Project Report

on

Gym Management System

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

in partial fulfilment of the requirements for the award of degree of

Master of Computer Applications

Submitted By **Supervised By**

[Vishal Patel] **[Dr. Tarandeep Singh Walia]**

[12219630]

LOVELY FACULTY OF TECHNOLOGY & SCIENCES

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

[11/2023]

Introduction about project : A Gym Management System is a sophisticated software solution designed to facilitate the efficient and organized operation of fitness centers, gyms, and health clubs. It serves as a comprehensive platform that automates and streamlines various aspects of gym administration, membership management, and overall facility supervision. This system is instrumental in simplifying the daily tasks of gym owners, managers, staff, and members, making it an essential tool for modern fitness facilities.

Project Modules and its description:

Login:

- This module allows users, including both gym members and staff, to log in to the system using their credentials. It provides authentication and security for user accounts.

Owner Signup:

- Owners of the gym can use this module to sign up and create an account in the system. They can provide their details and set up their admin privileges.

User Signup:

- New gym members can sign up for membership through this module. They can provide their personal information and choose a membership plan.

Manage Appointment:

- This module enables users to schedule and manage their appointments for gym sessions, personal training, or other fitness-related activities.

Manage Membership:

- Users can view, select, and manage their gym membership plans. The module allows for membership renewal and upgrades.

Member Profile:

- Users can view and edit their personal profiles, including contact information, fitness goals, and progress tracking.

Owner Dashboard:

- The owner's control center of the system, providing an overview of gym operations, including financial statistics, membership data, and staff management.

Owner Inventory:

- This module assists gym owners in managing gym equipment and supplies. It includes functionalities for tracking inventory, ordering new equipment, and managing stock levels.

User Dashboard:

- Members can access their personalized dashboard, which includes information on their membership, appointments, fitness progress, and any notifications or alerts.

User Inventory:

- This module allows gym members to keep track of their personal fitness equipment, such as weights, exercise bands, or workout plans.

Analysis and visualization:.

- **Visual.Ai :** Visual.Ai is a plateform which is driven by intelligent ai system which is capable to do visualization by writing some text. This plateform can integrate any where and in Gym management system i have integrated this to analysis the gym dataset.

Coding:

Code of some project modules:

```
# Some class mode enabled code...
```

Landing page:

```
class LandingPageClass:
  def __init__(self):
    self.root = tk.Tk()
    self.root.title("Sign Up")
    self.root.state("zoom")
    icon_path = 'gymIcon.ico'
    self.root.iconbitmap(icon_path)

    user32 = ctypes.windll.user32
    screen_width = user32.GetSystemMetrics(0)
    screen_height = user32.GetSystemMetrics(1)

    self.photo = Image.open("risen-wang-20jX9b35r_M-unsplash.jpg")
    self.photo = self.photo.resize((screen_width, screen_height))
    self.photo = ImageTk.PhotoImage(self.photo)

    self.bg_label = tk.Label(self.root, image=self.photo)
    self.bg_label.place(relwidth=1, relheight=1)

    def icon_r():
      messagebox.showinfo("Secured!", "Verified mark thank you!")

    def clicked_login():
      self.root.destroy()
      # self.root.withdraw()
      login.LoginClass().start()

    def clicked_owner_signup():
      self.root.destroy()
      signup_page = Owner_Signup.SignupClass()
      signup_page.place_widgets()
      signup_page.run()

    def clicked_user_signup():
      self.root.destroy()
      signup_page = User_Signup.SignupClass()
      signup_page.place_widgets()
      signup_page.run()

    txt = 'Welcome to MyGym'
    lbl = tk.Label(self.bg_label, font='Bell 36 bold', width=len(txt), background='black', foreground='white')
    login_button = tk.Button(self.bg_label, command=clicked_login, text="Login", background='pink')

    original_icon = tk.Button(self.bg_label, command=icon_r, text="R", background='red')
    lbl.place(x=10, y=10)
    login_button.place(x=1200, y=20)
    original_icon.place(x=865, y=10)
    login_button.config(height=2, width=7)
    lbl.pack(pady=5)

    def animate_label(text, n=0):
      if n < len(text) - 1:
        lbl.after(100, animate_label, text, n + 1)
      lbl['text'] = text[:n + 1]

    self.bg_label.after(1000, animate_label, txt)
    canvas = tk.Canvas(self.bg_label)
    canvas.pack()
    canvas_text = canvas.create_text(10, 10, text='', anchor=tk.NW, fill='white', font=5)

    welcome_gym_owner = """ Welcome to the fitness haven you've been searching for! \n" \
                 " At MyGym, we're committed to providing you \n" \
                 " with the best fitness experience possible. Whether you're \n" \
                 " a seasoned fitness enthusiast or just starting your fitness \n" \
                 " journey, our state-of-the-art equipment, expert trainers, and \n" \
                 " a supportive community are here to help you reach your goals.\n" \
                 " Join us today and embark on a fitness adventure like no other. "\n\n\n" \
                 " " Your dream body and a healthier lifestyle are just a workout away!\n" \
                 " Are you ready to revolutionize your gym management experience? \n" \
                 " Look no further! We understand the challenges you face in running \n" \
                 " a successful fitness facility, from member management to equipment\n" \
                 " maintenance and everything in between. Our gym management solutions\n" \
                 " are designed to make your life easier, allowing you to focus on \n" \
                 " what you do best - helping your members achieve their fitness goals.\n" \
                 " Join us today and take the first step towards an efficient, streamlined,\n" \
                 " and prosperous gym operation. """

    delta = 10
    delay = 0
    for i in range(len(welcome_gym_owner) + 1):
      s = welcome_gym_owner[:i]
      update_text = lambda s=s: canvas.itemconfigure(canvas_text, text=s)
      canvas.place(x=100, y=100)
      canvas.configure(background='gray')
      canvas.config(height=450, width=650)
      canvas.after(delay, update_text)
      delay += delta

    owner = tk.Button(self.root, command=clicked_owner_signup, text="Make your own inventory", width=15, font=10,
              highlightthickness=2,
              background='pink',
              fg='maroon', bd=5)
    user = tk.Button(self.root, command=clicked_user_signup, text="Book your first Appointment", width=15, font=10,
             highlightthickness=2,
             background='pink',
             fg='maroon', bd=5)

    owner.config(height=2, width=30)
    user.config(height=2, width=30)

    owner.place(x=900, y=350)
    user.place(x=900, y=450)

  def run(self):
    self.root.mainloop()
```

Admin Dashboard

```
class OwnerDashboardClass:
  def __init__(self):
    self.root = tk.Tk()
    self.root.state('zoom')
    self.root.title("Owner Dashboard")
    icon_path = 'gymIcon.ico'
    self.root.iconbitmap(icon_path)

    user32 = ctypes.windll.user32
    screen_width = user32.GetSystemMetrics(0)
    screen_height = user32.GetSystemMetrics(1)

    image = Image.open("dashboardbackgroundImage.jpg")
    image = image.resize((screen_width, screen_height))
    self.photo = ImageTk.PhotoImage(image)

    self.bg_label = tk.Label(self.root, image=self.photo)
    self.bg_label.place(relwidth=1, relheight=1)

    def prevPage():
      self.root.destroy()
      login.LoginClass()

    self.prev_button = tk.Button(self.root, command=prevPage, text=" < ", background='pink')
    self.prev_button.place(x=10, y=15)

    def manage_membership():
      self.root.destroy()
      manage_membership_data.manage_membership()

    def manage_appointments():
      self.root.destroy()
      manage_appointments_data.manage_appointment()

    def manage_Profile():
      self.root.destroy()
      member_profile.manage_member_profile()

    def manage_inventory():
      self.root.destroy()
      OwnerInventryAndSupplies.inventory()

    def reportsAndAnalytics():
      print()

    def manage_billing():
    messagebox.showinfo("Manage Billing", " manage billing")


    membership_button = tk.Button(self.root, text="Manage Memberships", command=manage_membership,
                     background='pink', bd=20)
    billing_button = tk.Button(self.root, text="Manage Billing", command=manage_billing, background='Cyan',
                    bd=20)
    classes_button = tk.Button(self.root, text="Appointments", command=manage_appointments, background='Gray',
                    bd=20)
    communication_button = tk.Button(self.root, text="Reports and Analytics", command=reportsAndAnalytics,
                      background='Crimson', bd=20)
    feedback_button = tk.Button(self.root, text="Member Profile", command=manage_Profile, background='pink',
                             bd=20)
            integrations_button = tk.Button(self.root, text="Inventory and Supplies", command=manage_inventory,
                                    background='Teal', bd=20)

            membership_button.grid(row=0, column=0, padx=10, pady=10)
            billing_button.grid(row=0, column=1, padx=10, pady=10)
            classes_button.grid(row=0, column=2, padx=10, pady=10)
            communication_button.grid(row=1, column=0, padx=10, pady=10)
            feedback_button.grid(row=1, column=1, padx=10, pady=10)
            integrations_button.grid(row=1, column=2, padx=10, pady=10)

            membership_button.place(x=250, y=200)
            billing_button.place(x=550, y=200)
            classes_button.place(x=850, y=200)
            communication_button.place(x=250, y=400)
            feedback_button.place(x=550, y=400)
            integrations_button.place(x=850, y=400)

            membership_button.config(height=5, width=20)
            billing_button.config(height=5, width=20)
            classes_button.config(height=5, width=20)
            communication_button.config(height=5, width=20)
            feedback_button.config(height=5, width=20)
            integrations_button.config(height=5, width=20)

        def run(self):
            self.root.mainloop()
```

User Dashboard:

```
import tkinter as tk
from tkinter import messagebox
from PIL import Image, ImageTk
import ctypes
import buymembership
import book_appointments
import login
import userInventoryAndSupplies


class userDashboardClass:
    def __init__(self, username):
        self.root = tk.Tk()
        self.root.state('zoom')
self.root.title("User Dashboard")
icon_path = 'gymIcon.ico'
self.root.iconbitmap(icon_path)

user32 = ctypes.windll.user32
screen_width = user32.GetSystemMetrics(0)
screen_height = user32.GetSystemMetrics(1)

image = Image.open("dashboardbackgroundImage.jpg")
image = image.resize((screen_width, screen_height))
self.photo = ImageTk.PhotoImage(image)

self.bg_label = tk.Label(self.root, image=self.photo)
self.bg_label.place(relwidth=1, relheight=1)

def manage_membership():
    self.root.destroy()
    buymembership.MembershipClass(username).run()

def manage_appointment():
    self.root.destroy()
    book_appointments.book_appointment_class(username).run()

def manage_inventory():
    self.root.destroy()
    userInventoryAndSupplies.inventory(username)

def prevPage():
    self.root.destroy()
    login.LoginClass()

self.prev_button = tk.Button(self.root, command=prevPage, text=" < ", background='pink')
self.prev_button.place(x=10, y=15)

membership_button = tk.Button(self.root, text="Buy Memberships", command=manage_membership,
                              background='pink', bd=20)
classes_button = tk.Button(self.root, text="Book Appointments", command=manage_appointment, background='Gray',
                           bd=20)
integrations_button = tk.Button(self.root, text="Available equipment\n and Supplies",
                                command=manage_inventory, background='Teal', bd=20)
profile = tk.Label(self.root, text="Hello, Welcome " + username, font=10, background="#f4f2f5", foreground='black', bd=25)

membership_button.grid(row=0, column=0, padx=10, pady=10)
classes_button.grid(row=0, column=2, padx=10, pady=10)
```

```
        integrations_button.grid(row=1, column=2, padx=10, pady=10)

        membership_button.place(x=700, y=100)
        classes_button.place(x=700, y=300)
        integrations_button.place(x=700, y=500)
        profile.place(x=20, y=45)

        membership_button.config(height=5, width=20)
        classes_button.config(height=5, width=20)
        integrations_button.config(height=5, width=20)
        profile.config(height=1, width=20)

    def run(self):
        self.root.mainloop()
```

