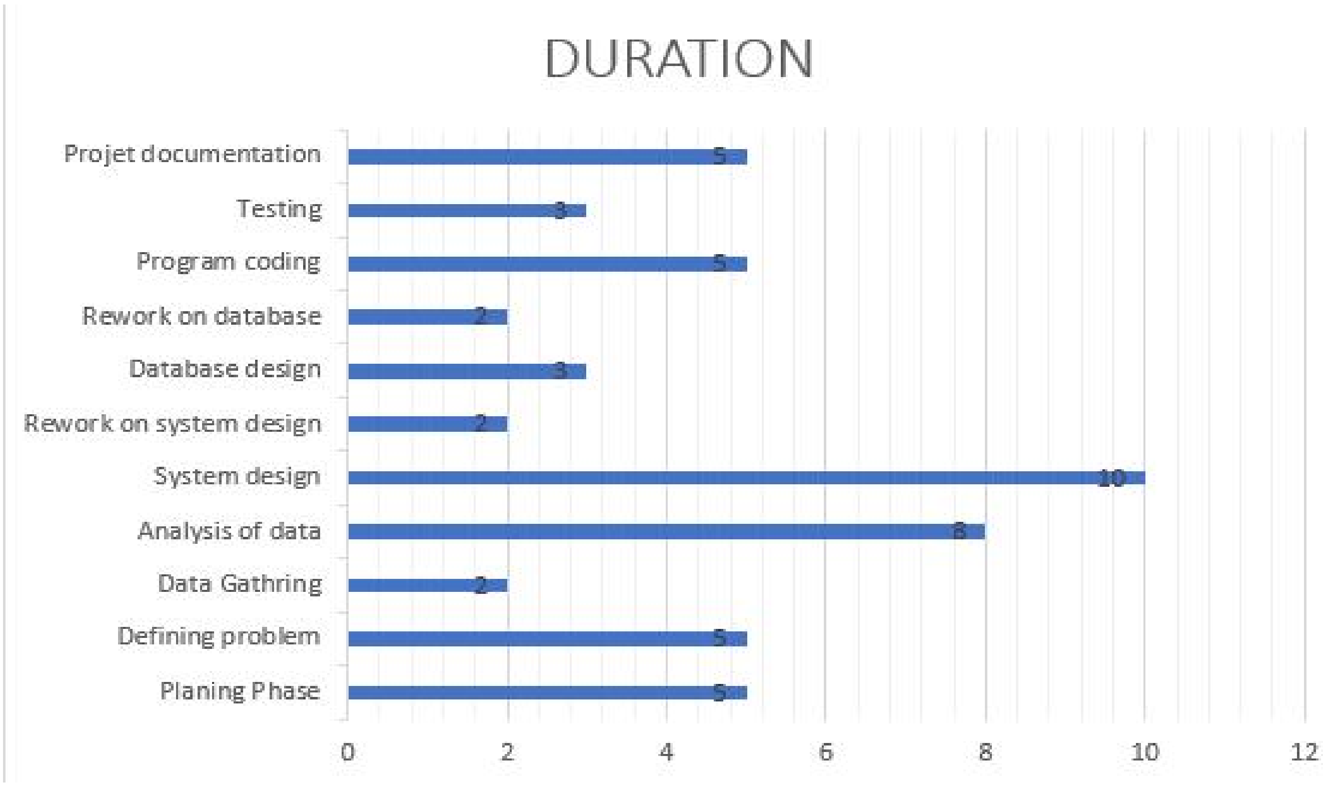
5. IT Team/Developers:

- The IT team or software developers are responsible for system development, maintenance, and troubleshooting. They work closely with other stakeholders to ensure the system meets requirements and runs smoothly.

6. System Administrators:

- System administrators manage server infrastructure, databases, backups, and security. They ensure the system's reliability, performance, and data integrity.

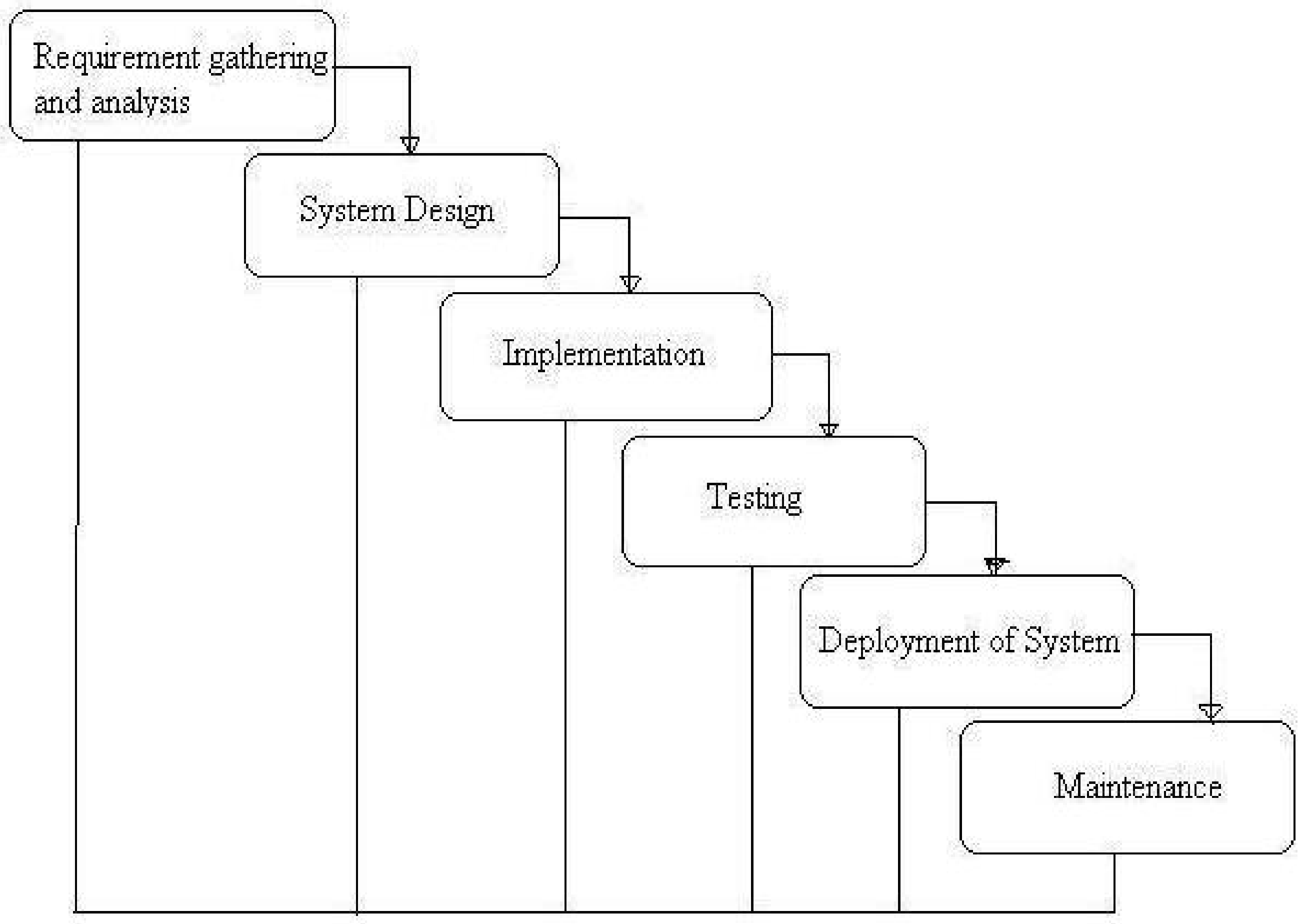
GANTT CHART


DURATION
Projet documentation
Testing
Program coding
Rework on database
Database design
Rework on system design
System design
Analysis of data
Data Gathring
Defining problem
Planing Phase
5
3
5
2
3
2
10
8
2
5
5
0
2
4
6
8
10
12

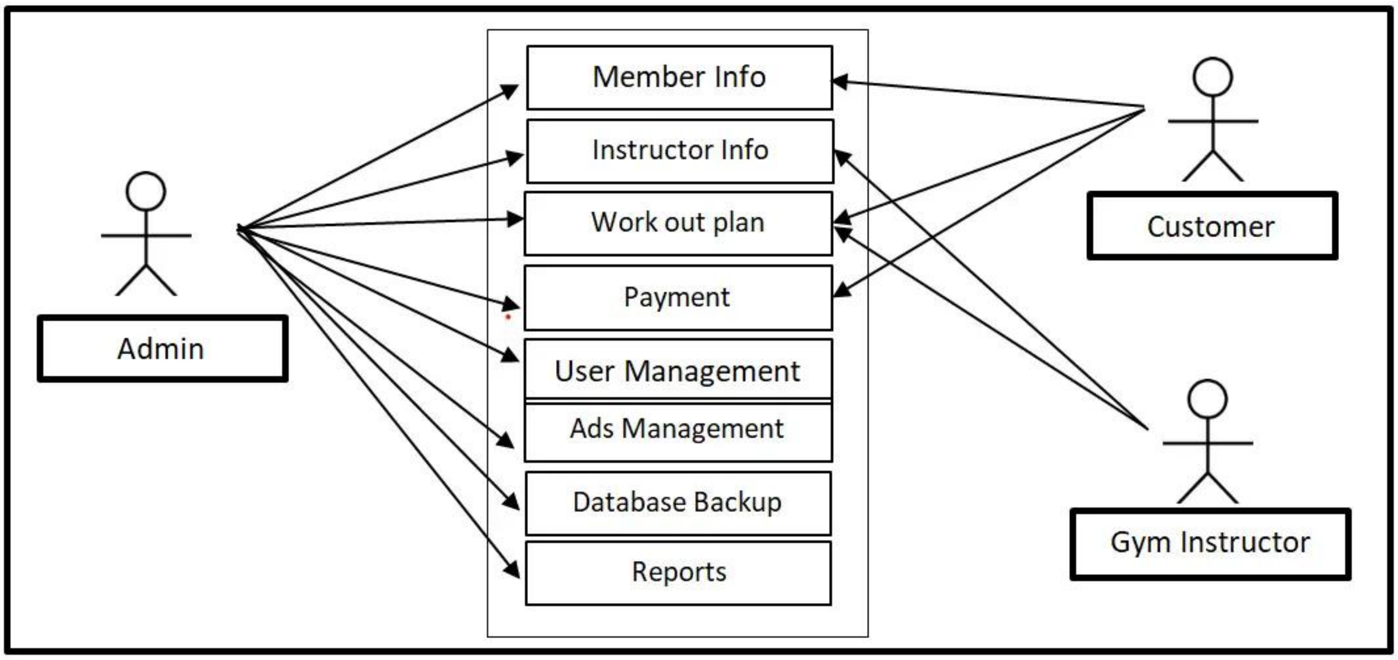
Event table

SR NO	EVENT	TRIGER	SOURCE	ACTIVE	RESPONSE	DESTINATION
1	LOGIN	LOGIN SYSTEM	ADMIN	LOGIN SUCCESSFULLY	LOGIN INSERTED	ADMIN
2	Assign TRAINER	Assign TRAINER	ADMIN	TRAINER Assigned	RECORD SAVE	ADMIN
3	ADD MEMBERS	JOIN MEMBERS	ADMIN	REGISTRATION SUCCESSFULLLY	RECORD SAVE	ADMIN
4	ADD / DELETE MEMBERS	ADD / DELETE MEMBERS	ADMIN	ADD / DELETE SUCCESSFULLLY	RECORD SAVE	ADMIN
5	ADD / DELETE TRAINERS	ADD / DELETE TRAINER	ADMIN	ADD / DELETE SUCCESSFULLLY	RECORD SAVE	ADMIN
6	UPDTAE REGISTERED DATA	UPDATE FINAL DATA	ADMIN	UPDATE FINAL DATA	RECORD SAVE	USERS /ADMIN

