Government Polytechnic Kolhapur

शासकीय तंत्रनिकेतन कोल्हापूर

(First Autonomous Institute of Government of Maharashtra)

DEPARTMENT OF INFORMATION TECHNOLOGY

A MICROPROJECT ON

"GYM ADMINSTRATOR SYSTEM"

Submitted By-

Sr no.	Name	Roll no.
1	Pranav Ankush Naik	236036

Under the guidance of-
Prof. K. A. Chavan

Course: Python Programming

Course ID: ITH401

Academic year (2024-2025)

<u>CERITIFICATE</u>

This is to certify that the Mr. Pranav Ankush Naik, student of Second Year Information Technology have completed their micro project work on **"GYM ADMINSTRATOR SYSTEM".** This project is submitted in partial fulfilment of the requirements for the award of Diploma in Information Technology, carried out under the supervision of Prof. K. A. Chavan, Lecturer, Department of Information Technology, Government Polytechnic, Kolhapur, during a period of the academic year 2024-25, as prescribed in curriculum of MPECS - 2023.

Place: Kolhapur

Date: _____

Prof. K. A. Patil   **Prof. R. B. Varne**
(Project Guide)   **(Head of Department)**
Dept. of   **Dept. of**
Information Technology   **Information Technology**

ACKNOWLEDGEMENT

I would like to express my deep and sincere gratitude to my project guide, **Prof. K. A. Chavan**, for their invaluable guidance throughout the course of this project. I am also deeply thankful to all the individuals who contributed to this work in various ways, offering their unwavering support from the very conception of the idea.

The four essential elements of any successful endeavor are **dedication, hard work, patience, and proper guidance**. I extend my heartfelt thanks to my project coordinator for their consistent support in ensuring the successful completion of this Python-based project.

I would also like to express my gratitude to our Head of Department, **Prof. R. B. Varne**, for their encouragement and support.

I am equally grateful to my classmates for their cooperation and encouragement throughout this journey. Lastly, I would like to thank all the teaching staff of the **Information Technology Department** for sharing their knowledge and providing valuable suggestions that greatly helped me improve the quality of my project.

TABLE OF CONTENTS

- ## *Introduction:*

The **Gym Adminstrator System** is a microproject designed to simplify and streamline the management of gym members, monitor their fitness progress, and handle membership plans efficiently. It is a user-friendly application that automates the process of registering members, tracking their goals, managing subscriptions, and facilitating communication via email or SMS. The system enhances the overall experience for both administrators and members through an intuitive interface and key automation features

Objective:
- o **Simplified Member Registration & Management:** Admins can register new members, manage existing profiles, and update personal or health-related information with ease.
- o **User Authentication and Security:** Members and admins are authenticated using login credentials to ensure secure access to personal and system data.
- o **BMI & Calorie Calculation:** The system provides tools to calculate BMI and estimate daily calorie requirements, helping members track their fitness progress.
- o **Membership Plan Management:** Admins can create, modify, and assign membership plans based on duration, cost, and features.
- o **Email & SMS Notifications:** The system can send automated email or SMS updates for reminders, promotions, or important notifications to members.

Scope:
The **Gym Adminstrator System** microproject offers a foundational framework for managing gym operations effectively. It supports functionalities such as user registration, secure login, BMI tracking, subscription handling, and automated communications.

For administrators, it provides full control over user management, plan creation, and tracking member activity. The system is developed using **Python** for its backend logic and a database to store user, plan, and activity information.

Future enhancements could include **progress tracking with**, and **a mobile app interface** for easier member access.

Methodology

1. Requirements Gathering

The initial phase focused on identifying and documenting the functional and non-functional requirements necessary for the development of a streamlined **Gym Adminstrator System.**

- **Functional Requirements:**

  - **User registration and login:** Members should be able to create accounts, log in securely, and manage their profiles.

  - **Member management:** Admins should have the ability to register members, update user details, and manage membership statuses.

  - **BMI and calorie tracking:** The system must allow users to input their physical data and calculate BMI and daily calorie requirements.

  - **Membership plan management:** Admins can create, assign, and update subscription plans based on different packages and durations.

  - **Communication:** The system should enable email or SMS notifications for sending reminders or updates to users.

- **Non-functional Requirements:**

  - **Usability:** The web interface should be clean, intuitive, and easy to navigate for both admins and members.

  - **Reliability:** All calculations (e.g., BMI) and data management functions should return accurate results and prevent data inconsistencies.