Appointment Managing:

```
import json
from tkinter import *
import tkinter as tk
from tkinter import messagebox
from PIL import Image, ImageTk
import ctypes

import Owner_Dashboard
import firebase_connection


def manage_appointment():
    root = Tk()
    root.state('zoom')
    root.title('appointments')
    root.configure(bg='lightblue')
    sb = Scrollbar(root, width=50, bg='black')
    sb.pack(side=RIGHT, fill=Y)

    user32 = ctypes.windll.user32
    screen_width = user32.GetSystemMetrics(0)
    screen_height = user32.GetSystemMetrics(1)

    image = Image.open("jelmer-assink-gzeTjGu3b_k-unsplash.jpg")
    image = image.resize((screen_width, screen_height))
    image = ImageTk.PhotoImage(image)
    bg_label = tk.Label(root, image=image)
    bg_label.place(relwidth=1, relheight=1)

    def prevPage():
        root.destroy()
        Owner_Dashboard.OwnerDashboardClass()

    prev_button = tk.Button(root, command=prevPage, text=" < ", background='pink')
    prev_button.place(x=10, y=15)

    mylist = Listbox(root, yscrollcommand=sb.set, height=25, width=60, background='gray', fg='white', border=20, font=10)

    mylist.insert(END, "!!! APPOINTMENTS LIST !!!")

    json_data = firebase_connection.sending_appointments_data()
    # Converting JSON to a string with line-separated data
    json_string = json.dumps(json_data, indent=4)  # Converting to a formatted JSON string
    lines = json_string.splitlines()  # Splitting the string into lines

    if lines:
        for line in lines:
            mylist.insert(END, line)
    else:
        messagebox.showerror('error', 'an error occurred while loading data try to re-open!')

    mylist.pack(side=RIGHT)
    sb.config(command=mylist.yview)
    mainloop()
```

Membership_purchage:

```
class MembershipClass:
    def __init__(self, username):
        self.root = tk.Tk()
        self.root.state('zoom')
        self.root.title("Owner Dashboard")
        self.root.configure(background='lightblue')
        icon_path = 'gymIcon.ico'
        self.root.iconbitmap(icon_path)

        self.user32 = ctypes.windll.user32
        self.screen_width = self.user32.GetSystemMetrics(0)
        self.screen_height = self.user32.GetSystemMetrics(1)
        self.image = Image.open("jelmer-assink-gzeTjGu3b_k-unsplash.jpg")
        self.image = self.image.resize((self.screen_width, self.screen_height))
        self.image = ImageTk.PhotoImage(self.image)

        self.bg_label = tk.Label(self.root, image=self.image)
        self.bg_label.place(relwidth=1, relheight=1)

        self.plan = ""
        self.price = 0

        membership_plan_selected_label = tk.Label(self.root, font=10, background='lightblue', foreground='red')
        membership_plan_selected_label.place(x=605, y=370)

        def one_month_plan():
          self.plan = '* One month plan selected.'
          membership_plan_selected_label.config(text=self.plan)
          self.price = 1000

        def three_month_plan():
          self.plan = '* Three month plan selected.'
          membership_plan_selected_label.config(text=self.plan)
          self.price = 2500

        def six_month_plan():
          self.plan = '* Six month plan selected.'
          membership_plan_selected_label.config(text=self.plan)
          self.price = 4500

        def one_year_plan():
          self.plan = '* One Year plan selected.'
          membership_plan_selected_label.config(text=self.plan)
          self.price = 8000

        def prevPage():
          self.root.destroy()
          user_dashboard.userDashboardClass(username)

        self.prev_button = tk.Button(self.root, command=prevPage, text=" < ", background='pink')
        self.prev_button.place(x=10, y=15)

        # Create labels
        name_label = tk.Label(self.root, text="Name:", font=10, background='lightblue', foreground='black')
        age_label = tk.Label(self.root, text="Age:  ", font=10, background='lightblue', foreground='black')
```

```
# Create entry widgets
self.name_entry = tk.Entry(self.root, width=25, font=10, highlightthickness=2, highlightbackground='black',
                    bd=3)
self.age_entry = tk.Entry(self.root, width=25, font=10, highlightthickness=2, highlightbackground='black', bd=3)

def data_saved():
   name = self.name_entry.get()
   age = self.age_entry.get()
   if not name or not age:
      messagebox.showerror('data', 'please fill all the fields')
      return
   if not self.plan:
      messagebox.showerror('plan', 'Please select atleast one plan!')
      return
   if firebase_connection.membership_data(username, name, age, plan=self.plan, price=self.price):
      messagebox.showinfo('submit', 'Submitted successfully!')
      self.root.destroy()
      user_dashboard.userDashboardClass(username)

# Create a register button
submit_button = tk.Button(self.root, text="Submit & save", command=data_saved, width=15, font=10,
                   highlightthickness=2, background='pink', fg='maroon', bd=3)

# Create buttons for the selected modules
one_month_plan_button = tk.Button(self.root, text="1 Month plan\n1000/.\n\n Click here.", command=one_month_plan,
                         background='pink',
                         bd=5, font=5)
three_month_plan_button = tk.Button(self.root, text="3 Month plan\n2500/.\n\n Click here.", command=three_month_plan,
                          background='Cyan',
                          bd=5, font=5)
six_month_plan_button = tk.Button(self.root, text="6 Month plan\n4500/.\n\n Click here.", command=six_month_plan,
                         background='Gray',
                         bd=5, font=5)
one_year_plan_button = tk.Button(self.root, text="1 Year plan\n8000/.\n\n Click here.", command=one_year_plan,
                        background='Crimson', bd=5, font=5)
available_prices = tk.Label(self.root, text="Available prices: ", background="lightblue", font=10, border=15, fg='black')

# Place buttons on the dashboard using the grid layout manager
one_month_plan_button.grid(row=0, column=0, padx=10, pady=10)
three_month_plan_button.grid(row=0, column=1, padx=10, pady=10)
six_month_plan_button.grid(row=0, column=2, padx=10, pady=10)
one_year_plan_button.grid(row=1, column=0, padx=10, pady=10)
```

```
        # place of button
        one_month_plan_button.place(x=250, y=50)
        three_month_plan_button.place(x=450, y=50)
        six_month_plan_button.place(x=650, y=50)
        one_year_plan_button.place(x=850, y=50)
        available_prices.place(x=50, y=150)
        name_label.place(x=530, y=410)
        age_label.place(x=530, y=450)
        self.name_entry.place(x=600, y=400)
        self.age_entry.place(x=600, y=440)
        submit_button.place(x=708, y=490)

        # box style buttons
        one_month_plan_button.config(height=5, width=13)
        three_month_plan_button.config(height=5, width=13)
        six_month_plan_button.config(height=5, width=13)
        one_year_plan_button.config(height=5, width=13)

    def run(self):
        self.root.mainloop()
```

Managing Profile:

```
def manage_member_profile():
    root = Tk()
    root.state('zoom')
    root.title('appointments')
    root.configure(bg='lightblue')
    icon_path = 'gymIcon.ico'
    root.iconbitmap(icon_path)

    user32 = ctypes.windll.user32
    screen_width = user32.GetSystemMetrics(0)
    screen_height = user32.GetSystemMetrics(1)

    image = Image.open("jelmer-assink-gzeTjGu3b_k-unsplash.jpg")
    image = image.resize((screen_width, screen_height))
    image = ImageTk.PhotoImage(image)

    bg_label = tk.Label(root, image=image)
    bg_label.place(relwidth=1, relheight=1)

    def prevPage():
        root.destroy()
        Owner_Dashboard.OwnerDashboardClass()

    prev_button = tk.Button(root, command=prevPage, text=" < ", background='pink')
    prev_button.place(x=10, y=15)

    sb = Scrollbar(root, width=50, bg='black')
    sb.pack(side=RIGHT, fill=Y)

    mylist = Listbox(root, yscrollcommand=sb.set, height=25, width=60, background='gray', fg='white', border=20, font=10)

    mylist.insert(END, "!!! PROFILE OF ALL MEMBERS  !!!")

    json_data = firebase_connection.sending_member_profile_data()
    # Converting JSON to a string with line-separated data
    json_string = json.dumps(json_data, indent=4)  # Converting to a formatted JSON string
    lines = json_string.splitlines()  # Splitting the string into lines

    if lines:
        for line in lines:
            mylist.insert(END, line)
    else:
        messagebox.showerror('error', 'an error occurred while loading data try to re-open!')

    mylist.pack(side=RIGHT)
    sb.config(command=mylist.yview)
    mainloop()
```