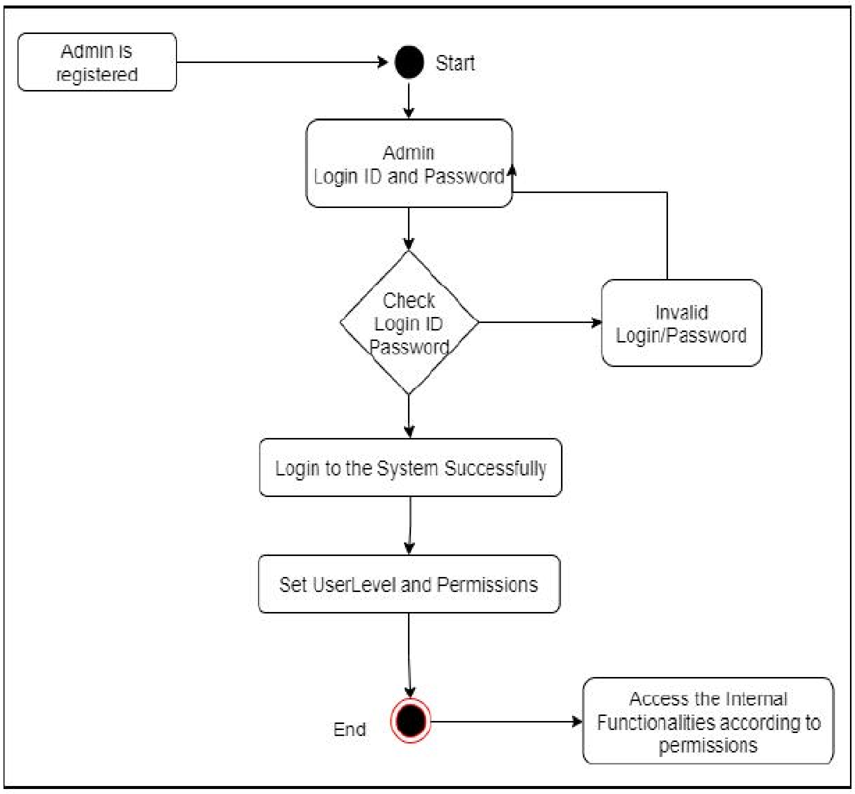
WATERFALL MODEL



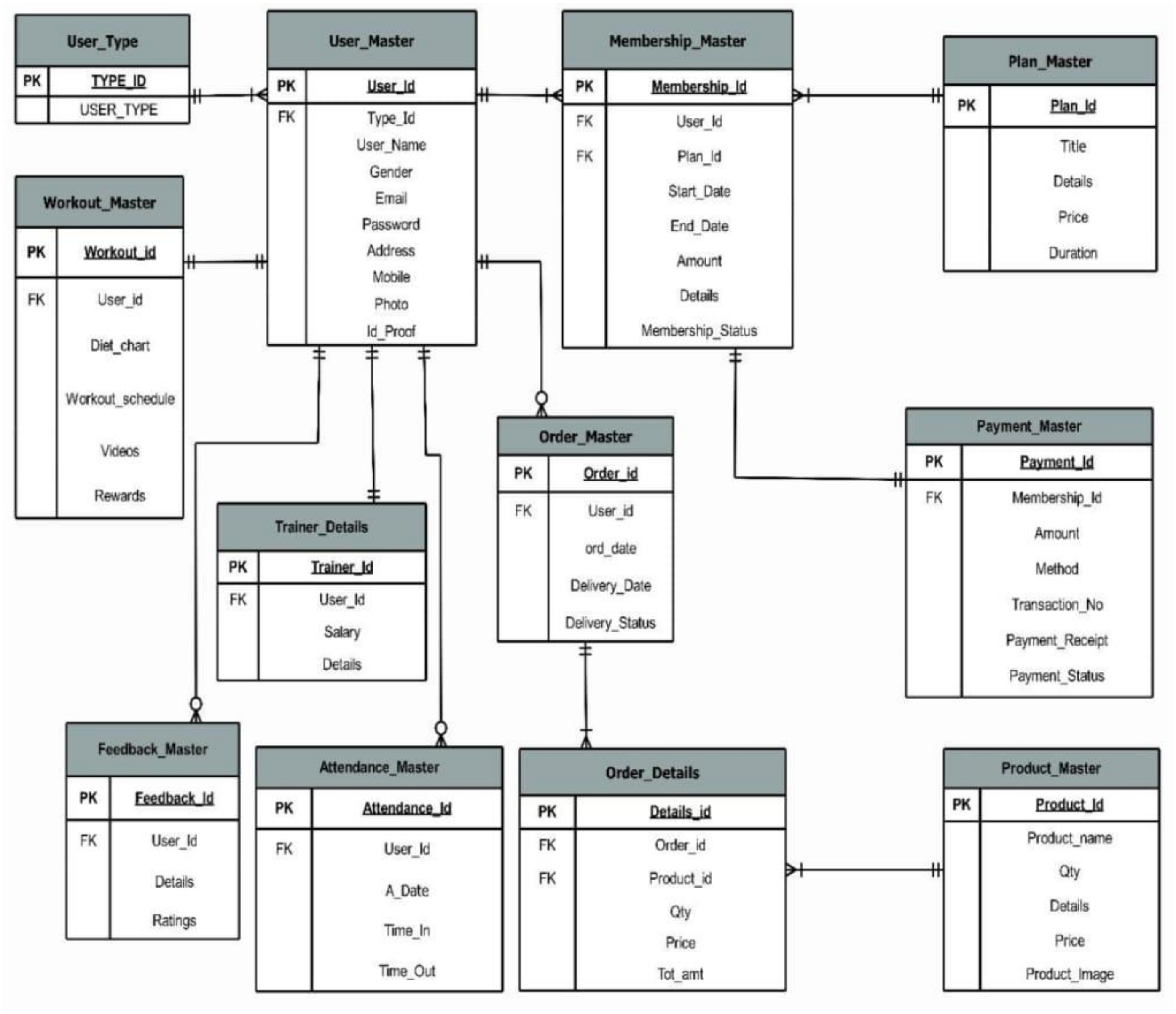
USE CASE DIGRAM



ACTIVITY DIAGRAM


Admin is registered
Start
Admin Login ID and Password
Check Login ID Password
Invalid Login/Password
Login to the System Successfully
Set UserLevel and Permissions
End
Access the Internal Functionalities according to permissions

ER DIGRAM



User_Type
PK TYPE_ID
USER_TYPE
User_Master
PK User_Id
FK Type_Id
User_Name
Gender
Email
Password
Address
Mobile
Photo
Id_Proof
Membership_Master
PK Membership_Id
FK User_Id
FK Plan_Id
Start_Date
End_Date
Amount
Details
Membership_Status
Plan_Master
PK Plan_Id
Title
Details
Price
Duration
Workout_Master
PK Workout_id
FK User_id
Diet_chart
Workout_schedule
Videos
Rewards
Order_Master
PK Order_id
FK User_id
ord_date
Delivery_Date
Delivery_Status
Payment_Master
PK Payment_Id
FK Membership_Id
Amount
Method
Transaction_No
Payment_Receipt
Payment_Status
Trainer_Details
PK Trainer_Id
FK User_Id
Salary
Details
Feedback_Master
PK Feedback_Id
FK User_Id
Details
Ratings
Attendance_Master
PK Attendance_Id
FK User_Id
A_Date
Time_In
Time_Out
Order_Details
PK Details_id
FK Order_id
FK Product_id
Qty
Price
Tot_amt
Product_Master
PK Product_Id
Product_name
Qty
Details
Price
Product_Image

6.Input Output and Screen Layout

INPUT PROGRAMS: -

Models.py

```
from Django. dB import models

# Create your models here.
class Contact(models.Model):
   name=models.CharField(max_length=25)
   email=models.EmailField()
   phonenumber=models.CharField(max_length=12)
   description=models.TextField()


   def _str_(self):
      return self.email


class Enrollment(models.Model):
   FullName=models.CharField(max_length=25)
   Email=models.EmailField()
   Gender=models.CharField(max_length=25)
   PhoneNumber=models.CharField(max_length=12)
   DOB=models.CharField(max_length=50)
   SelectMembershipplan=models.CharField(max_length=200)
   SelectTrainer=models.CharField(max_length=55)
   Reference=models.CharField(max_length=55)
   Address=models.TextField()
   paymentStatus=models.CharField(max_length=55,blank=True,null=True)
   Price=models.IntegerField(max_length=55,blank=True,null=True)
   DueDate=models.DateTimeField(blank=True,null=True)
 timeStamp=models.DateTimeField(auto_now_add=True,blank=True,)


   def __str__(self):
      return self.FullName
```

```
class Trainer(models.Model):
   name=models.CharField(max_length=55)
   gender=models.CharField(max_length=25)
   phone=models.CharField(max_length=25)
   salary=models.IntegerField(max_length=25)
   timeStamp=models.DateTimeField(auto_now_add=True,blank=True)
   def __str__(self):
      return self.name

class MembershipPlan(models.Model):
   plan=models.CharField(max_length=185)
   price=models.IntegerField(max_length=55)

   def __int__(self):
      return self.id



class Attendance(models.Model):
   Selectdate=models.DateTimeField(auto_now_add=True)
   phonenumber=models.CharField(max_length=15)
   Login=models.CharField(max_length=200)
   Logout=models.CharField(max_length=200)
   SelectWorkout=models.CharField(max_length=200)
   TrainedBy=models.CharField(max_length=200)
   def __int__(self):
      return self.id
```

Admin.py

```
from django.contrib import admin

from authapp.models import Contact,MembershipPlan,Enrollment,Trainer,Attendance

# Register your models here.



admin.site.register(Contact)

admin.site.register(MembershipPlan)

admin.site.register(Enrollment)

admin.site.register(Trainer)

admin.site.register(Attendance)
```

Url.py

```
from django.urls import path

from authapp import views



urlpatterns = [

   path('', views.Home,name="Home"),

   path('signup',views.signup, name="signup"),

   path('login',views.handlelogin, name="handlelogin"),

   path('logout',views.handleLogout,name="handleLogout"),

   path('contact',views.contact,name="contact"),

   path('join',views.enroll,name="enroll"),

   path('profile',views.profile,name="profile"),

   path('gallery',views.gallery,name="gallery"),

   path('attendance',views.attendance,name="attendance"),

   path('blog',views.blog,name="blog"),

   path('service',views.service,name="service"),

    path('about',views.about,name="about"),

 ]
```

View.py

```
from django.shortcuts import render,redirect

from django.contrib import messages

from django.contrib.auth.models import User

from django.contrib.auth import authenticate,login,logout

from authapp.models import Contact,MembershipPlan,Trainer,Enrollment,Attendance

# Create your views here.

def Home(request):

    return render(request,"index.html")


def blog(request):

    return render(request,"blog.html")


def about(request):

    return render(request,"about.html")




def service(request):

    return render(request,"service.html")


def signup(request):

    if request.method=="POST":

        username=request.POST.get('usernumber')

        email=request.POST.get('email')

        pass1=request.POST.get('pass1')

        pass2=request.POST.get('pass2')
```