  - **Security:** The application should ensure secure access with proper session management and data privacy measures.

  - **Performance**: The system should perform efficiently, handling multiple users without noticeable lag or delays.

2. System Design

In this phase, the overall architecture and flow of the system were defined. The system was designed with a focus on modularity and ease of use, leveraging modern web technologies and lightweight storage.

- **Database Design:**
- Even though you're using a lightweight sqlitedict key-value store instead of a traditional RDBMS, it's still important to define how your data will be structured.
- The database is designed to store and manage key information related to gym operations, including user profiles, BMI records, membership plans, and communication logs.
- **Typical key-value entries in sqlitedict might include:**
  - **Users:**
  user_id → {name, email, password, phone_number, age, gender, height, weight}–
  - Membership Plans:
  plan_id → {plan_name, duration, cost, description}
  - **BMI Records:**
  user_id → [{date, height, weight, bmi, calories}, {…}]
  - **Messages (for notifications):**
  message_id → {to_email, content, date_sent}

- ──────────────
- **User Interface Design:**
- The UI is crafted to be intuitive and responsive using HTML, CSS, and JavaScript, and rendered dynamically using Jinja2 templates.
  - Member Interface
  - Registration and login pages
  - Personal dashboard with fitness stats and plan details
  - BMI calculator form
  - Subscription plan selection and history
  - Admin Interface:
  - Add/update/delete users
  - Create and manage membership plans
  - View BMI records and user activity
  - Send messages or reminders via email/SMS

- ──────────────
- • **Main Modules:**
  - **User Authentication Module:**
  Handles secure registration and login of users using hashed passwords and session management via Flask.

- **BMI & Calorie Tracker Module:**

Allows members to input weight and height, then calculates BMI and estimates daily calorie needs using standard formulas.

- **Membership Plan Module:**

Enables admins to create, assign, and update membership plans for users based on different durations and benefits.

- **Communication Module:**

Sends automated notifications to users via email or SMS for membership updates, reminders, or progress milestones.

- **Admin Dashboard Module:**

Gives administrators control over user data, plans, and system monitoring via a secured backend interface.

- **3. Technology Stack**

Choosing the right tools was essential to build a lightweight, efficient, and scalable gym management solution.

- **Frontend:**
  - **HTML & CSS:** For page structure and styling
  - **JavaScript:** For dynamic content and form validation
- **Backend:**
  - Python (Flask): To handle server-side logic, routing, and processing form inputs
  - Jinja2: For template rendering and connecting backend logic with frontend views
- **Database:**
  - **sqlitedict:** A simple, persistent, key-value store built on top of SQLite, ideal for lightweight projects without the overhead of a full RDBMS

- ---

- **4. Development**

- This phase involved writing and integrating the frontend, backend, and data storage components into a complete working application.
- **Frontend Development:**
  - Designed the user interface with clear navigation, user-friendly forms, and responsive elements.
  - **Pages included:** Login, Register, Dashboard, BMI Calculator, Plan Selection, and Admin Panel.
- **Backend Logic Implementation:**
  - Authentication: Implemented secure login and session handling.
  - **BMI Calculation:** Developed logic for dynamic BMI and calorie calculation based on user input.

- o **Plan Management:** Admin-side functionalities for creating, updating, and assigning plans.
- o **Notification System:** Integrated email (or SMS logic) to notify users about plan updates or fitness milestones.
- **Data Integration:**
  - o Used sqlitedict to persist user data, plan details, and fitness metrics.
  - o Defined functions to read/write data, validate inputs, and avoid duplication or data loss.
  - o Ensured that all modules interacted smoothly through Flask routes and helper functions.

Aim:

To develop a comprehensive **Gym Adminstrator System** using **Flask, HTML, CSS, JavaScript, Jinja**, and **sqlitedict**, providing a streamlined, efficient, and user-friendly platform for managing gym members, tracking fitness progress, and handling membership plans. The primary objective is to enable gym members to register, log in, calculate their BMI, and track personal fitness goals, while also offering administrators the tools to manage member data, create subscription plans, and oversee the overall functioning of the gym.

The **Gym Adminstrator System** aims to enhance gym operations through the following objectives:

---

1. Member Module:
- Allow users to create accounts, log in securely, and manage their personal profiles.
- Enable users to calculate and track **BMI and daily calorie needs** using input forms.
- Provide access to membership details and allow plan selection based on individual needs.
- Deliver notifications or reminders through email or SMS to enhance engagement and accountability.