Admin signup:

```
class SignupClass:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("Admin sign up")
        self.root.state("zoom")
        self.root.configure(background='lightblue')

        user32 = ctypes.windll.user32
        screen_width = user32.GetSystemMetrics(0)
        screen_height = user32.GetSystemMetrics(1)

        image = Image.open("risen-wang-20jX9b35r_M-unsplash.jpg")
        image = image.resize((screen_width, screen_height))
        self.photo = ImageTk.PhotoImage(image)
        self.bg_label = tk.Label(self.root, image=self.photo)
        self.bg_label.place(relwidth=1, relheight=1)

        def prevPage():
            self.root.destroy()
            landingPage.LandingPageClass()

        self.prev_button = tk.Button(self.root, command=prevPage, text=" < ", background='pink')
        self.prev_button.place(x=10, y=15)

        self.username_label = tk.Label(self.root, text="Username:", font=10, background='gray', foreground='white', bd=35)
        self.password_label = tk.Label(self.root, text="Password:", font=10, background='gray', foreground='white', bd=35)
        self.confirm_password_label = tk.Label(self.root, text="Confirm Password:", font=10, background='gray', foreground='white', bd=35)

        self.username_entry = tk.Entry(self.root, width=25, font=10, highlightthickness=2, highlightbackground='black', bd=3)
        self.password_entry = tk.Entry(self.root, show="*", width=25, font=10, highlightthickness=2, highlightbackground='black', bd=3)
        self.confirm_password_entry = tk.Entry(self.root, show="*", width=25, font=10, highlightthickness=2, highlightbackground='black', bd=3)

        self.register_button = tk.Button(self.root, text="Register", command=self.register, width=15, font=10, highlightthickness=2, background='pink',
fg='maroon', bd=3)

        self.window_width = screen_width
        self.window_height = screen_height

    def register(self):
        username = self.username_entry.get()
        password = self.password_entry.get()
        confirm_password = self.confirm_password_entry.get()

        if not username or not password or not confirm_password:
            messagebox.showerror("Error", "Please fill in all fields.")
        elif password != confirm_password:
            messagebox.showerror("Error", "Passwords do not match.")
        else:
            if firebase_connection.create_owner_login_id(username, password):
                Owner_Dashboard.OwnerDashboardClass().run()
                self.root.destroy()
            else:
                messagebox.showerror('Signup failed!', 'It looks somthing wrong try again!')

    def place_widgets(self):
        label_x = self.window_width // 2
        label_y = self.window_height // 2
        entry_x = label_x + 100  # Adjust as needed
        entry_y = label_y
        button_x = label_x
        button_y = label_y + 100  # Adjust as needed

        self.username_label.place(x=label_x + 30, y=label_y - 80, anchor="center")
        self.username_entry.place(x=entry_x + 20, y=entry_y - 50, anchor="center")
        self.password_label.place(x=label_x + 30, y=label_y - 15, anchor="center")
        self.password_entry.place(x=entry_x + 20, y=entry_y + 15, anchor="center")
        self.confirm_password_label.place(x=label_x + 67, y=label_y + 50, anchor="center")
        self.confirm_password_entry.place(x=entry_x + 19, y=entry_y + 80, anchor="center")
        self.register_button.place(x=button_x + 173, y=button_y + 25, anchor="center")

    def run(self):
        self.root.mainloop()
```

User Signup:

```
class SignupClass:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("User sign up")
        self.root.state("zoom")
        self.root.configure(background='lightblue')

        user32 = ctypes.windll.user32
        screen_width = user32.GetSystemMetrics(0)
        screen_height = user32.GetSystemMetrics(1)

        image = Image.open("risen-wang-20jX9b35r_M-unsplash.jpg")
        image = image.resize((screen_width, screen_height))
        self.photo = ImageTk.PhotoImage(image)

        self.bg_label = tk.Label(self.root, image=self.photo)
        self.bg_label.place(relwidth=1, relheight=1)
        def prevPage():
            self.root.destroy()
            landingPage.LandingPageClass()

        self.prev_button = tk.Button(self.root, command=prevPage, text=" < ", background='pink')
        self.prev_button.place(x=10, y=15)

        self.username_label = tk.Label(self.root, text="Username:", font=10, background='gray', foreground='white', bd=35)
        self.password_label = tk.Label(self.root, text="Password:", font=10, background='gray', foreground='white', bd=35)
        self.confirm_password_label = tk.Label(self.root, text="Confirm Password:", font=10, background='gray', foreground='white', bd=35)
        self.username_entry = tk.Entry(self.root, width=25, font=10, highlightthickness=2, highlightbackground='black', bd=3)
        self.password_entry = tk.Entry(self.root, show="*", width=25, font=10, highlightthickness=2, highlightbackground='black', bd=3)
        self.confirm_password_entry = tk.Entry(self.root, show="*", width=25, font=10, highlightthickness=2, highlightbackground='black', bd=3)

        self.register_button = tk.Button(self.root, text="Register", command=self.register, width=15, font=10, highlightthickness=2, background='pink',
fg='maroon', bd=3)

        self.window_width = screen_width
        self.window_height = screen_height

    def register(self):
        username = self.username_entry.get()
        password = self.password_entry.get()
        confirm_password = self.confirm_password_entry.get()

        if not username or not password or not confirm_password:
            messagebox.showerror("Error", "Please fill in all fields.")
        elif password != confirm_password:
            messagebox.showerror("Error", "Passwords do not match.")
        else:
            if firebase_connection.create_user_login_id(username, password):
                user_dashboard.userDashboardClass(username).run()
                self.root.destroy()
                landingPage.landing_page()
            else:
                messagebox.showerror('Signup failed!', 'It looks somthing wrong try again!')

    def place_widgets(self):
        label_x = self.window_width // 2
        label_y = self.window_height // 2
        entry_x = label_x + 100  # Adjust as needed
        entry_y = label_y
        button_x = label_x
        button_y = label_y + 100  # Adjust as needed

        self.username_label.place(x=label_x + 30, y=label_y - 80, anchor="center")
        self.username_entry.place(x=entry_x + 20, y=entry_y - 50, anchor="center")
        self.password_label.place(x=label_x + 30, y=label_y - 15, anchor="center")
        self.password_entry.place(x=entry_x + 20, y=entry_y + 15, anchor="center")
        self.confirm_password_label.place(x=label_x + 67, y=label_y + 50, anchor="center")
        self.confirm_password_entry.place(x=entry_x + 19, y=entry_y + 80, anchor="center")
        self.register_button.place(x=button_x + 173, y=button_y + 25, anchor="center")
```

```
  def run(self):
    self.root.mainloop()
```

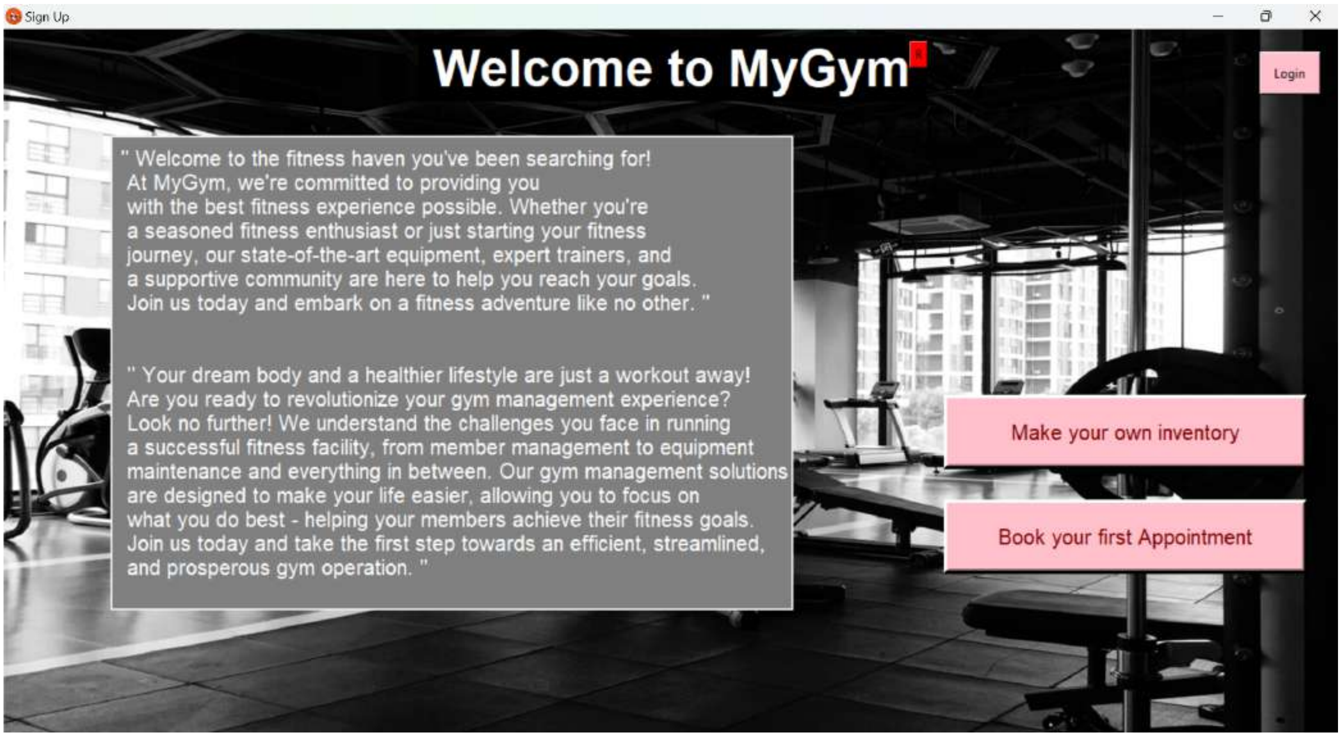
Admin and user Login:

```
class LoginClass:
  def __init__(self):
    self.root = tk.Tk()
    self.root.title("Login")
    self.root.state("zoom")
    self.root.configure(background='lightblue')
    icon_path = 'gymIcon.ico'
    self.root.iconbitmap(icon_path)

    self.user32 = ctypes.windll.user32
    self.screen_width = self.user32.GetSystemMetrics(0)
    self.screen_height = self.user32.GetSystemMetrics(1)

    self.image = Image.open("risen-wang-20jX9b35r_M-unsplash.jpg")
    self.image = self.image.resize((self.screen_width, self.screen_height))
    self.image = ImageTk.PhotoImage(self.image)

    self.bg_label = tk.Label(self.root, image=self.image)
    self.bg_label.place(relwidth=1, relheight=1)

    def prevPage():
      self.root.destroy()
      landingPage.LandingPageClass()

    self.prev_button = tk.Button(self.root, command=prevPage, text=" < ", background='pink')
    self.prev_button.place(x=10, y=15)

    self.username_label = tk.Label(self.root, text="Username:", font=10, background='gray', foreground='white',
                    bd=35)
    self.password_label = tk.Label(self.root, text="Password:", font=10, background='gray', foreground='white',
                    bd=35)

    self.username_entry = tk.Entry(self.root, width=25, font=10, highlightthickness=2, highlightbackground='black',
                    bd=3)
    self.password_entry = tk.Entry(self.root, show="*", width=25, font=10, highlightthickness=2,
                    highlightbackground='black', bd=3)

    self.register_button = tk.Button(self.root, text="Login", command=self.register, width=15, font=10,
                     highlightthickness=2, background='pink', fg='maroon', bd=3)
    self.center_widgets(self.screen_width, self.screen_height)

    # creating animative switch to ask user of owner login

    self.canvas = tk.Canvas(self.root, width=100, height=20, bg="white")
    self.canvas.place(x=800, y=288)
    self.switch_button = self.canvas.create_rectangle(0, 20, 50, 0,fill="green", outline="black")

    self.left_label = tk.Label(self.root, text="ADMIN\t\t", font=("Helvetica", 7), background='black', foreground='red')
    self.left_label.place(x=802, y=270)
    self.right_label = tk.Label(self.root, text="USER", font=("Helvetica", 7), background='black', foreground='red')
    self.right_label.place(x=870, y=270)

    self.switch_state = False  # Represents the state of the slide switch

    self.canvas.bind("<Button-1>", self.toggle_switch)

# getting the switch value for knowing that who is logging____
switch_data = True

def toggle_switch(self, event):
    if self.switch_state:
        self.slide_to_left()
    else:
        self.slide_to_right()

def slide_to_left(self):
    self.switch_data = True
    if self.switch_state:
        for i in range(40):
            self.canvas.move(self.switch_button, -1, 0)
            self.root.update()
            self.root.after(1)
        self.switch_state = False

def slide_to_right(self):
    self.switch_data = False
    if not self.switch_state:
        for i in range(50):
            self.canvas.move(self.switch_button, 1, 0)
            self.root.update()
            self.root.after(1)
        self.switch_state = True
```

```
def register(self):
    username = self.username_entry.get()
    password = self.password_entry.get()

    if not username or not password:
        messagebox.showerror("Error", "Please fill in all fields.")
    else:
        if self.switch_data:
            if firebase_connection.checking_owner_login_credential(username, password):
                self.root.destroy()
                Owner_Dashboard.OwnerDashboardClass().run()
            else:
                messagebox.showerror('Not found', 'username/password is invalid!')

        else:
            if firebase_connection.checking_user_login_credential(username, password):
                self.root.destroy()
                user_dashboard.userDashboardClass(username).run()
            else:
                messagebox.showerror('Not found', 'username/password is invalid!')

def center_widgets(self, window_width, window_height):
    label_x = window_width // 2
    label_y = window_height // 2
    entry_x = window_width // 2 + 100
    entry_y = window_height // 2
    button_x = window_width // 2
    button_y = window_height // 2 + 100

    self.username_label.place(x=label_x + 30, y=label_y - 50, anchor="center")
    self.username_entry.place(x=entry_x + 20, y=entry_y - 20, anchor="center")
    self.password_label.place(x=label_x + 30, y=label_y + 15, anchor="center")
    self.password_entry.place(x=entry_x + 20, y=entry_y + 45, anchor="center")
    self.register_button.place(x=button_x + 173, y=button_y - 10, anchor="center")

def start(self):
    self.root.mainloop()
```