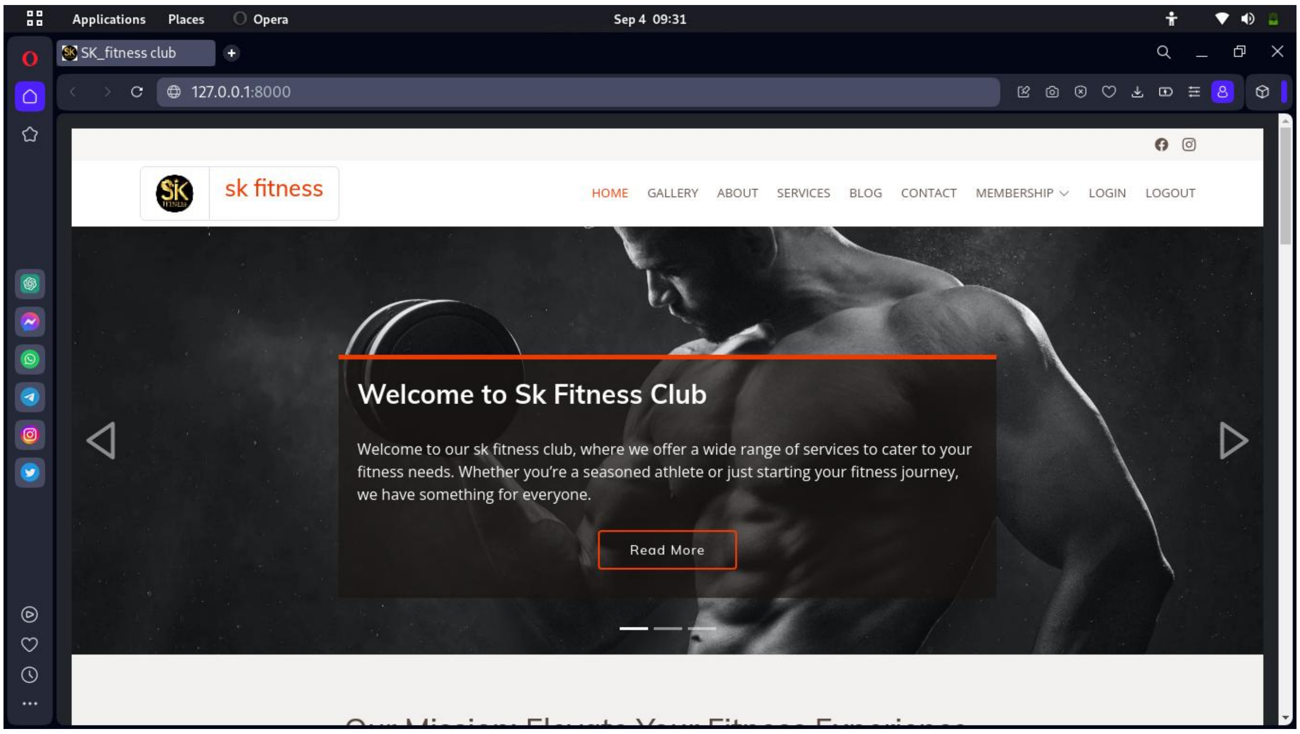
```
if len(username)>10 or len(username)<10:
      messages.info(request,"Phone Number Must be 10 Digits")
      return redirect('/signup')



   if pass1!=pass2:
      messages.info(request,"Password is not Matching")
      return redirect('/signup')



   try:
      if User.objects.get(username=username):
         messages.warning(request,"Phone Number is Taken")
         return redirect('/signup')

   except Exception as identifier:
      pass



   try:
      if User.objects.get(email=email):
         messages.warning(request,"Email is Taken")
         return redirect('/signup')

   except Exception as identifier:
      pass



   myuser=User.objects.create_user(username,email,pass1)
   myuser.save()
```

```
        messages.success(request,"User is Created Please Login")
        return redirect('/login')

    return render(request,"signup.html")



def handlelogin(request):
    if request.method=="POST":
        username=request.POST.get('usernumber')
        pass1=request.POST.get('pass1')
        myuser=authenticate(username=username,password=pass1)
        if myuser is not None:
            login(request,myuser)
            messages.success(request,"Login Successful")
            return redirect('/')
        else:
            messages.error(request,"Invalid Credentials")
            return redirect('/login')



    return render(request,"handlelogin.html")


def handleLogout(request):
    logout(request)
    messages.success(request,"Logout Success")
    return redirect('/login')


def contact(request):
    if request.method=="POST":
        name=request.POST.get('fullname')
        email=request.POST.get('email')
        number=request.POST.get('num')
        desc=request.POST.get('desc')

        myquery=Contact(name=name,email=email,phonenumber=number,description=desc)

        myquery.save()

        messages.info(request,"Thanks for Contacting us we will get back you soon")

        return redirect('/contact')


    return render(request,"contact.html")


def enroll(request):

    if not request.user.is_authenticated:

        messages.warning(request,"Please Login and Try Again")

        return redirect('/login')


    Membership=MembershipPlan.objects.all()

    SelectTrainer=Trainer.objects.all()

    context={"Membership":Membership,"SelectTrainer":SelectTrainer}

    if request.method=="POST":

        FullName=request.POST.get('FullName')

        email=request.POST.get('email')

        PhoneNumber=request.POST.get('PhoneNumber')

        gender=request.POST.get('gender')

        member=request.POST.get('member')

        trainer=request.POST.get('trainer')

query=Enrollment(SelectMembershipplan=member,PhoneNumber=PhoneNumber,FullName=FullName,Email=email,
Gender=gender,SelectTrainer=trainer)

        query.save()

        messages.success(request,"Thanks For Enrollment")

        return redirect('/join')

    return render(request,"enroll.html",context)
```

```
def profile(request):
    if not request.user.is_authenticated:
        messages.warning(request,"Please Login and Try Again")
        return redirect('/login')
    user_phone=request.user
    posts=Enrollment.objects.filter(PhoneNumber=user_phone)
    print(posts)
    context={"posts":posts}
    return render(request,"profile.html",context)


def gallery(request):
    return render(request,"gallery.html")


def attendance(request):
    if not request.user.is_authenticated:
        messages.warning(request,"Please Login and Try Again")
        return redirect('/login')
    SelectTrainer=Trainer.objects.all()
    context={"SelectTrainer":SelectTrainer}
    if request.method=="POST":
        phonenumber=request.POST.get('PhoneNumber')
        Login=request.POST.get('logintime')
        Logout=request.POST.get('loginout')
        SelectWorkout=request.POST.get('workout')
        TrainedBy=request.POST.get('trainer')

query=Attendance(phonenumber=phonenumber,Login=Login,Logout=Logout,SelectWorkout=SelectWorkout,Trained
By=TrainedBy)
        query.save()
        messages.warning(request,"Attendace Applied Success")
        return redirect('/attendance')
    return render(request,"attendance.html",context)
```

Templates output

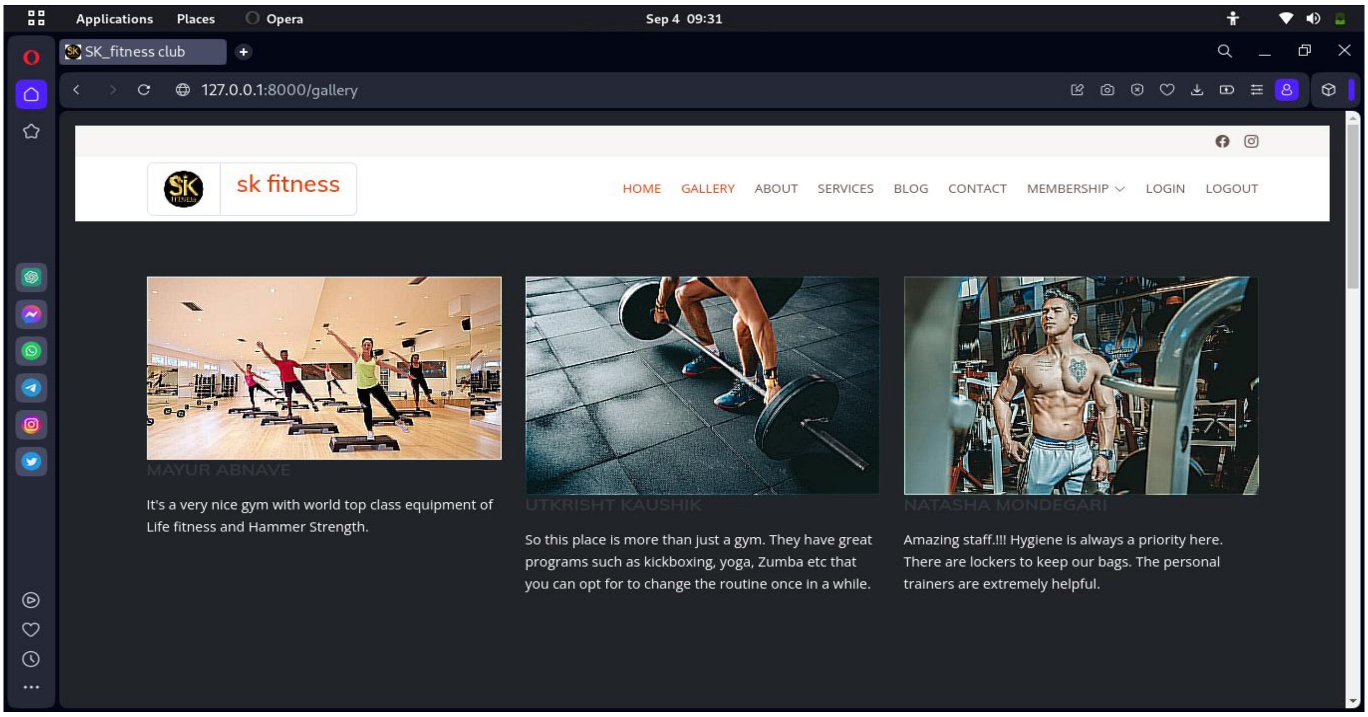
Home.html



Welcome to Sk Fitness Club

Welcome to our sk fitness club, where we offer a wide range of services to cater to your fitness needs. Whether you're a seasoned athlete or just starting your fitness journey, we have something for everyone.

Gallery.html



MAYUR ABNAVE

It's a very nice gym with world top class equipment of Life fitness and Hammer Strength.

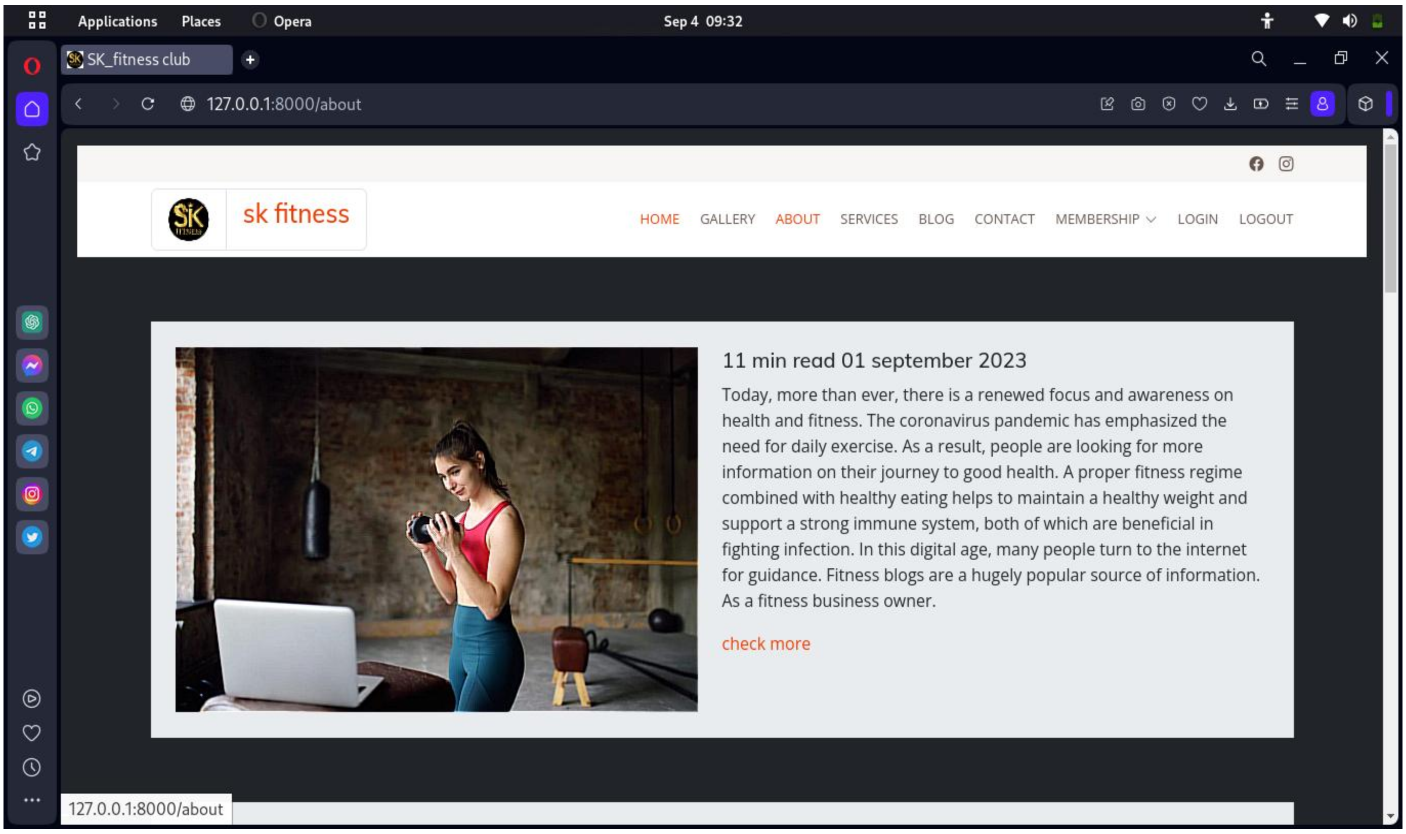
UTKRISHT KAUSHIK

So this place is more than just a gym. They have great programs such as kickboxing, yoga, Zumba etc that you can opt for to change the routine once in a while.

NATASHA MONDEGARI

Amazing staff.!!! Hygiene is always a priority here. There are lockers to keep our bags. The personal trainers are extremely helpful.

