**Names: NIYONZIMA NSHUTI Fabrice**

**ID: 23169**

**Project Documentation: Gym Management System**

* **Requirements of the 4 Models: Admin, Client, Product and Trainer**
* **Admin Model Requirements:**

- Attributes: Id, Name, password.

- Operations: Create, Read, Update, Delete.

- Constraints: Id must be unique.

* **Client Model Requirements:**

- Attributes: Id, membershipId, first name, last name, email, phone number, password, address, address, age, weight.

- Operations: Create, Read, Update, Delete.

- Constraints: MembershipId must be unique.

* **Product Model Requirements:**

- Attributes: Id, product code, product name, description, price, status.

- Operations: Create, Read, Update, Delete

- Constraints: product code must be unique.

* **Trainer Model Requirements:**

- Attributes: Id, CID, first name, last name, email, phone number, password, address, profession.

- Operations: Create, Read, Update, Delete.

- Constraints: CID must be unique.

* Booking Model Requirements:

- Attributes: Date, Time, Doctor (Reference to Doctor Model), Patient (Reference to Patient Model), Room (Reference to Room Model), Medicine (Reference to Medicine Model), Notes.

- Operations: CRUD.

- Constraints: Date and Time must be in the future, Room and Doctor must be available.

* **Project Plan**
* **Phase 1: Planning and Design**

- Define project scope, objectives, and stakeholders.

- Conduct interviews and workshops to gather detailed requirements.

- Create a comprehensive database schema, considering normalization and relationships.

* **Phase 2: Development**

- Set up the development environment, including version control and issue tracking.

- Implement models, controllers, and views for Doctor, Medicine, Patient, and Room.

- Create CRUD functionality for each model, considering input validation and error handling.

- Develop user authentication and authorization mechanisms.

- Implement appointment creation, ensuring data integrity with references.

* **Phase 3: Testing**

- Conduct unit testing for each module, focusing on edge cases.

- Perform integration testing to validate the interaction between modules.

- Conduct system testing to ensure the entire system meets requirements.

- Identify and fix bugs promptly.

* **Phase 4: Deployment**

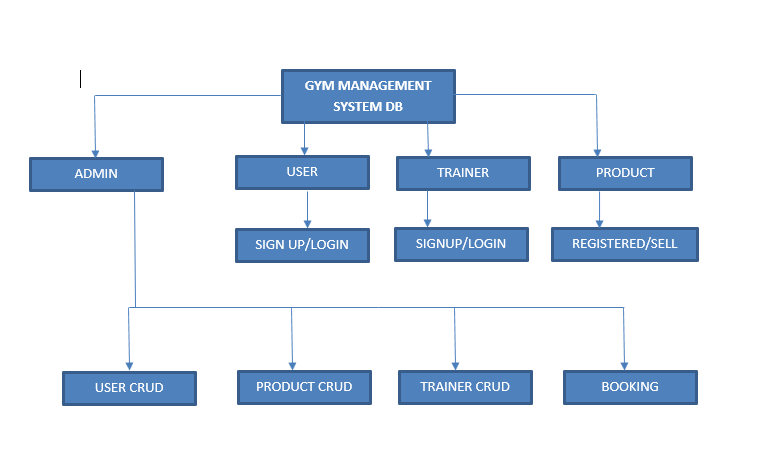
- Deploy the application to a secure hosting environment.

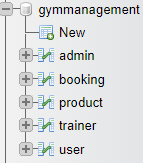
- Configure databases and servers for optimal performance.

- Implement continuous monitoring and error tracking.

- Provide user training and support during the initial rollout.