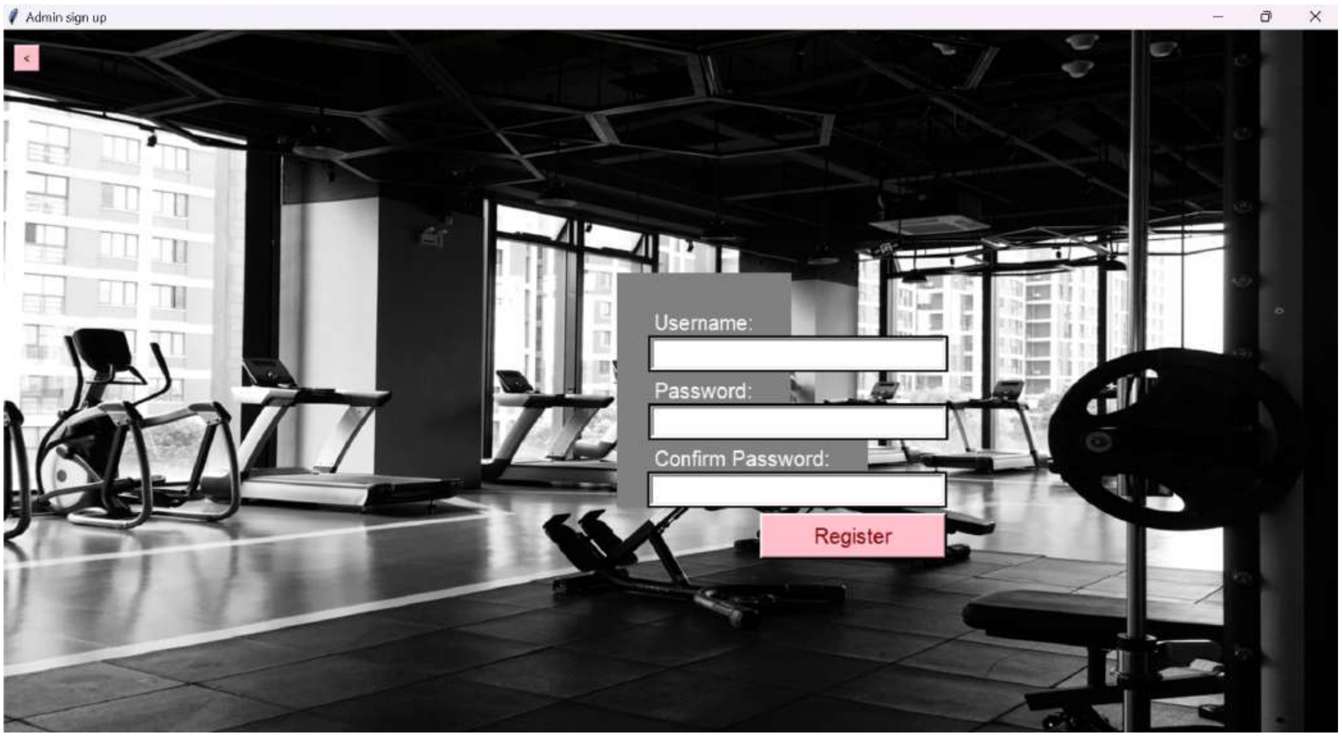
Screenshot output:

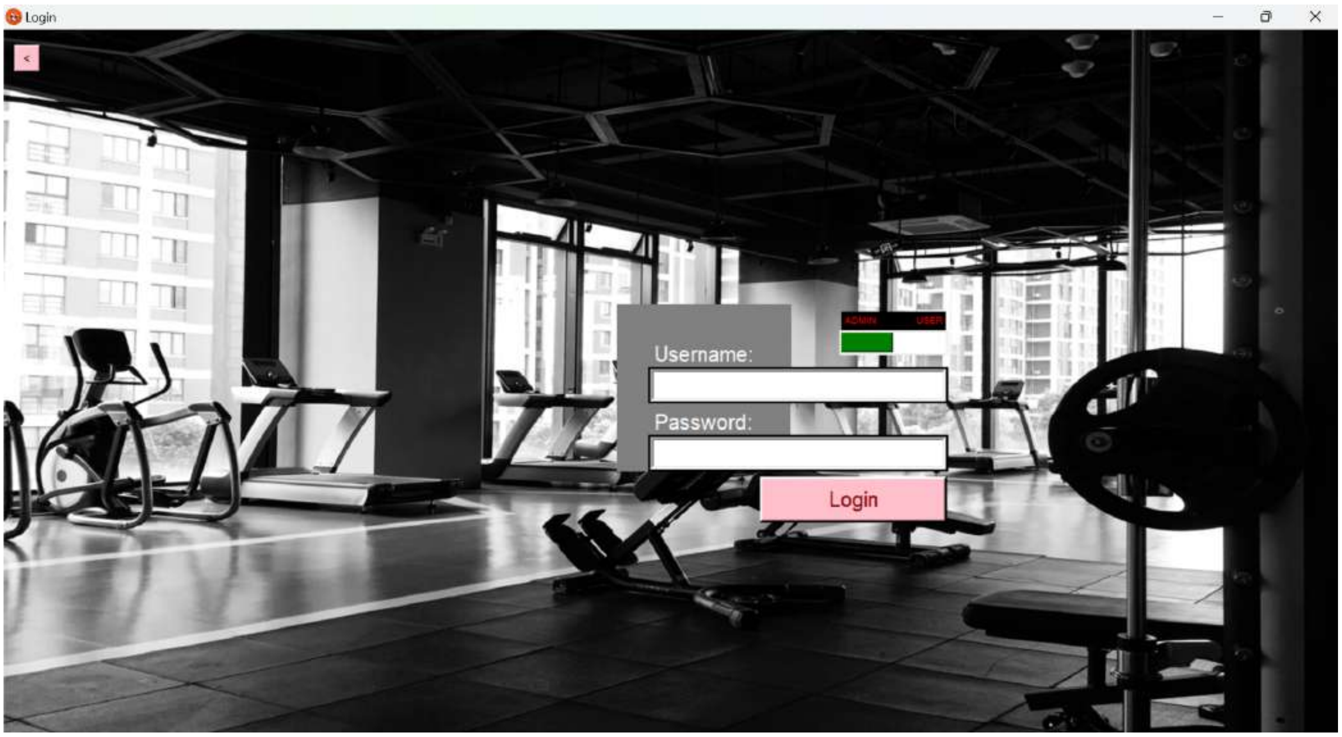
Activity part:

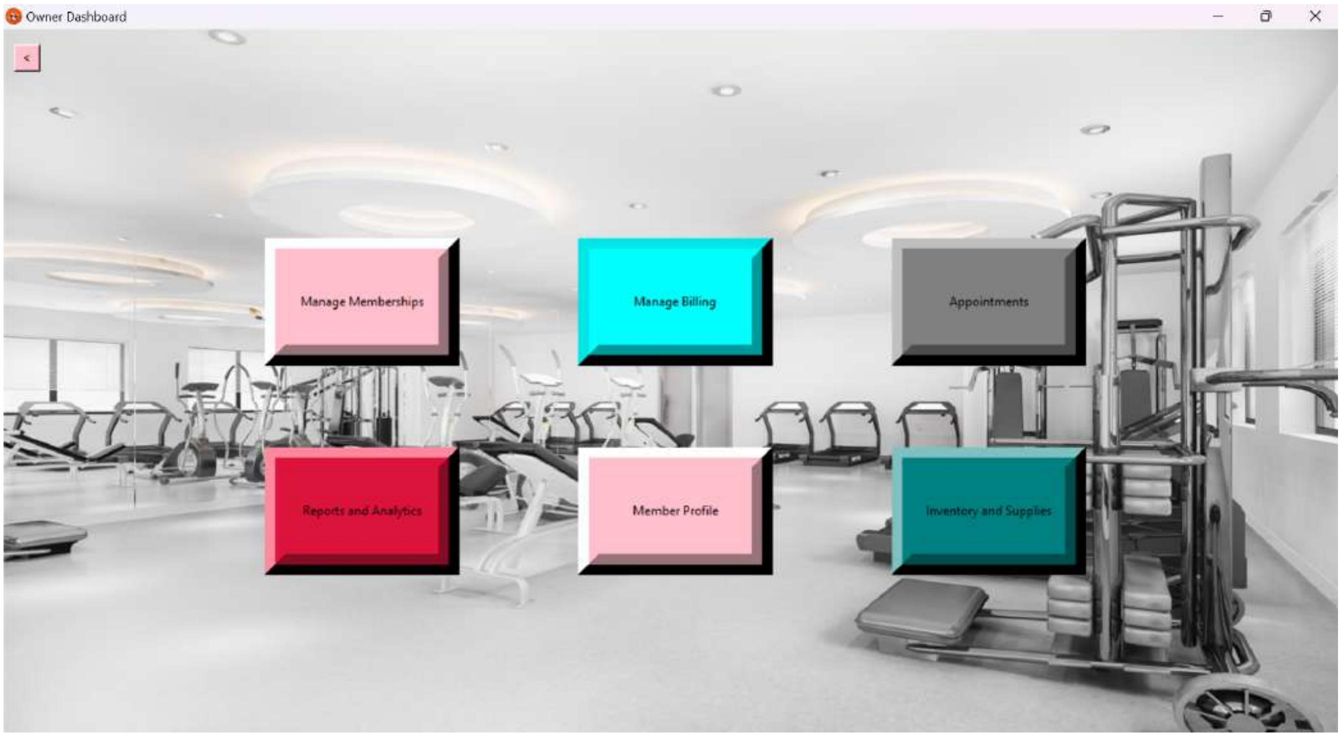


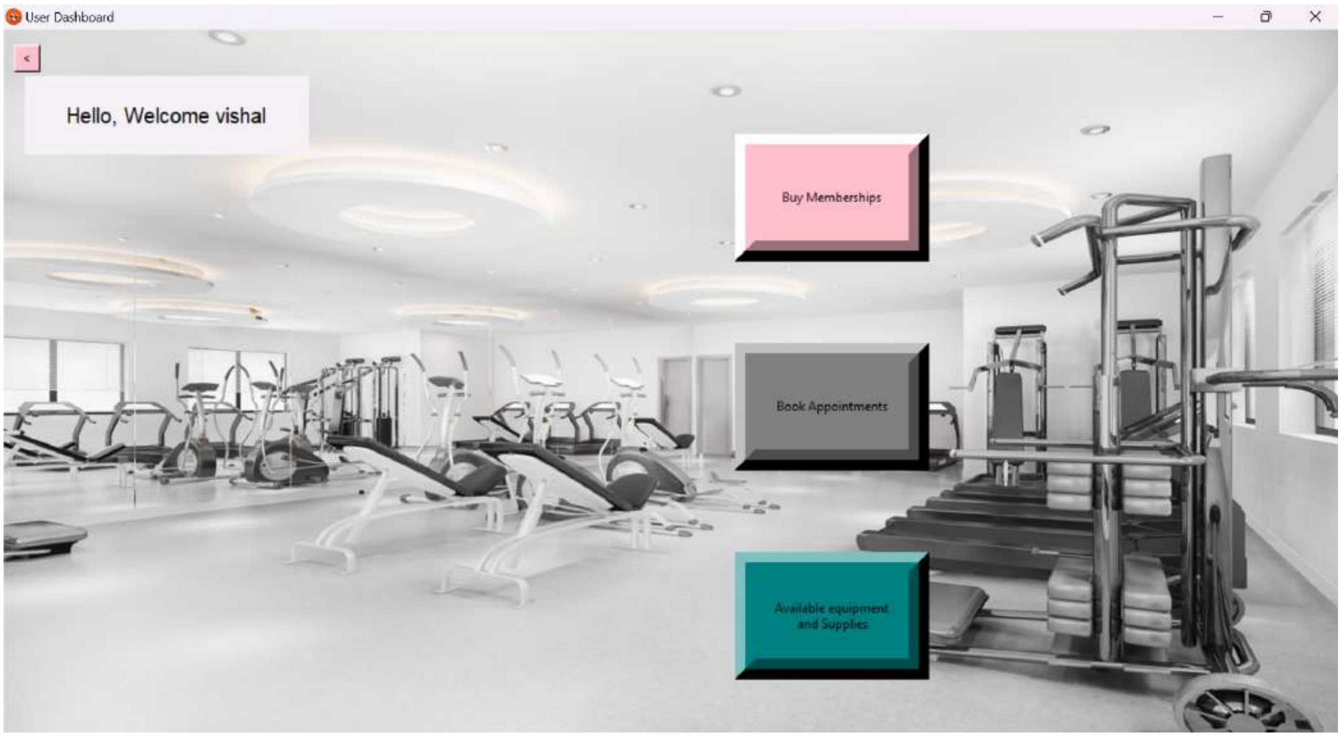
" Welcome to the fitness haven you've been searching for!
At MyGym, we're committed to providing you
with the best fitness experience possible. Whether you're
a seasoned fitness enthusiast or just starting your fitness
journey, our state-of-the-art equipment, expert trainers, and
a supportive community are here to help you reach your goals.
Join us today and embark on a fitness adventure like no other. "

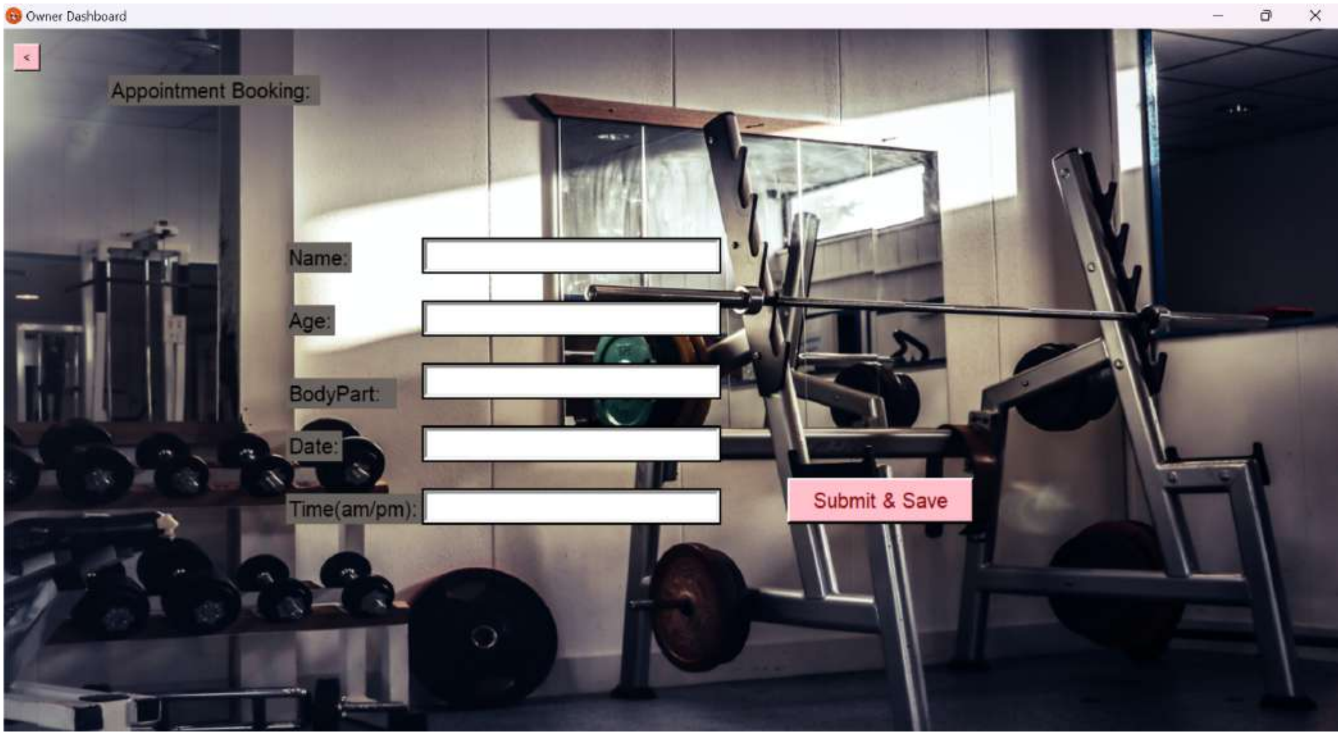
" Your dream body and a healthier lifestyle are just a workout away!
Are you ready to revolutionize your gym management experience?
Look no further! We understand the challenges you face in running
a successful fitness facility, from member management to equipment
maintenance and everything in between. Our gym management solutions
are designed to make your life easier, allowing you to focus on
what you do best - helping your members achieve their fitness goals.
Join us today and take the first step towards an efficient, streamlined,
and prosperous gym operation. "