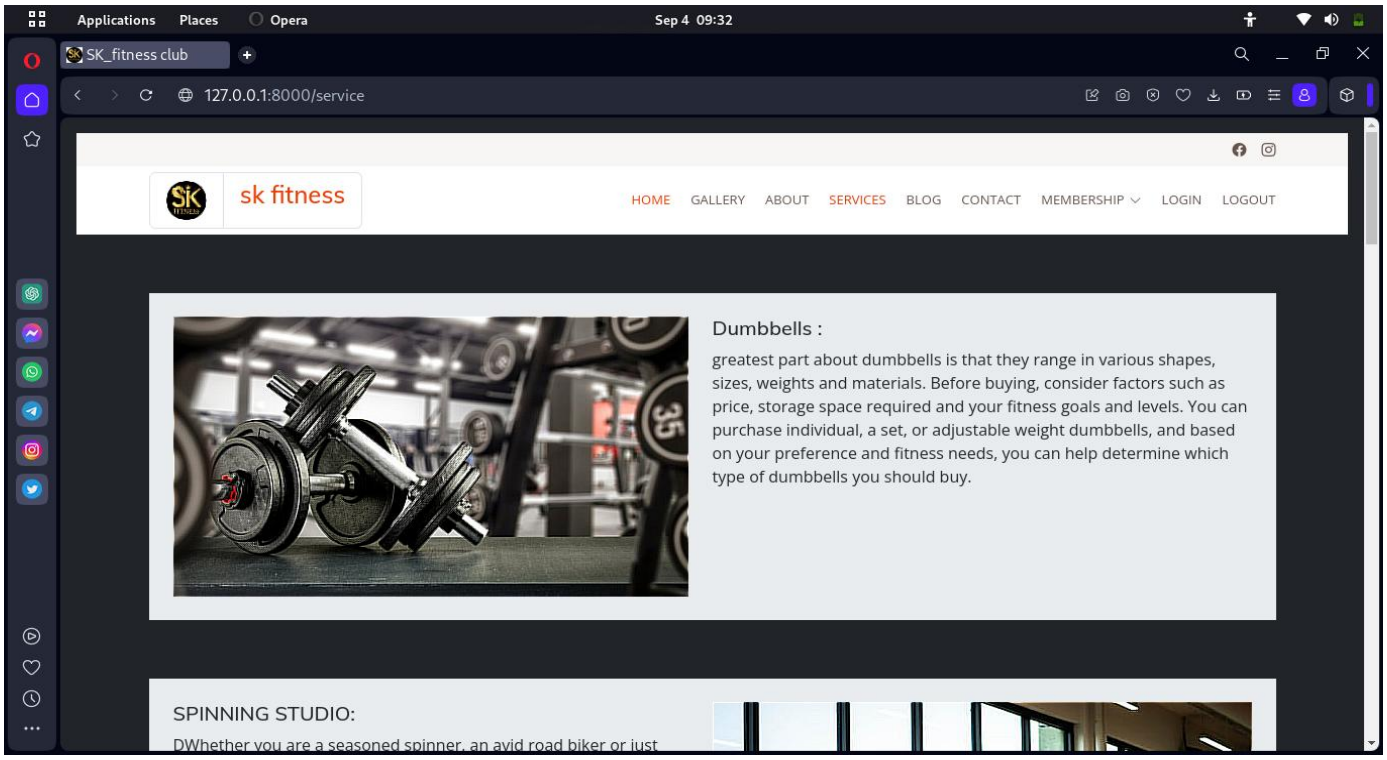
About.html



11 min read 01 september 2023

Today, more than ever, there is a renewed focus and awareness on health and fitness. The coronavirus pandemic has emphasized the need for daily exercise. As a result, people are looking for more information on their journey to good health. A proper fitness regime combined with healthy eating helps to maintain a healthy weight and support a strong immune system, both of which are beneficial in fighting infection. In this digital age, many people turn to the internet for guidance. Fitness blogs are a hugely popular source of information. As a fitness business owner.

Service.html



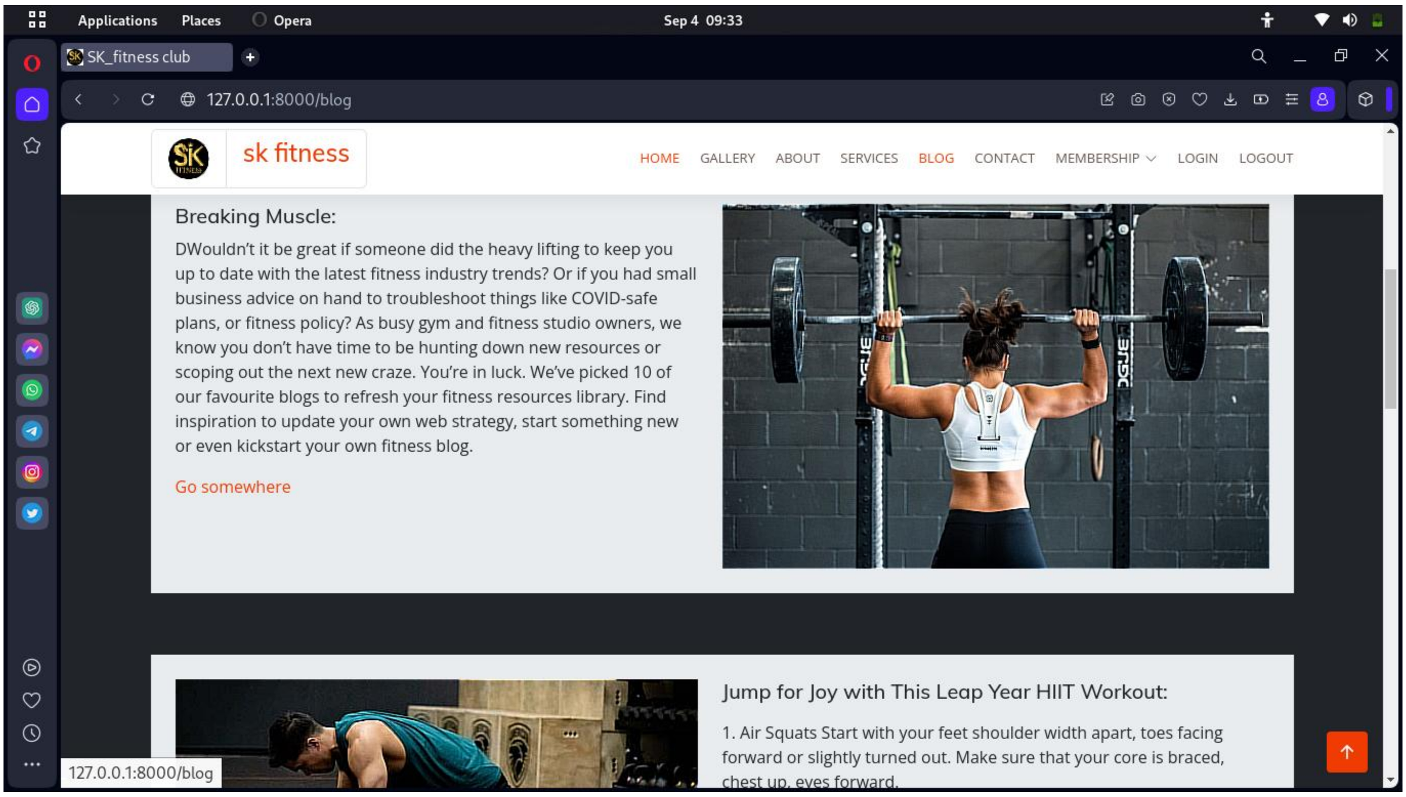
Dumbbells :

greatest part about dumbbells is that they range in various shapes, sizes, weights and materials. Before buying, consider factors such as price, storage space required and your fitness goals and levels. You can purchase individual, a set, or adjustable weight dumbbells, and based on your preference and fitness needs, you can help determine which type of dumbbells you should buy.

SPINNING STUDIO:

DWhether you are a seasoned spinner, an avid road biker or just

Blogs.html



Breaking Muscle:

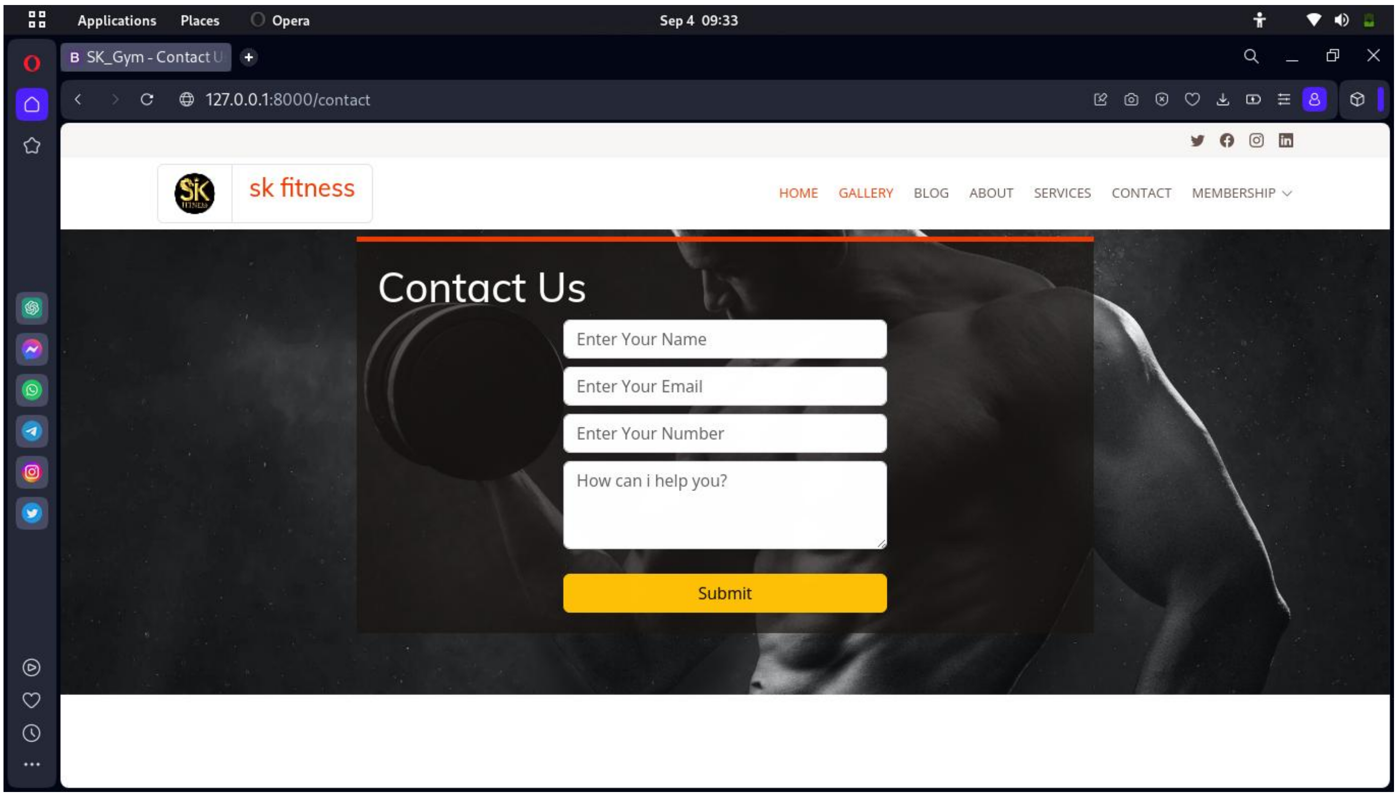
DWouldn't it be great if someone did the heavy lifting to keep you up to date with the latest fitness industry trends? Or if you had small business advice on hand to troubleshoot things like COVID-safe plans, or fitness policy? As busy gym and fitness studio owners, we know you don't have time to be hunting down new resources or scoping out the next new craze. You're in luck. We've picked 10 of our favourite blogs to refresh your fitness resources library. Find inspiration to update your own web strategy, start something new or even kickstart your own fitness blog.