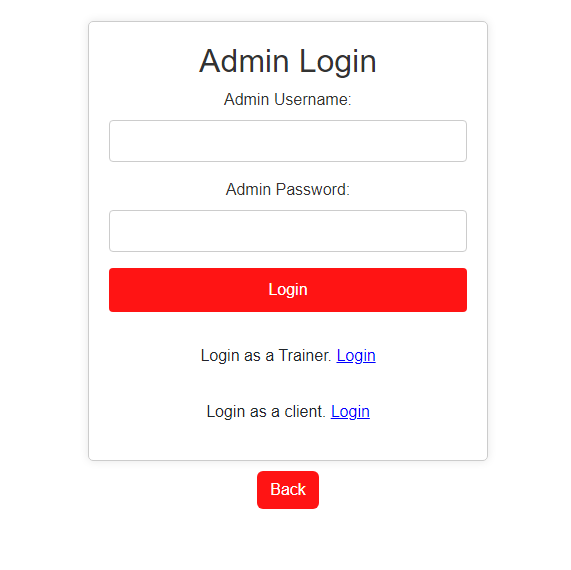
* **Database Schema**



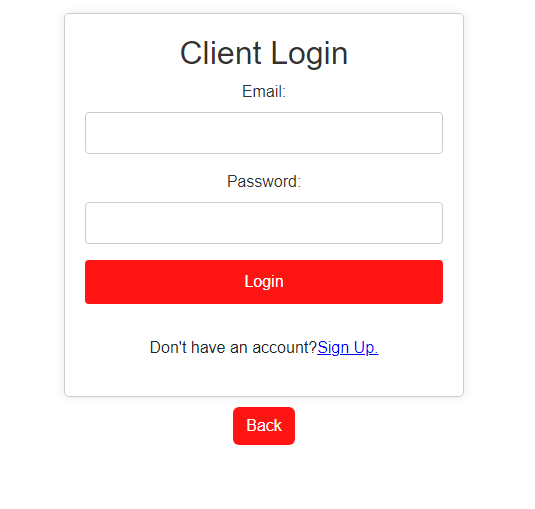


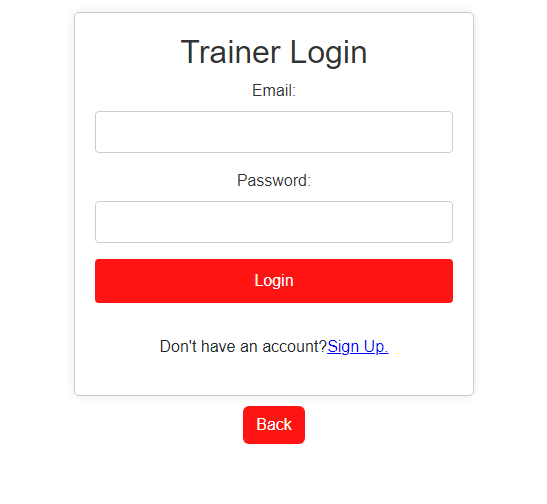
This shows the illustration of database, I used MySQL.

* **User Documentation**
* **Registration and Login:**



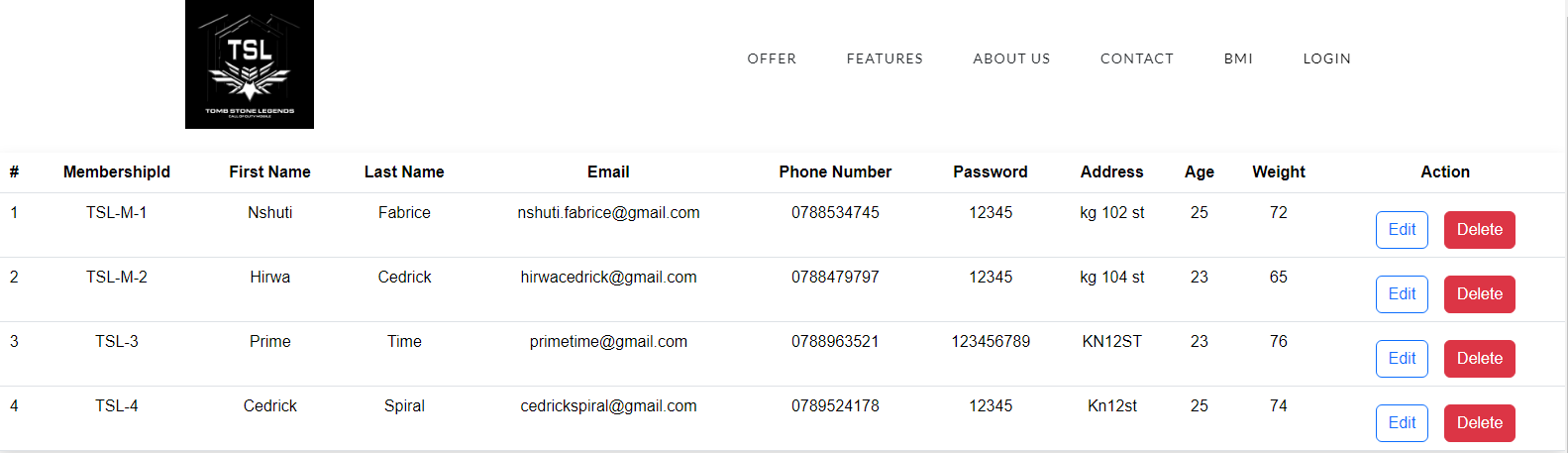
Admin was created with a predefined ID and password for simpler and smoother interaction. Admin can directly manage clients CRUD, Trainers’ and booking CRUD.