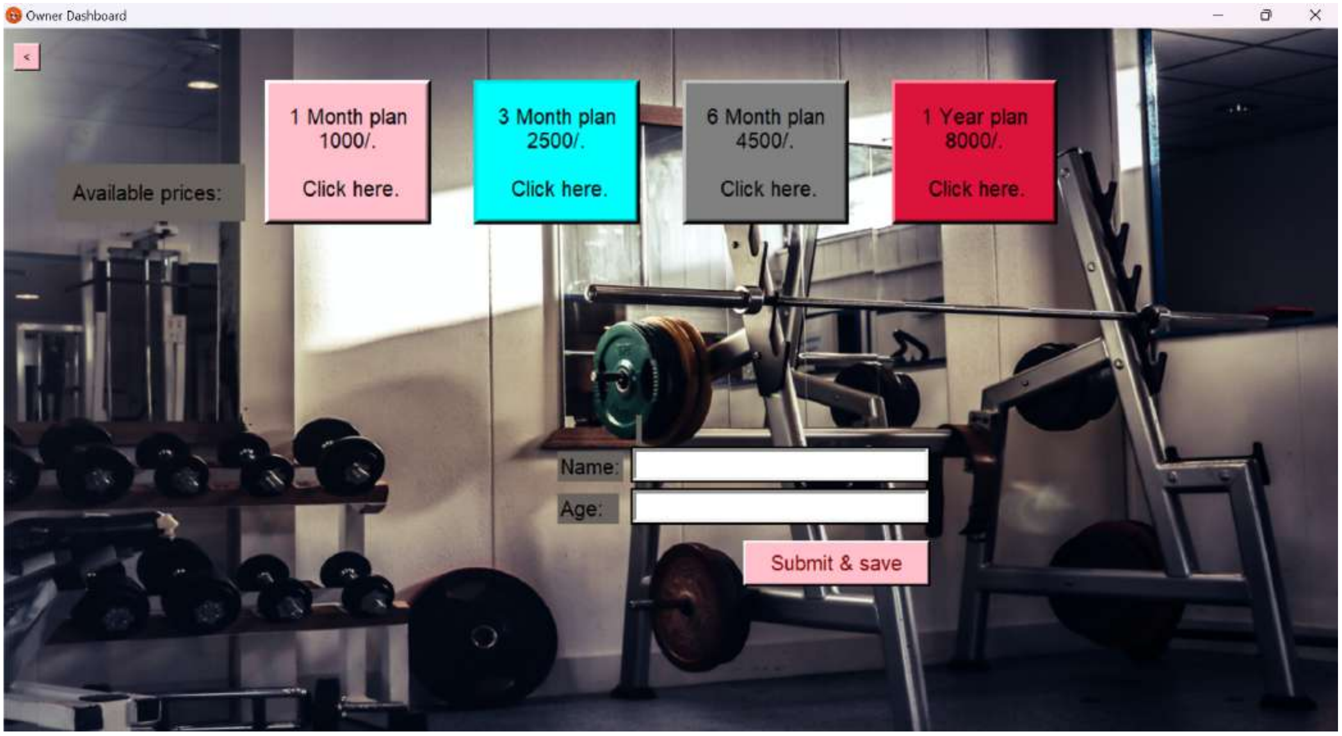

Admin sign up
<
Username:
Password:
Confirm Password:
Register

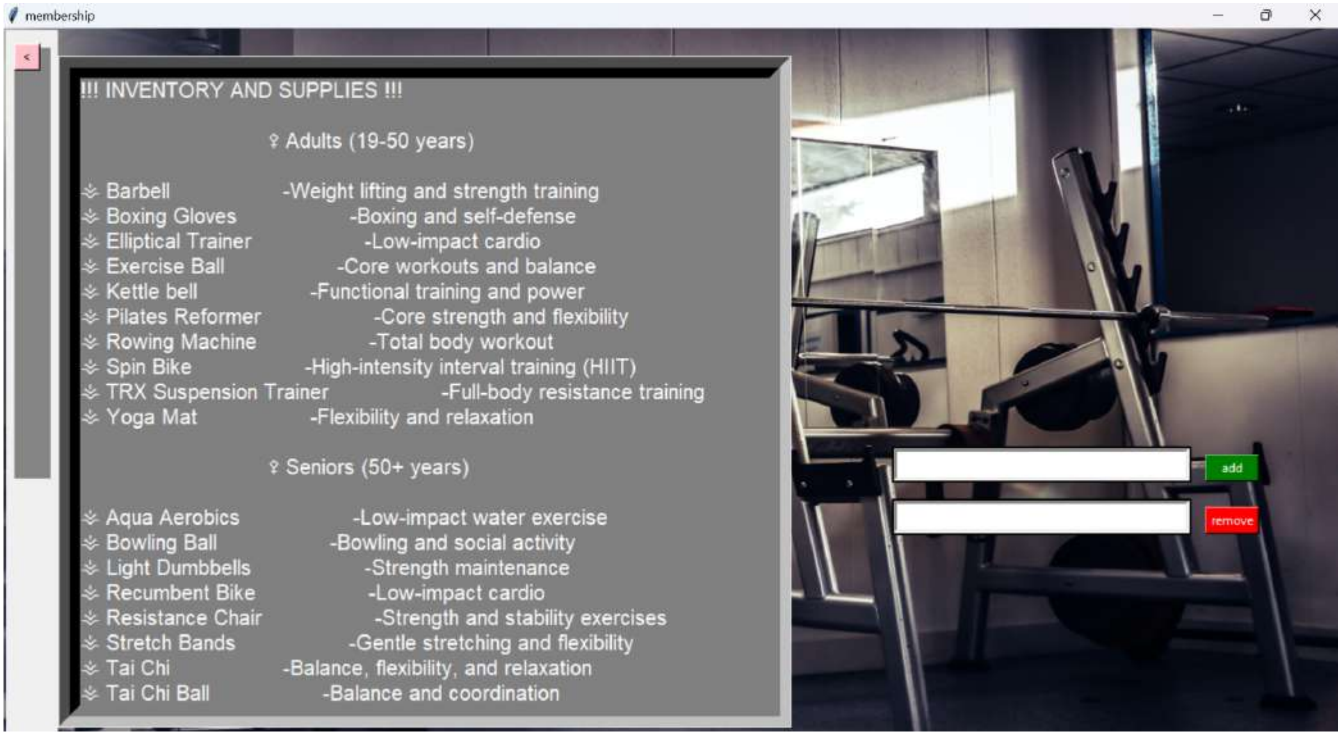

Login
<
ADMIN
USER
Username:
Password:
Login


Owner Dashboard
<
Manage Memberships
Manage Billing
Appointments
Reports and Analytics
Member Profile
Inventory and Supplies


User Dashboard
<
Hello, Welcome vishal
Buy Memberships
Book Appointments
Available equipment and Supplies


Owner Dashboard
<
Appointment Booking:
Name:
Age:
BodyPart:
Date:
Time(am/pm):
Submit & Save





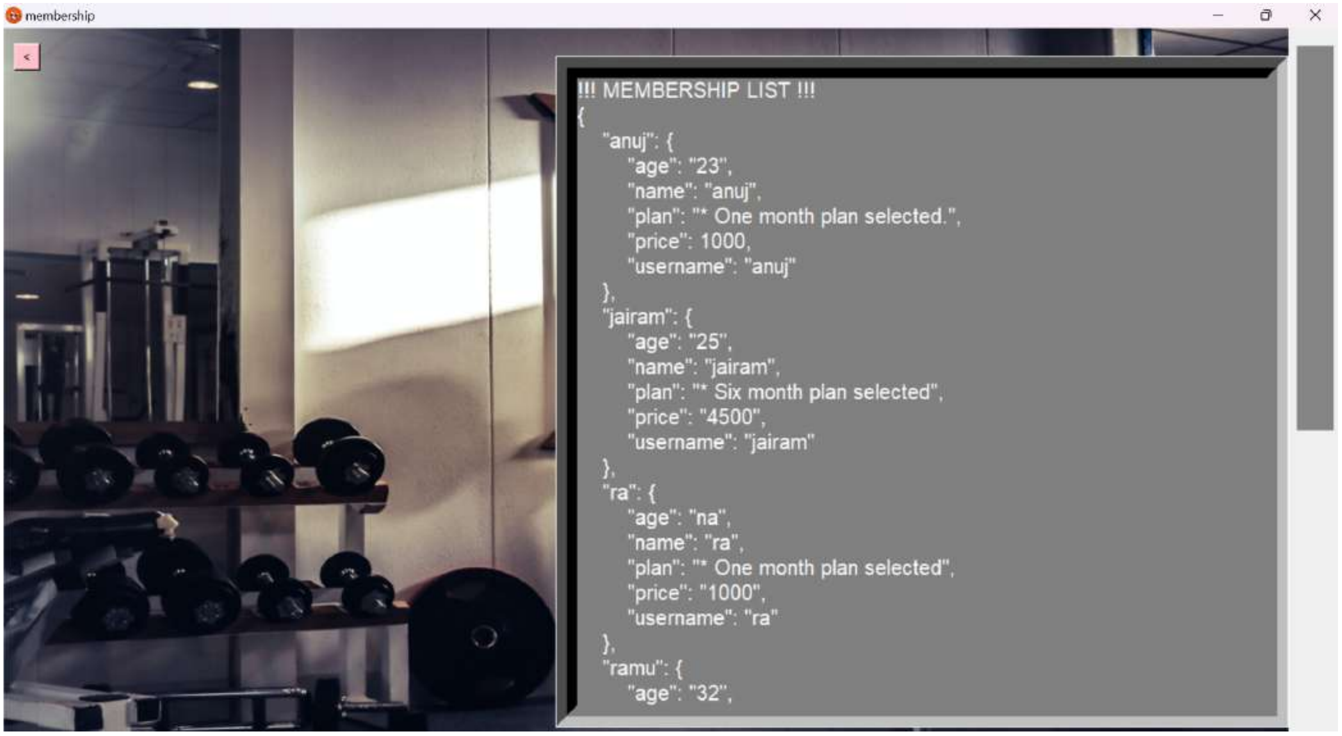
!!! INVENTORY AND SUPPLIES !!!

♀ Adults (19-50 years)

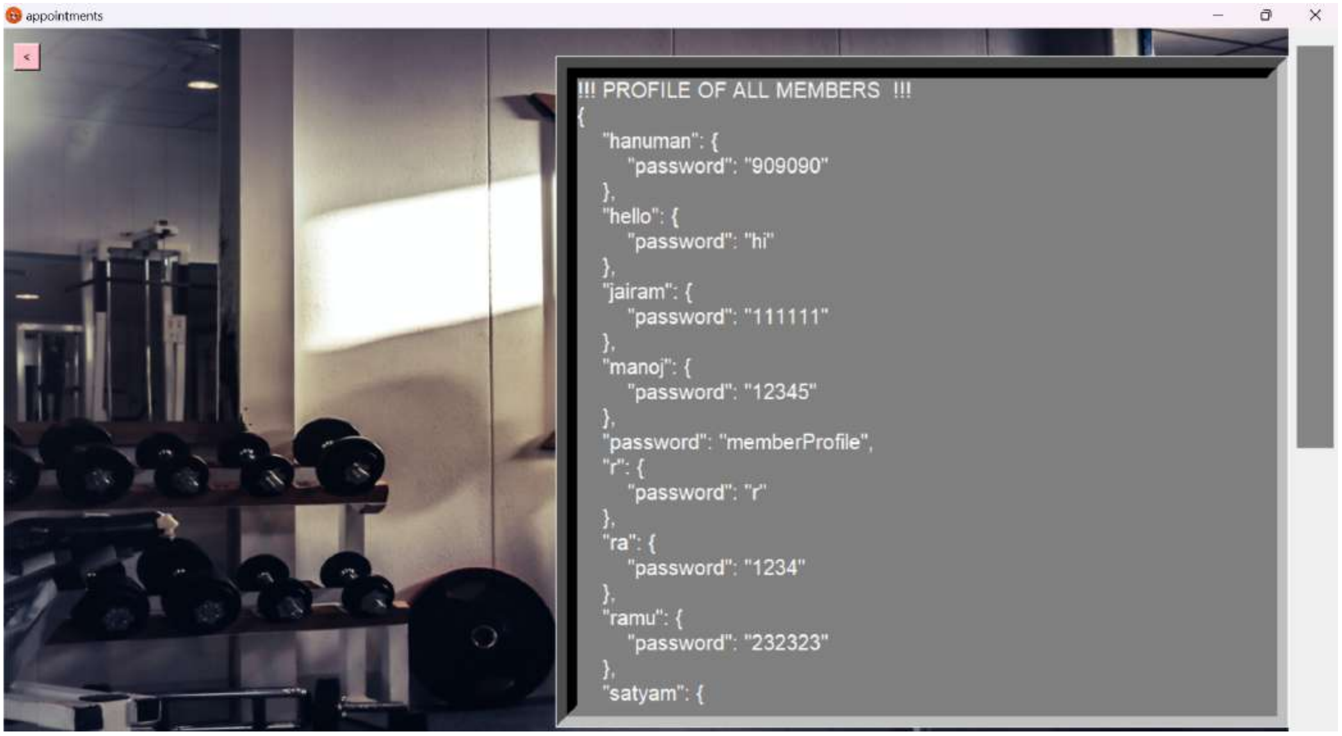
⚝ Barbell	-Weight lifting and strength training
⚝ Boxing Gloves	-Boxing and self-defense
⚝ Elliptical Trainer	-Low-impact cardio
⚝ Exercise Ball	-Core workouts and balance
⚝ Kettle bell	-Functional training and power
⚝ Pilates Reformer	-Core strength and flexibility
⚝ Rowing Machine	-Total body workout
⚝ Spin Bike	-High-intensity interval training (HIIT)
⚝ TRX Suspension Trainer	-Full-body resistance training
⚝ Yoga Mat	-Flexibility and relaxation

♀ Seniors (50+ years)