Go somewhere

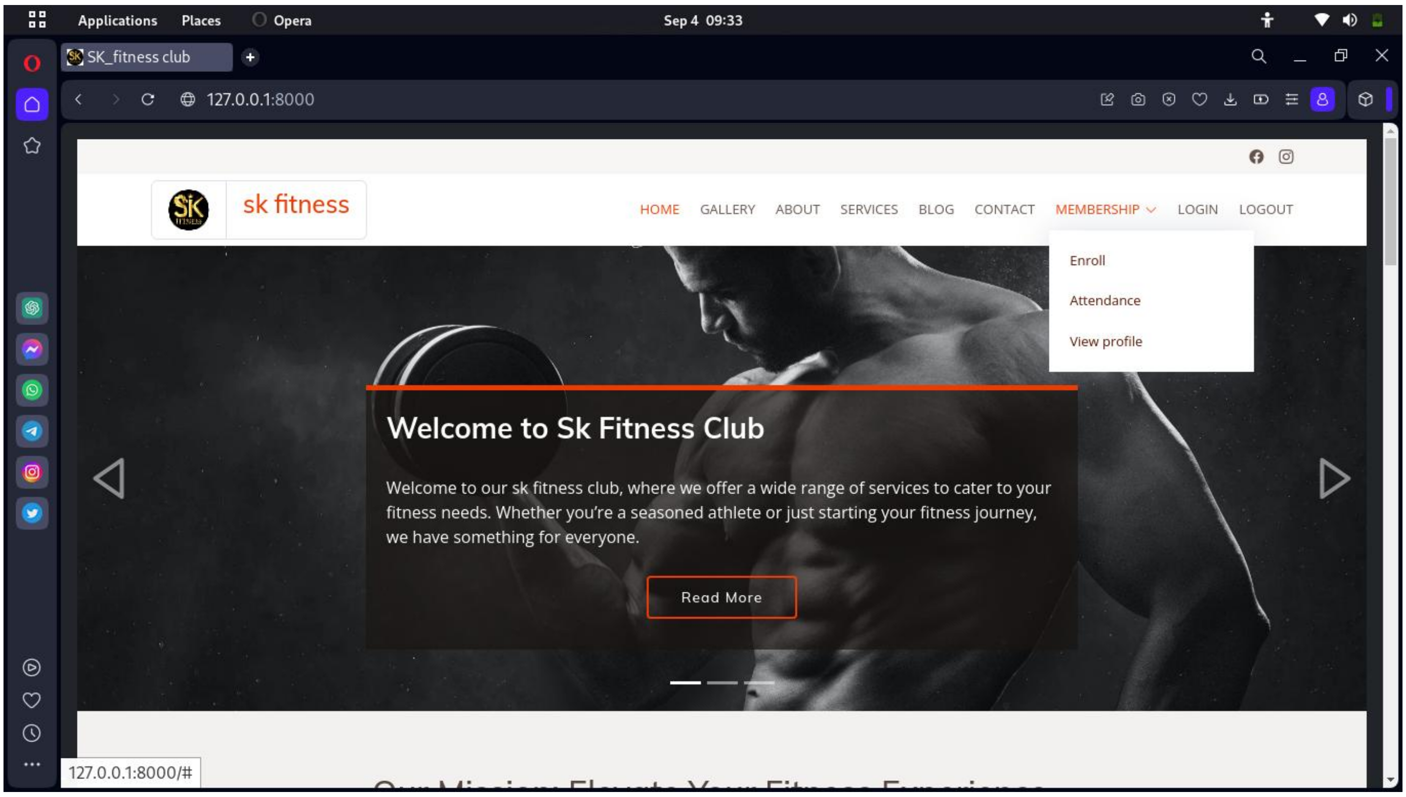
Jump for Joy with This Leap Year HIIT Workout:

1. Air Squats Start with your feet shoulder width apart, toes facing forward or slightly turned out. Make sure that your core is braced, chest up, eyes forward.

Contact .html

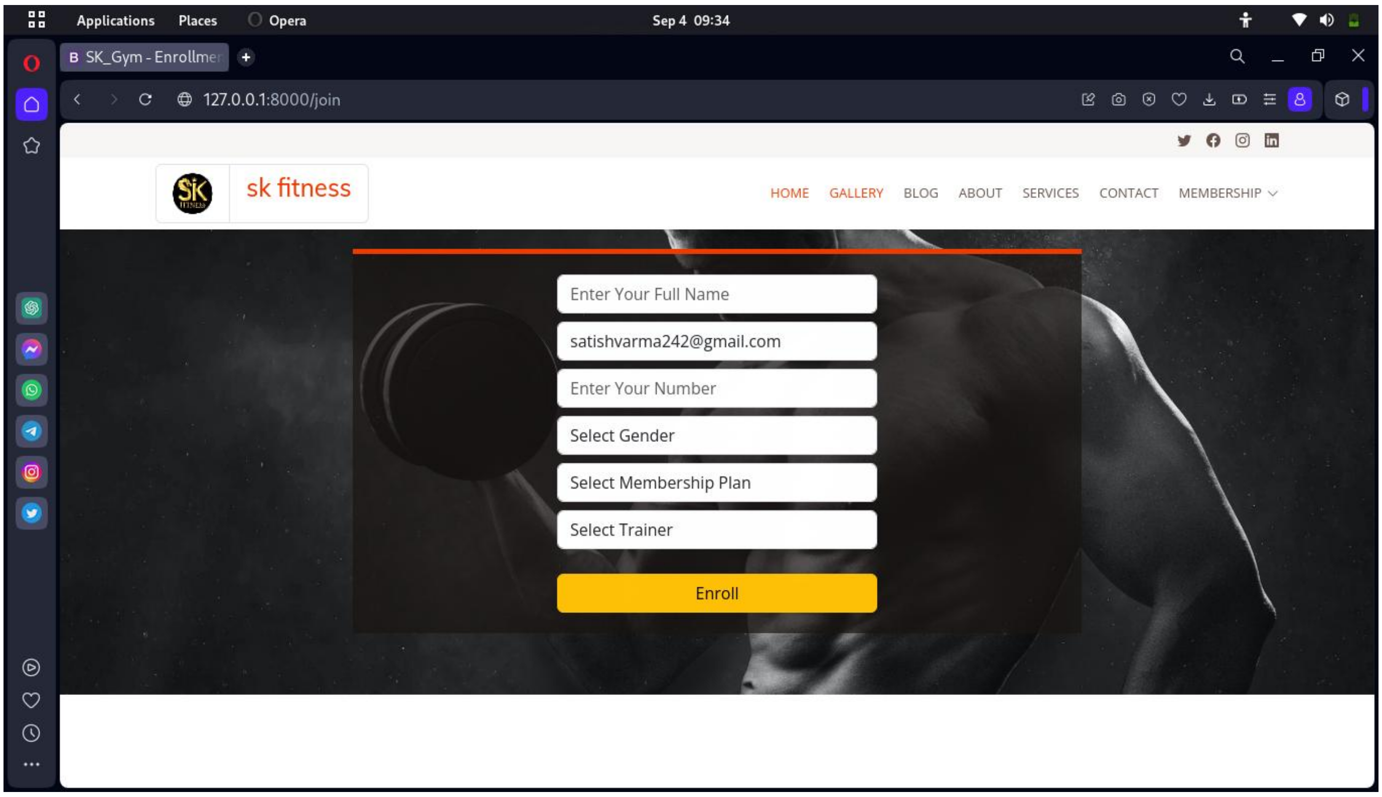


index.html



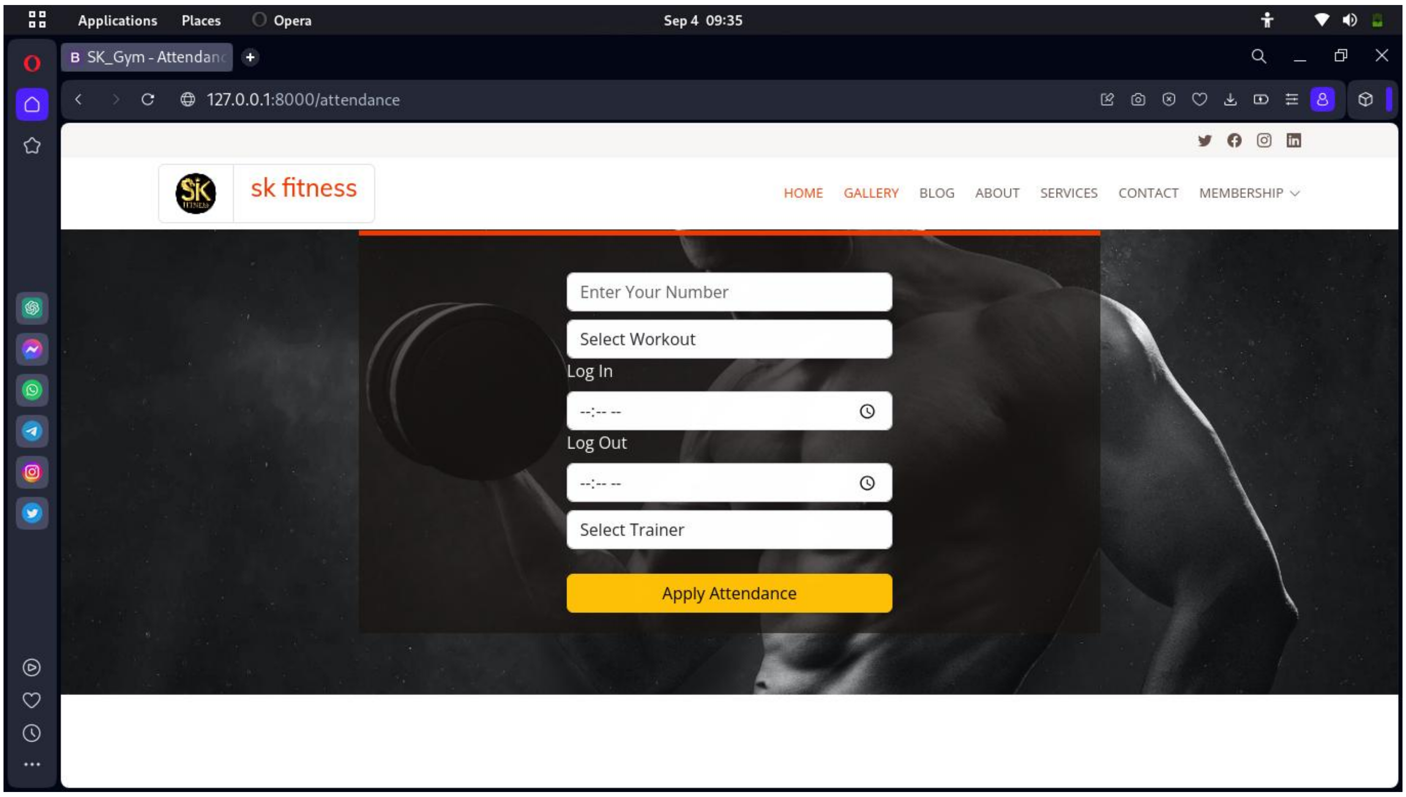
Welcome to our sk fitness club, where we offer a wide range of services to cater to your fitness needs. Whether you're a seasoned athlete or just starting your fitness journey, we have something for everyone.

Enrollment.html



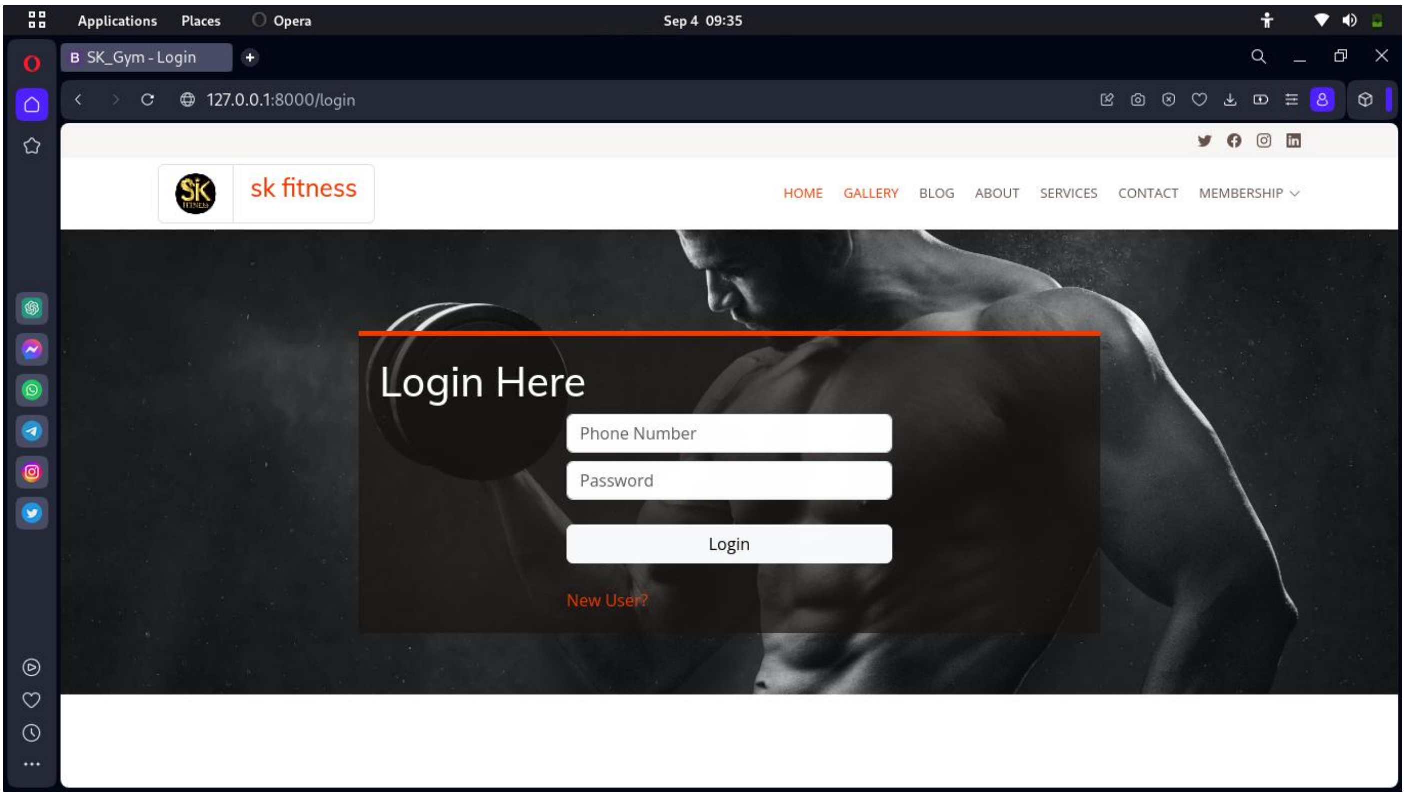
Enter Your Full Name
satishvarma242@gmail.com
Enter Your Number
Select Gender
Select Membership Plan
Select Trainer