---

2. Fitness and Plan Management:
- Implement key features for **BMI tracking**, fitness history, and calorie estimation.
- Offer multiple **membership plan options** with different durations, pricing, and benefits.
- Ensure secure and accurate data storage for member information and health metrics.
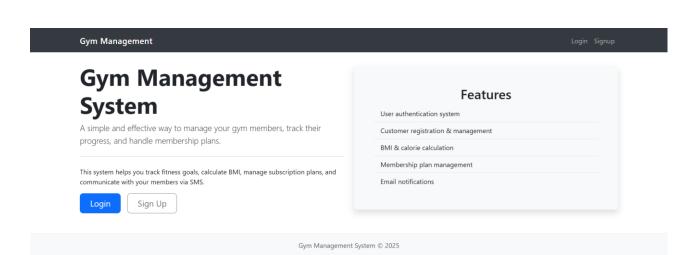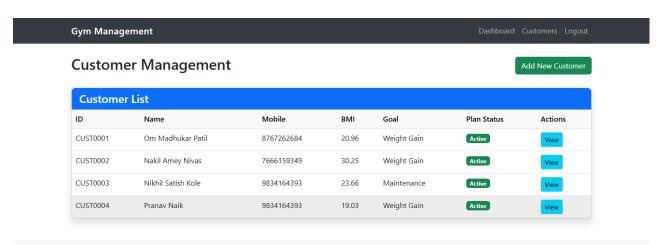
---

3. Administrative Module:
- Provide admins with tools to manage **user accounts, subscription plans**, and **track member progress**.
- Support operations for **adding or updating plans**, reviewing BMI logs, and sending communications.
- Allow oversight of the system's functionality, ensuring smooth performance and data consistency.
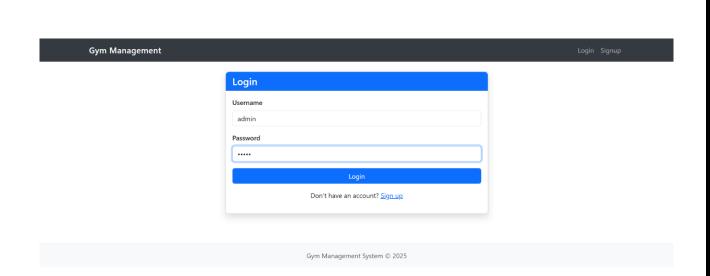
---

4. Data Management:

- Utilize **sqlitedict** to implement a lightweight, key-value based persistent storage system.
- Maintain clear data organization for efficient access to user records, plan details, and BMI logs.
- Ensure data reliability, user authentication, and secure storage of sensitive user information.

❖ *Output:*

Gym Management   Login   Signup

Login

Username

admin

Password

•••••

Login

Don't have an account? Sign up

Gym Management System © 2025

Gym Management   Dashboard   Customers   Logout

Dashboard

Total Customers

4

View Details

Active Plans

4

View Details

Quick Actions

Manage gym operations

Add New Customer

Gym Management System

Welcome to your Gym Management Dashboard

This simple system allows you to:

- Register new customers and collect their details
- Calculate BMI and daily calorie requirements
- Create membership plans (monthly, quarterly, yearly)
- Send Email notifications to customers
- Track customer fitness goals

Manage Customers

Add New Customer

Gym Management System © 2025

❖ *Conclusion:*

The Gym Management System successfully addresses the operational and engagement challenges commonly faced by gym administrators and members. By leveraging a combination of Flask, HTML, CSS, JavaScript, and sqlitedict, the system provides a lightweight yet powerful platform that supports efficient member registration, fitness tracking, and membership plan management.

Through secure user authentication, automated BMI and calorie tracking, and real-time notifications, the system enhances member engagement and promotes a data-driven approach to fitness. Administrators benefit from a centralized dashboard that simplifies user management, plan customization, and progress monitoring.

The modular design and clear separation of concerns between the frontend, backend, and data layers ensure maintainability and potential for scalability. With the foundational framework in place, this system can be further extended to include advanced analytics, mobile integration, and wearable device syncing in future developments—making it a robust solution for modern fitness center management.

<u>ANNEXURE: ASSESSMENT SCHEME</u>

Sr.no	Assessment Parameter	Out of marks	Obtained marks
1.	Technical preparedness for practical	5	
2.	Operating skill/algorithm/flowchart	5	
3.	Observation/Logic/Program	5	
4.	Results/Outputs	5	
5.	Safety/Discipline and Punctuality	5	
	Total	**Out of 25:**	**Signature with date:**