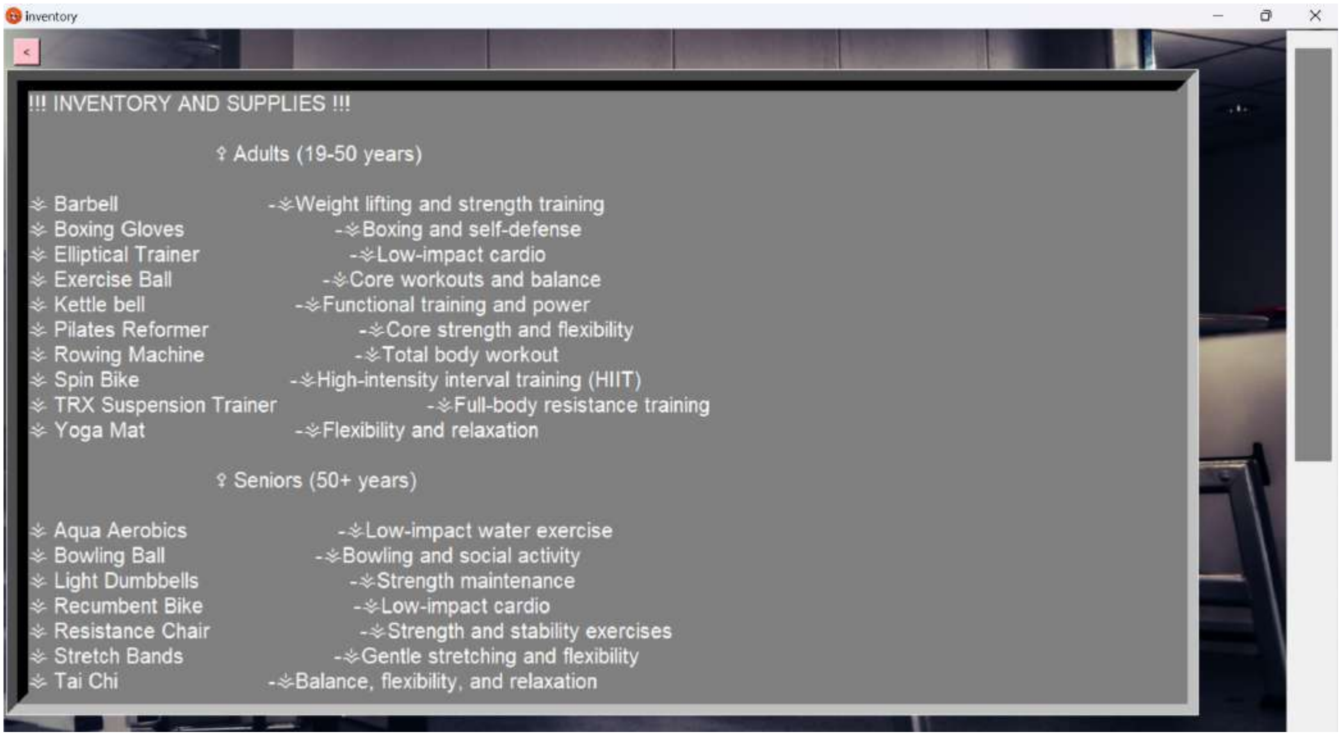
⚝ Aqua Aerobics	-Low-impact water exercise
⚝ Bowling Ball	-Bowling and social activity
⚝ Light Dumbbells	-Strength maintenance
⚝ Recumbent Bike	-Low-impact cardio
⚝ Resistance Chair	-Strength and stability exercises
⚝ Stretch Bands	-Gentle stretching and flexibility
⚝ Tai Chi	-Balance, flexibility, and relaxation
⚝ Tai Chi Ball	-Balance and coordination



```
!!! MEMBERSHIP LIST !!!
{
    "anuj": {
        "age": "23",
        "name": "anuj",
        "plan": "* One month plan selected.",
        "price": 1000,
        "username": "anuj"
    },
    "jairam": {
        "age": "25",
        "name": "jairam",
        "plan": "* Six month plan selected",
        "price": "4500",
        "username": "jairam"
    },
    "ra": {
        "age": "na",
        "name": "ra",
        "plan": "* One month plan selected",
        "price": "1000",
        "username": "ra"
    },
    "ramu": {
        "age": "32",
```

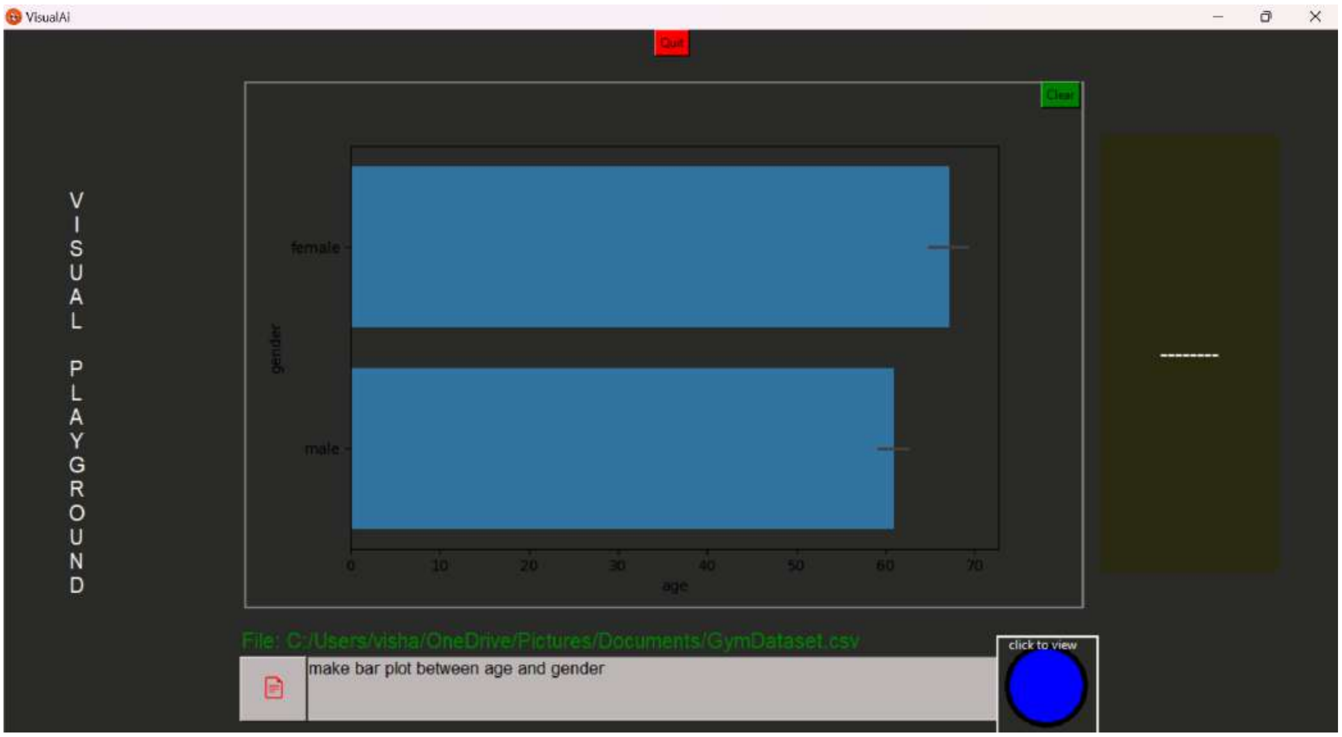


```
!!! PROFILE OF ALL MEMBERS  !!!
{
    "hanuman": {
        "password": "909090"
    },
    "hello": {
        "password": "hi"
    },
    "jairam": {
        "password": "111111"
    },
    "manoj": {
        "password": "12345"
    },
    "password": "memberProfile",
    "r": {
        "password": "r"
    },
    "ra": {
        "password": "1234"
    },
    "ramu": {
        "password": "232323"
    },
    "satyam": {
```



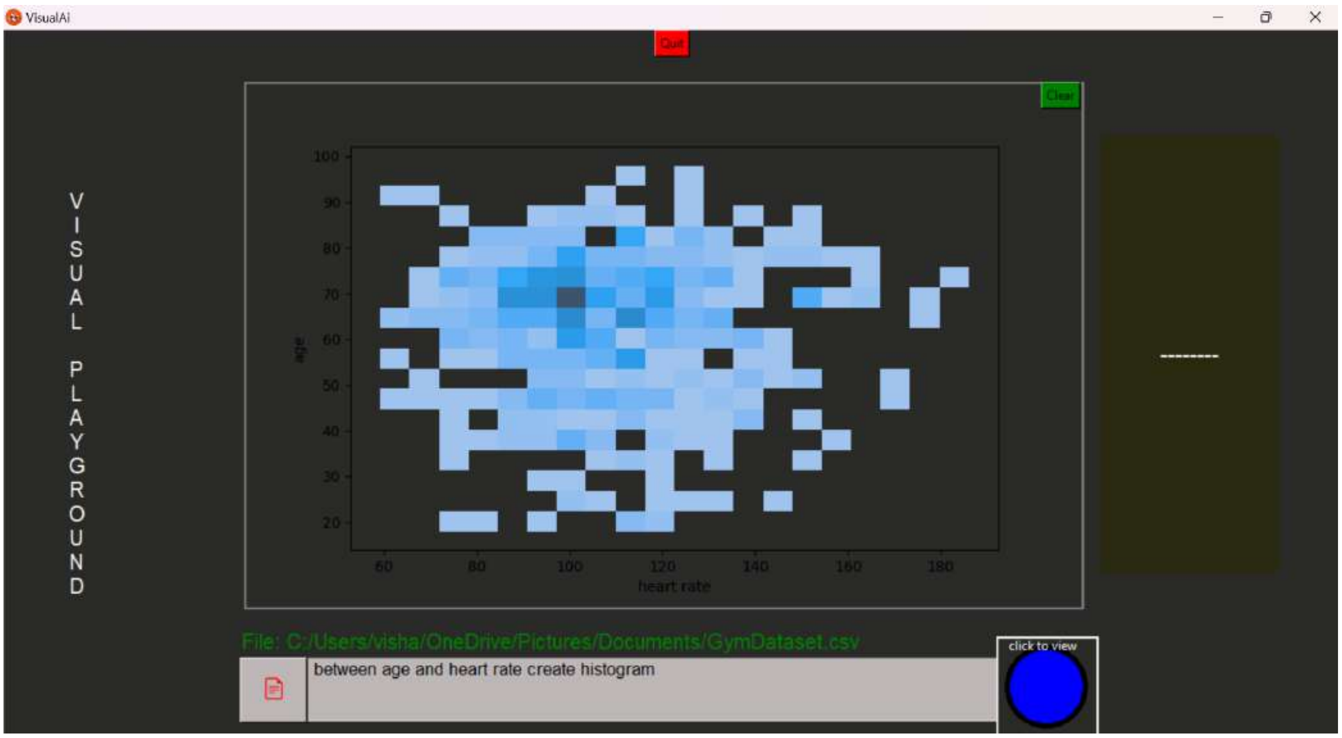
```
!!! INVENTORY AND SUPPLIES !!!

                    ⚲ Adults (19-50 years)

⸙ Barbell                  -⸙Weight lifting and strength training
⸙ Boxing Gloves                 -⸙Boxing and self-defense
⸙ Elliptical Trainer             -⸙Low-impact cardio
⸙ Exercise Ball               -⸙Core workouts and balance
⸙ Kettle bell               -⸙Functional training and power
⸙ Pilates Reformer               -⸙Core strength and flexibility
⸙ Rowing Machine                -⸙Total body workout
⸙ Spin Bike                -⸙High-intensity interval training (HIIT)
⸙ TRX Suspension Trainer              -⸙Full-body resistance training
⸙ Yoga Mat                 -⸙Flexibility and relaxation

                    ⚲ Seniors (50+ years)

⸙ Aqua Aerobics                 -⸙Low-impact water exercise
⸙ Bowling Ball                -⸙Bowling and social activity
⸙ Light Dumbbells               -⸙Strength maintenance
⸙ Recumbent Bike                -⸙Low-impact cardio
⸙ Resistance Chair               -⸙Strength and stability exercises
⸙ Stretch Bands               -⸙Gentle stretching and flexibility
⸙ Tai Chi                  -⸙Balance, flexibility, and relaxation
```