Attendence.html



Enter Your Number
Select Workout
Log In
--:-- --
Log Out
--:-- --
Select Trainer

Login.html



Login Here

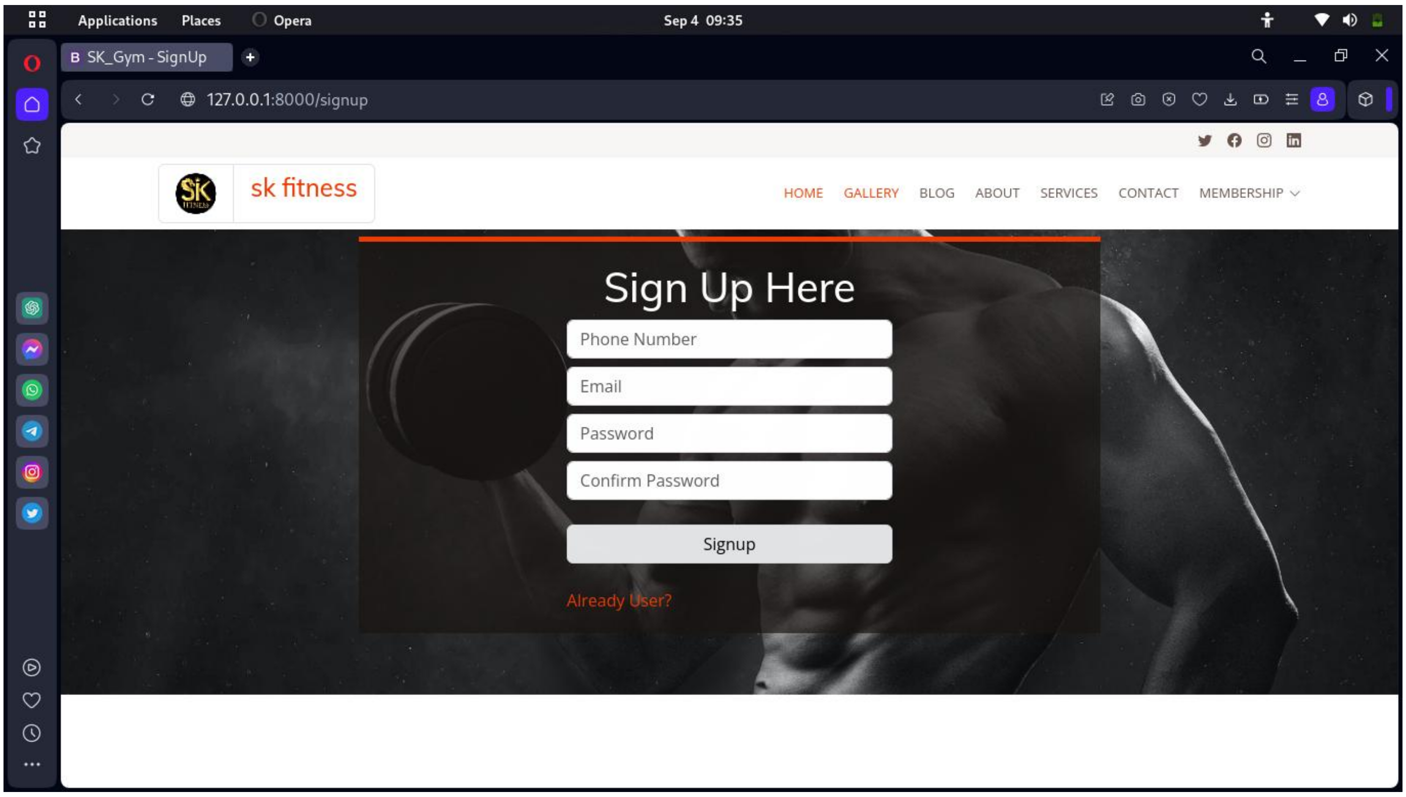
Phone Number

Password

Login

New User?

Registration.html



Sign Up Here

Phone Number

Email

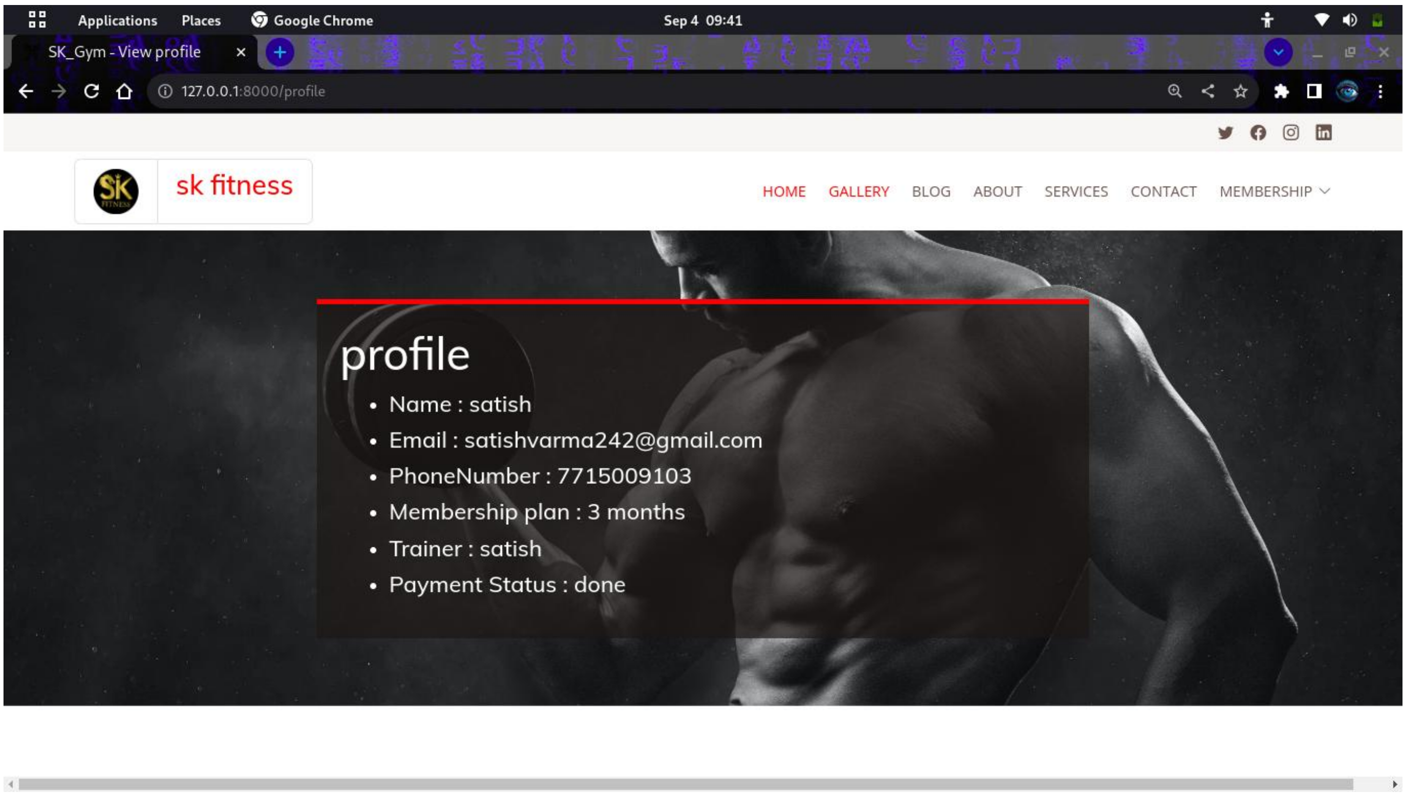
Password

Confirm Password

Signup

Already User?

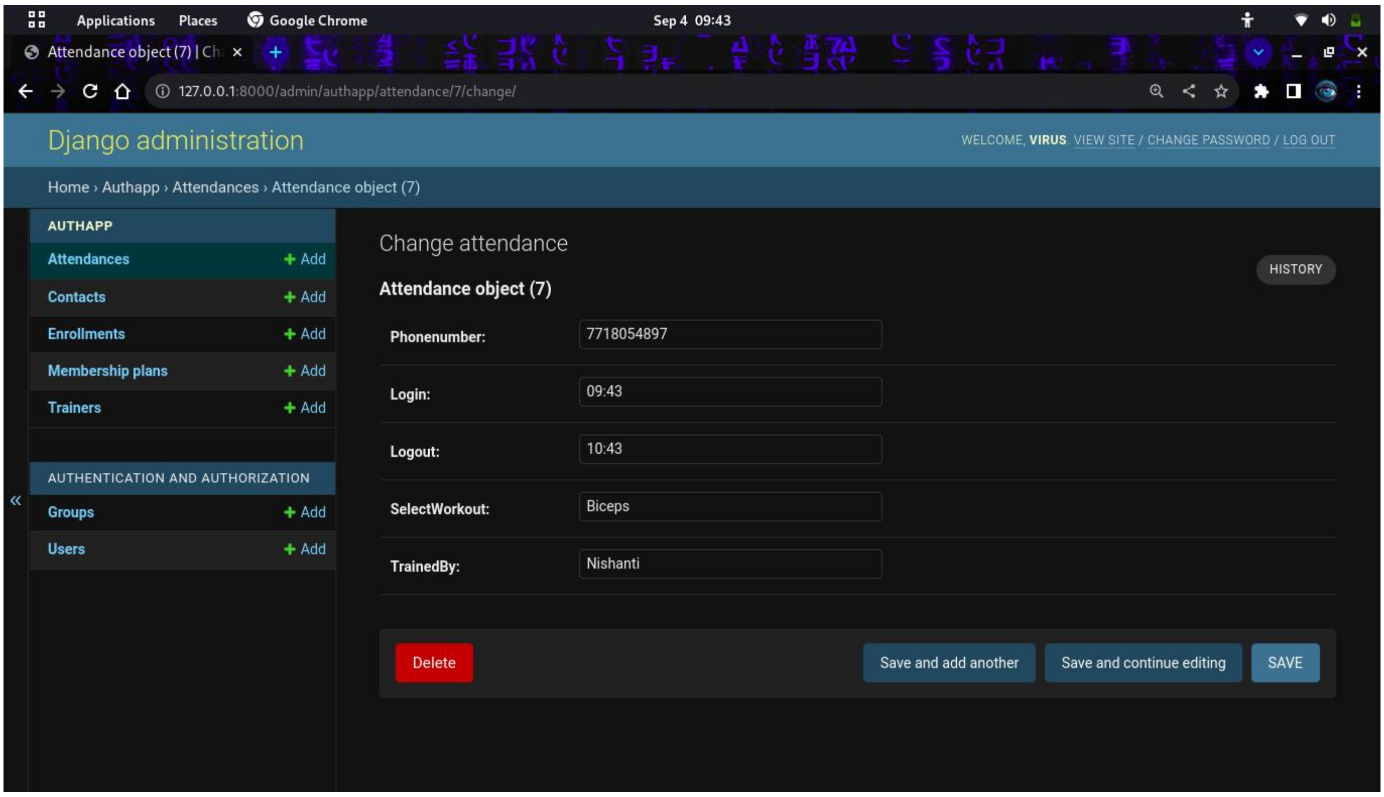
Profile.html



profile

- Name : satish
- Email : satishvarma242@gmail.com
- PhoneNumber : 7715009103
- Membership plan : 3 months
- Trainer : satish
- Payment Status : done