****

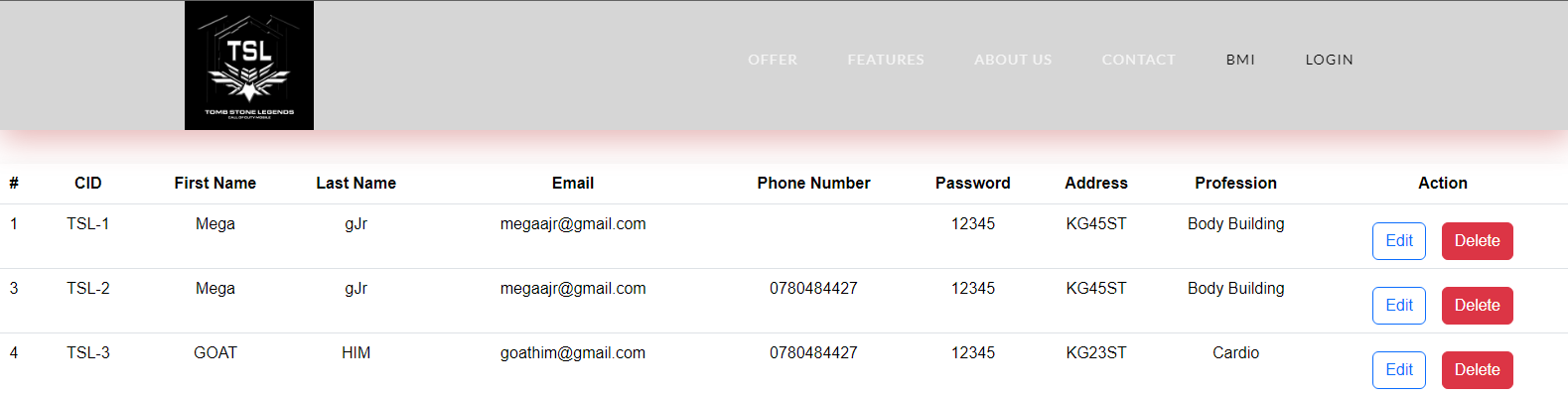
****

* **Tables:**

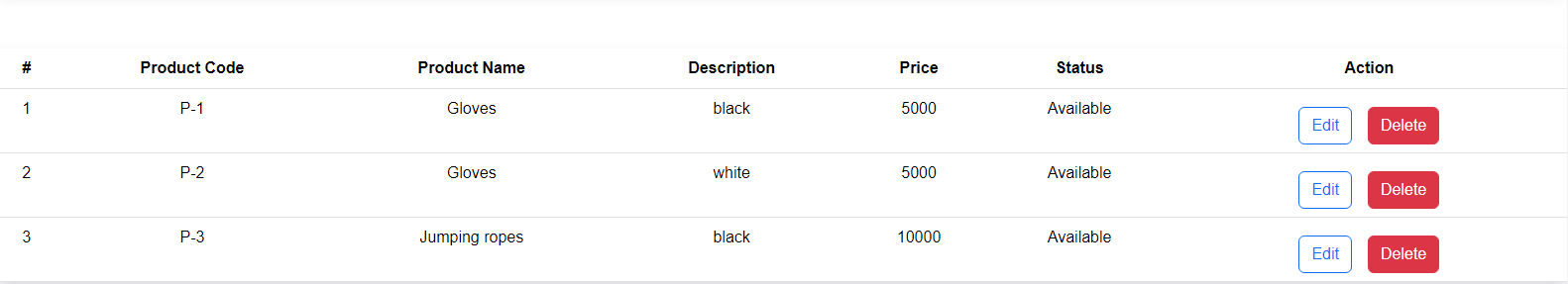
**-Client Table**

****

**-Trainer Table**

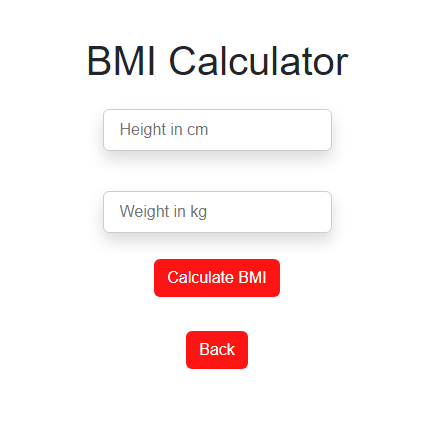