- **Analysis part:**



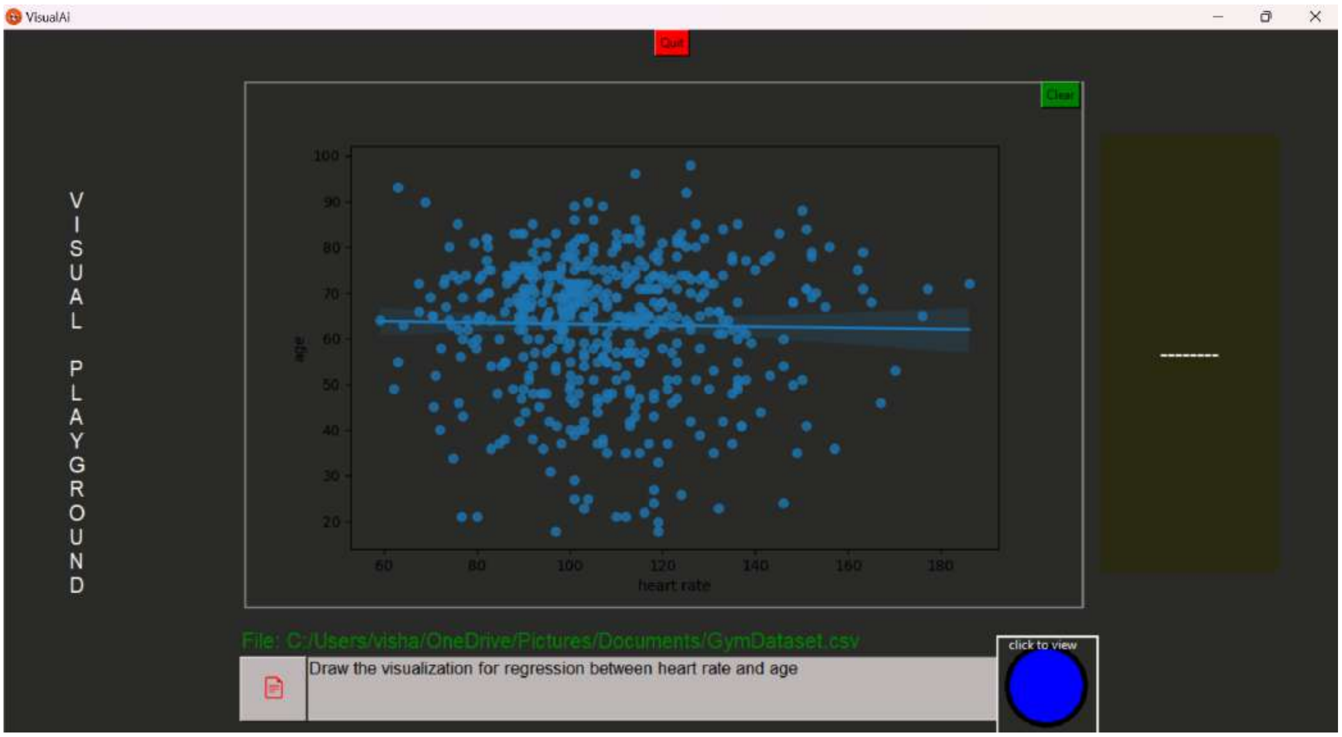
File: C:/Users/visha/OneDrive/Pictures/Documents/GymDataset.csv

make bar plot between age and gender



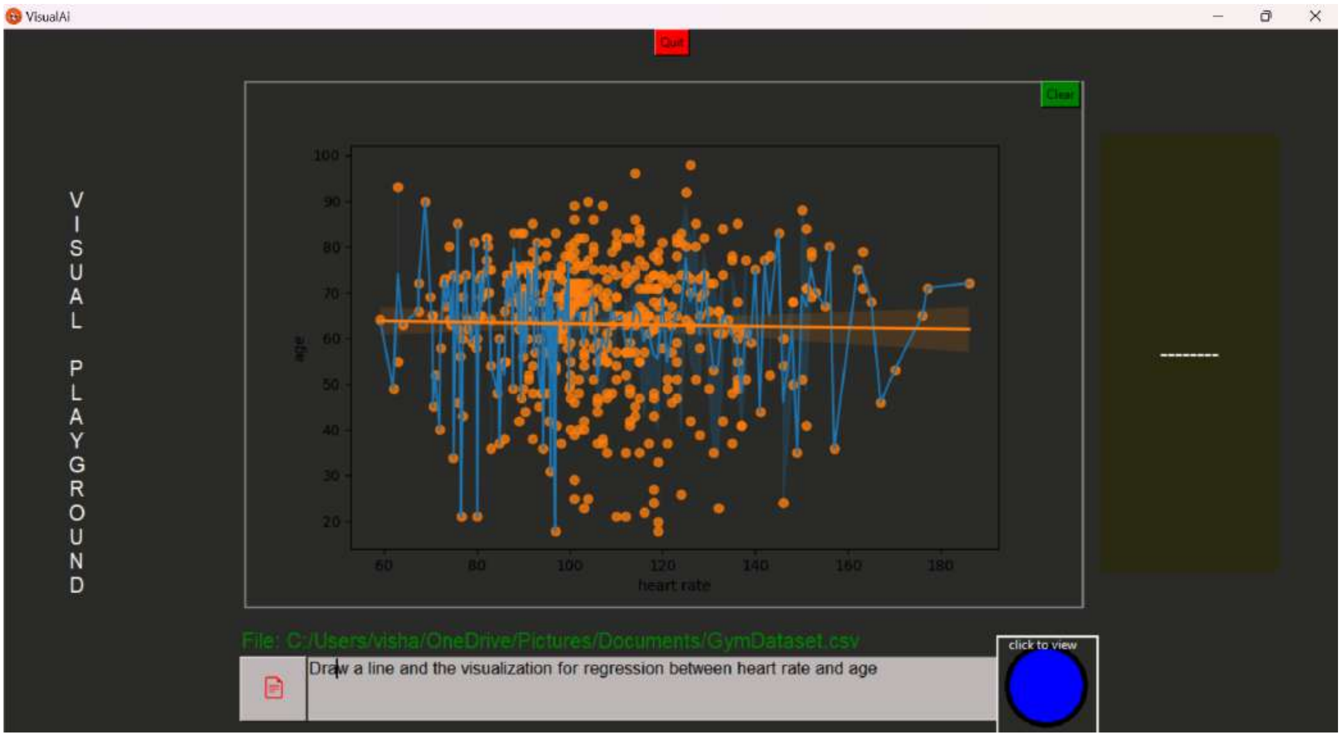
File: C:/Users/visha/OneDrive/Pictures/Documents/GymDataset.csv

between age and heart rate create histogram



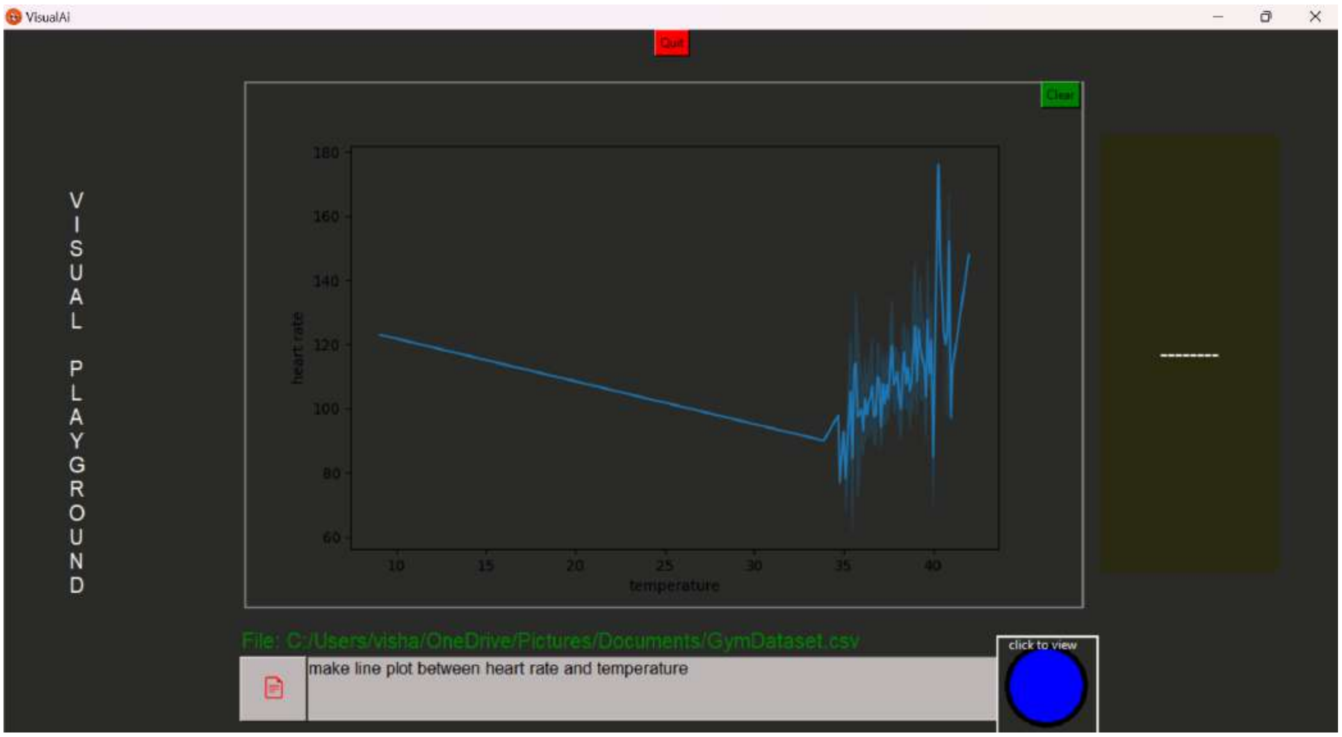
File: C:/Users/visha/OneDrive/Pictures/Documents/GymDataset.csv

Draw the visualization for regression between heart rate and age



File: C:/Users/visha/OneDrive/Pictures/Documents/GymDataset.csv

Draw a line and the visualization for regression between heart rate and age



File: C:/Users/visha/OneDrive/Pictures/Documents/GymDataset.csv

make line plot between heart rate and temperature