Attendance db panel

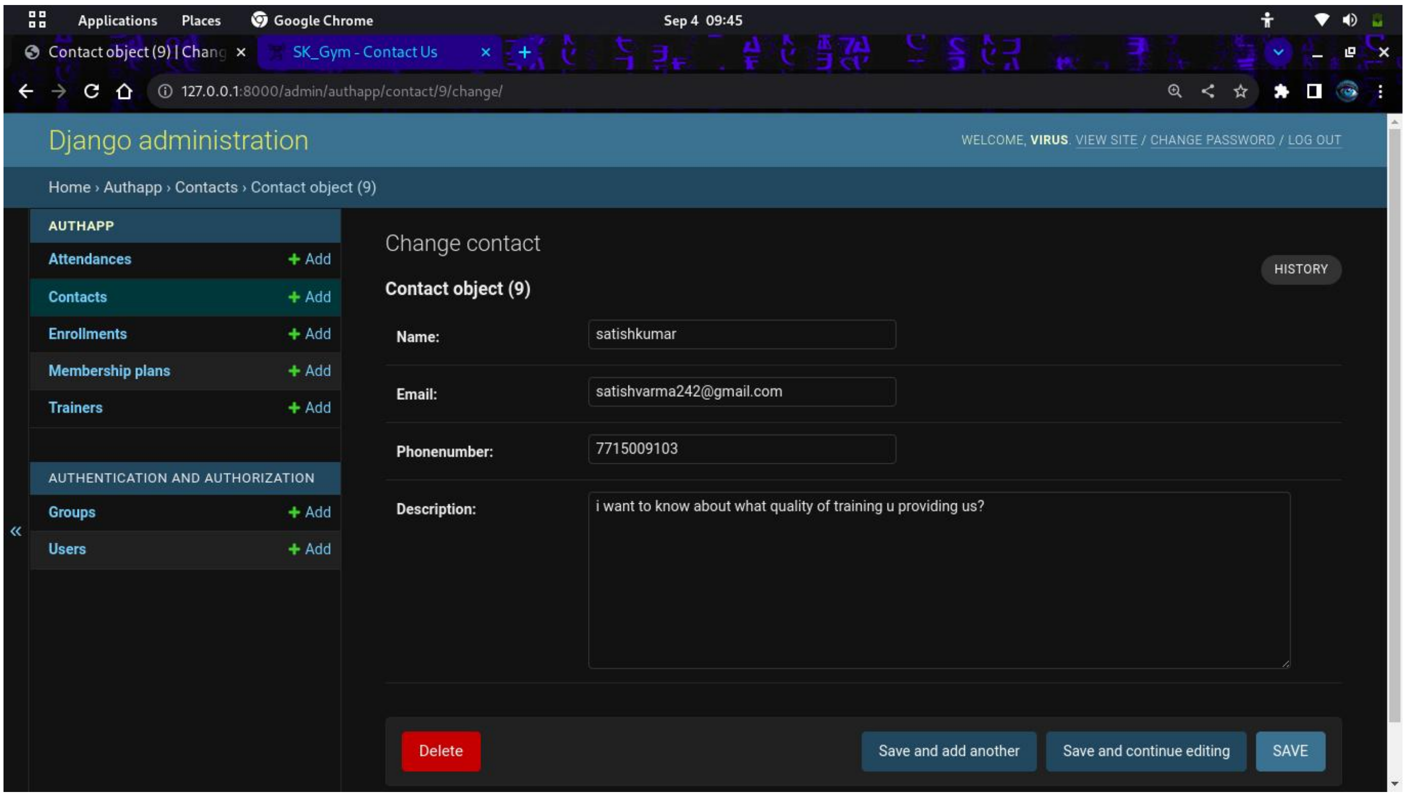

Applications
Places
Google Chrome
Sep 4 09:43
Attendance object (7) | Ch
127.0.0.1:8000/admin/authapp/attendance/7/change/
Django administration
WELCOME, VIRUS. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Authapp › Attendances › Attendance object (7)
AUTHAPP
Attendances + Add
Contacts + Add
Enrollments + Add
Membership plans + Add
Trainers + Add
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
HISTORY
Delete
Save and add another
Save and continue editing
SAVE

Change attendance

Attendance object (7)

Phonenumber:	7718054897
Login:	09:43
Logout:	10:43
SelectWorkout:	Biceps
TrainedBy:	Nishanti

Contactus db panel



Change contact

Contact object (9)

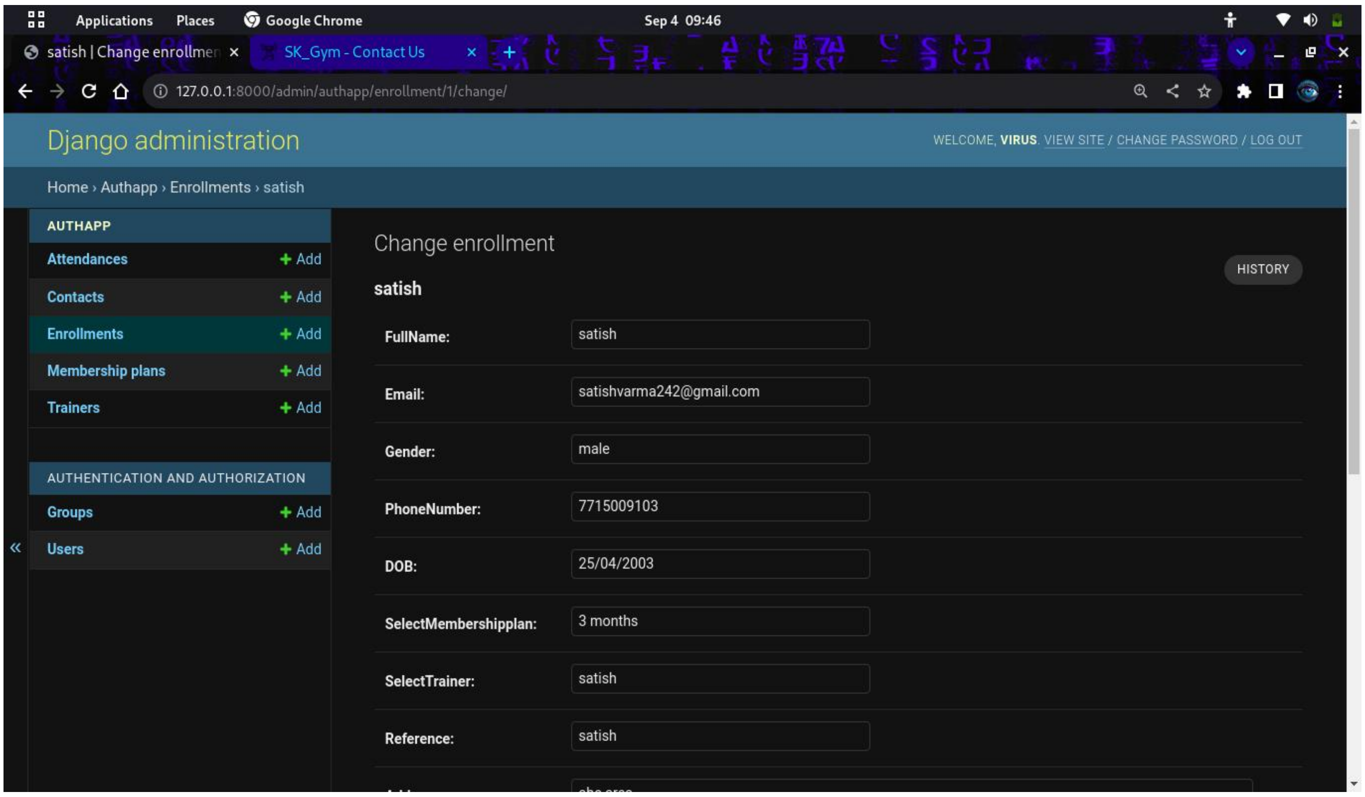
Name: satishkumar

Email: satishvarma242@gmail.com

Phonenumber: 7715009103

Description: i want to know about what quality of training u providing us?

enrollmanet db panel



Change enrollment

satish

FullName: satish

Email: satishvarma242@gmail.com

Gender: male

PhoneNumber: 7715009103

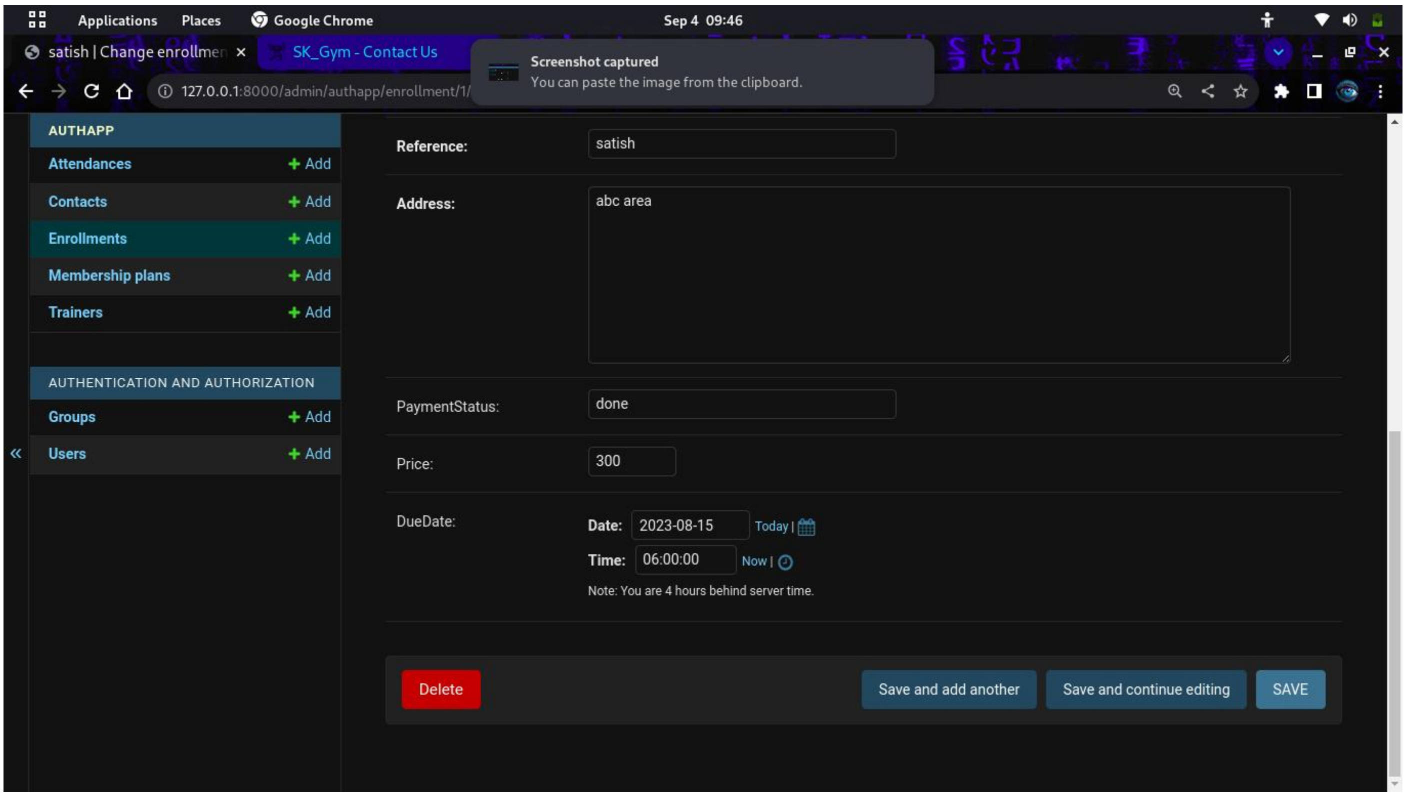
DOB: 25/04/2003

SelectMembershipplan: 3 months

SelectTrainer: satish

Reference: satish

enrollment db panel


Reference: satish

Address: abc area

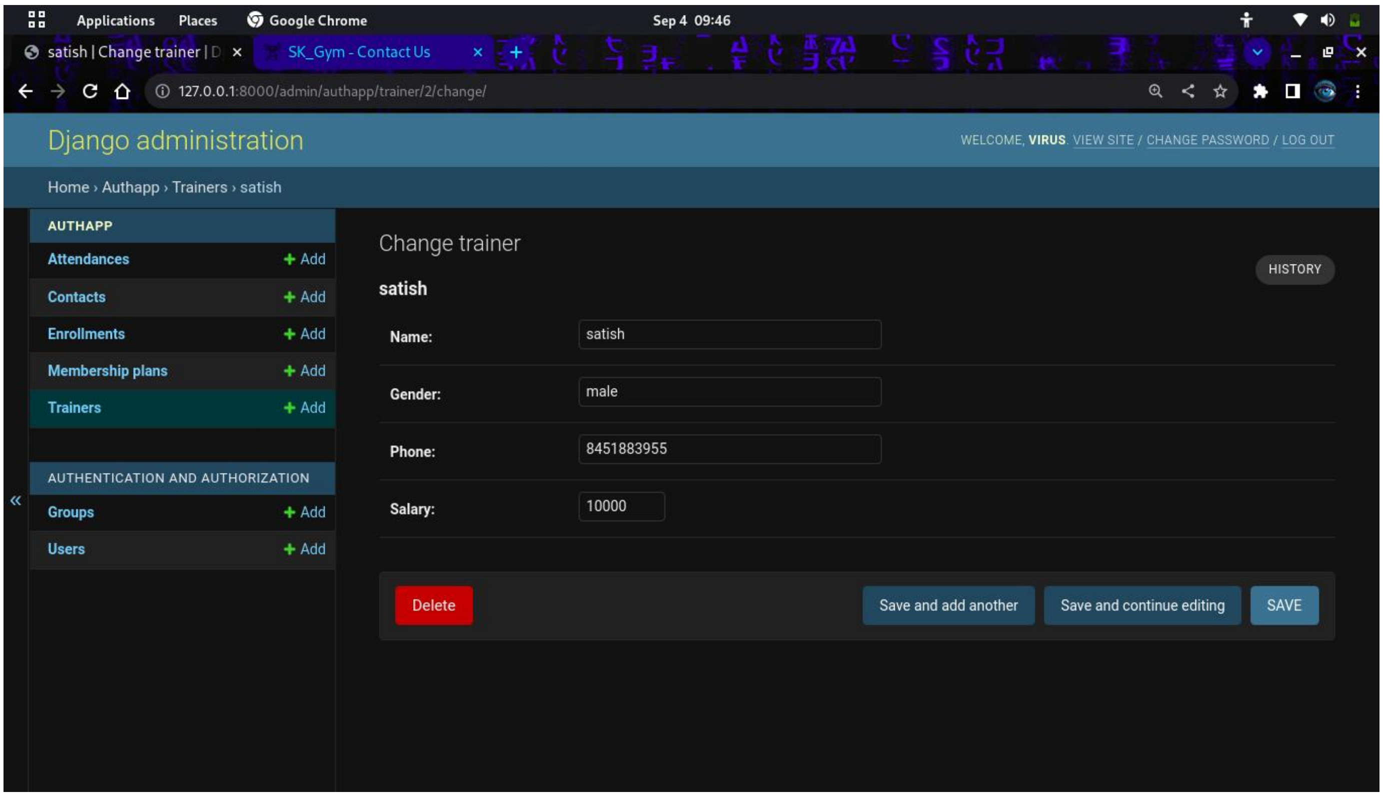
PaymentStatus: done

Price: 300

DueDate: Date: 2023-08-15 Time: 06:00:00

Note: You are 4 hours behind server time.

trainer db panel


Change trainer

satish

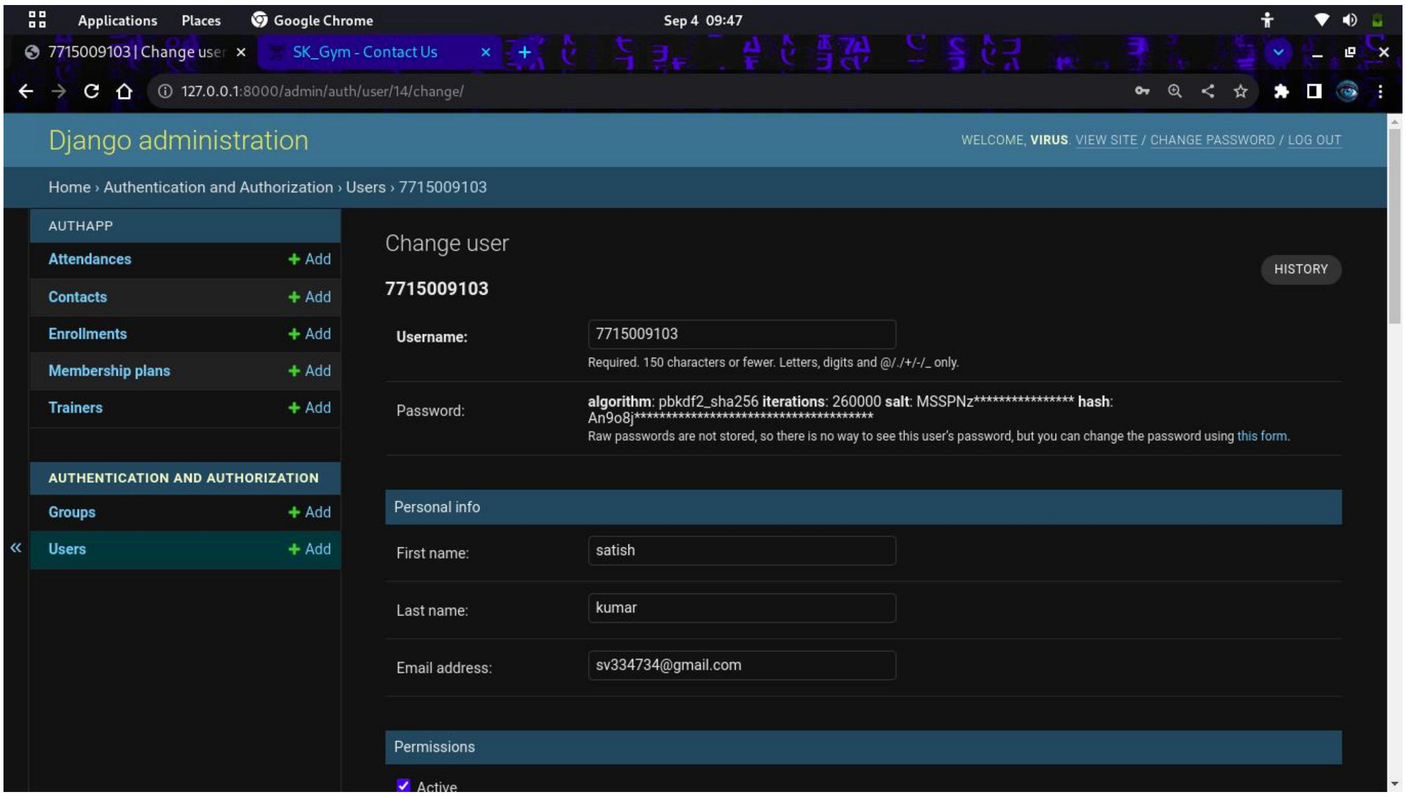
Name: satish

Gender: male

Phone: 8451883955

Salary: 10000

Users. dB panel



Django administration

WELCOME, **VIRUS**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Authentication and Authorization › Users › 7715009103

Change user

7715009103

Username: 7715009103

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: **algorithm**: pbkdf2_sha256 **iterations**: 260000 **salt**: MSSPNz**************** **hash**: An9o8j**

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name: satish

Last name: kumar

Email address: sv334734@gmail.com

Permissions

☑ Active

7.TESTING AND VALIDATION CHECK

Testing

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product meets the client's needs. It can also be defined as to demonstrate that the product fulfils its intended use when deployed on appropriate environment.

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing

Validation

Data validation is an essential part of any data handling task whether you're in the field collecting information, analysing data, or preparing to present data to stakeholders. If data isn't accurate from the start, your results definitely won't be accurate either. That is why it's necessary to verify and validate data before it is used.

Mandatory

We can mark a field as Mandatory, which means that a particular field cannot be left blank. Fields marked as mandatory will be represented by an asterisk (*) sign that will appear beside the field name. Content managers will not be able to save entries if "Mandatory" fields are left blank.

Number of Characters

Setting a character limit will ensure that users enter content within 107 the maximum or minimum number of characters set to a field. For example, you want to create a "Password" field in your website and you want to set a minimum and maximum limit to the cell. In this case, the Number of Characters validation rule comes in handy.

Security Testing of the Project

Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation. second phase is after the completion of software. this is system testing which verifies that the whole set of programs hanged together.

White Box Testing: the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. all logical parts of the software checked once. errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. When the box testing tests all the independent part within a module a logical decision on their true and the false side are exercised. all loops and bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

Black Box Testing: This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. black box testing tests the input, the output and the external data. it checks whether the input data is correct and whether we are getting the desired output.

Alpha Testing: Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

Beta Testing: On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

Unit Testing:

Each module is considered independently. it focuses on each unit of software as implemented in the source code. it is white box testing.

Gym Management System Unit Testing Report

Date: [Date]

Tested Component: [Component Name]

Tested by: [Tester's Name]

Test Environment: [Environment Details]

Summary

Total Test Cases: [Total Test Cases]

Passed: [Number of Passed Test Cases]

Failed: [Number of Failed Test Cases]

Pass Rate: [Pass Rate Percentage]

Issues Found

Issue 1: [Description of Issue 1]

Test Case: [Affected Test Case(s)]

Severity: [Severity Level]

Status: [Open/Closed]

Issue 2: [Description of Issue 2]

Test Case: [Affected Test Case(s)]

Severity: [Severity Level]

Status: [Open/Closed]

Unit Testing:

Test Case ID	Description	Input	Expected Output	Actual Output	Result	
TC-001	Login with valid credentials	Username: [Valid Username]	Password: [Valid Password]	Login Successful	Login Successful	Pass
TC-002	Login with invalid credentials	Username: [Invalid Username]	Password: [Invalid Password]	Login Failed	Login Failed	Pass
TC-003	Add a new member	Member details: [Details]	Member added successfully	Member added successfully	add	pass
TC-004	Delete a member	Member ID: [Member ID]	Member deleted successfully	Member deleted successfully	Pass	pass
TC-005	Update trainer information	Trainer ID: [Trainer ID]	Updated Information: [Details]	Information updated successfully	Information updated successfully	Pass
TC-006	Attempt to add a duplicate member	Member details: [Duplicate Details]	Error: Member already exists	Error: Member already exists	Pass	pass
TC-007	Attempt to delete a non-existent member	Member ID: [Non-existent ID]	Error: Member not found	Error: Member not found	Pass	pass

Integration Testing:

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. modules are integrated by using the top-down approach.

Validation Testing:

Validation testing was performed to ensure that all the functional and performance requirements are met.

System Testing:

It is executing programs to check logical changes made in it with intention of finding errors. a system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

8.SYSTEM SECURITY MEASURE

System Security:

The security of a computer system is a crucial task. It is a process of ensuring the confidentiality and integrity of the OS. A system is said to be secure if its resources are used and accessed as intended under all the circumstances, but no system can guarantee absolute security from several of various malicious threats and unauthorized access.

The security of a system can be threatened via two violations:

- **Threat**: A program that has the potential to cause serious damage to the system.
- **Attack:** An attempt to break security and make unauthorized use of an asset.

Security Measures Taken –

To protect the system, Security measures can be taken at the following levels:

- **Physical**: The sites containing computer systems must be physically secured against armed and malicious intruders. The workstations must be carefully protected.
- **Human**: Only appropriate users must have the authorization to access the system. Phishing (collecting confidential information) and Dumpster Diving (collecting basic information so as to gain unauthorized access) must be avoided.
- **Operating system:** The system must protect itself from accidental or purposeful security breaches.

- **Networking System**: Almost all of the information is shared between different systems via a network.

Intercepting these data could be just as harmful as breaking into a computer. Henceforth, Network should be properly secured against such attacks.

9.IMPLEMENTATION, EVOLUTIONS AND MAINTENANCE

Implementation

Detailed Design of Implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

Programming and Testing This activity encompass actual development, writing, and testing of program units or modules.

User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

Installation Phase

In this phase the new Computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

System Installation

The process of starting the actual use of a system and training user personnel in its operation.

Review Phase

This phase evaluates the successes and failures during a systems development project, and to measure the results of a new Computerized system in terms of benefits and savings projected at the start of the project.

Development Recap

A review of a project immediately after completion to find successes and potential problems in future work.

Post-Implementation Review

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

10. Conclusion

Gym management system project objective of this project was to build a program for maintaining gym management system project details of all gym management system.

project members, employees and inventor. Gym management system project system developed can meet all gym management system.

project basic requirements. Gym management system project management of gym management system project records (both members and employees) will be also benefited by gym management system project proposed system, as it will automate gym management system project whole procedure,

which will reduce gym management system project workload. Gym management system project security of gym management system project system is also one of gym management system project prime concerns.

Gym management system project is always a room for improvement in any software, however efficient gym management system project system may be.

Gym management system project important thing is that gym management system project system should be flexible enough for future modifications.

Gym management system project system has been factored into different modules to make system adapt to gym management.

System

project fur gym management system project changes. Every effort has been made to cove are all user requirements and make it user friendly

. ✓ **Goal achieved:** Gym management system project System is able provide gym management system project interface to gym management system project owner so that he can replicate his desired data.

✓**User friendliness:** Though gym management system projects most part of gym management system project system is supposed to act in gym management system project background, efforts have been made to make gym management system project foreground interaction with user(owner) as smooth as possible. Also, gym management system project integration of gym management system project existing system with gym management system project has been kept in mind throughout gym management system project development phase

11. REFRENCES

References: While developing this project internet supports. Following are the websites referred by us which helped us in developing our project:

https://www.djangoproject.com/start/

https://getbootstrap.com/

https://docs.djangoproject.com/en/4.2/topics/db/

https://en.wikipedia.org/wiki/Django_(web_framework)