**-Product Table**

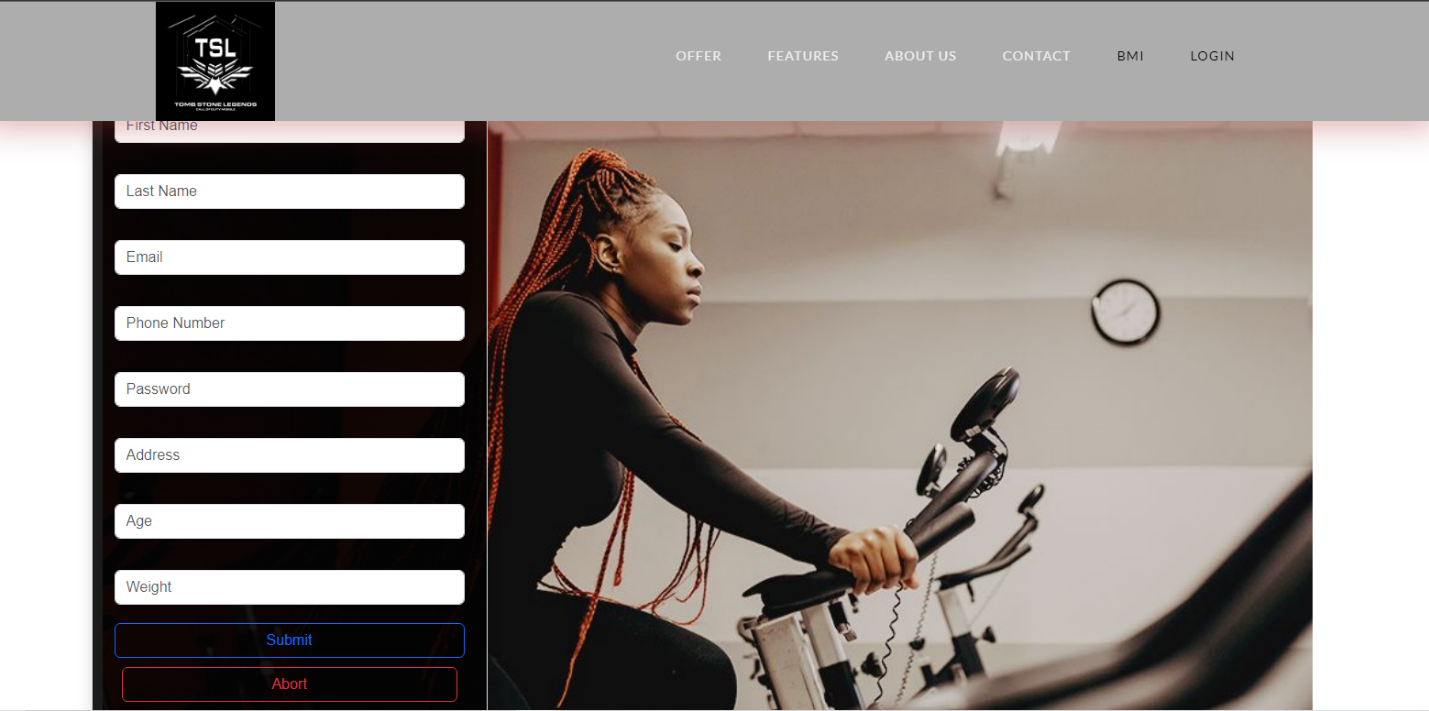


* **BMI Calculator:**

Body mass index (BMI) is a measure of body fat based on height and weight that applies to adult men and women.

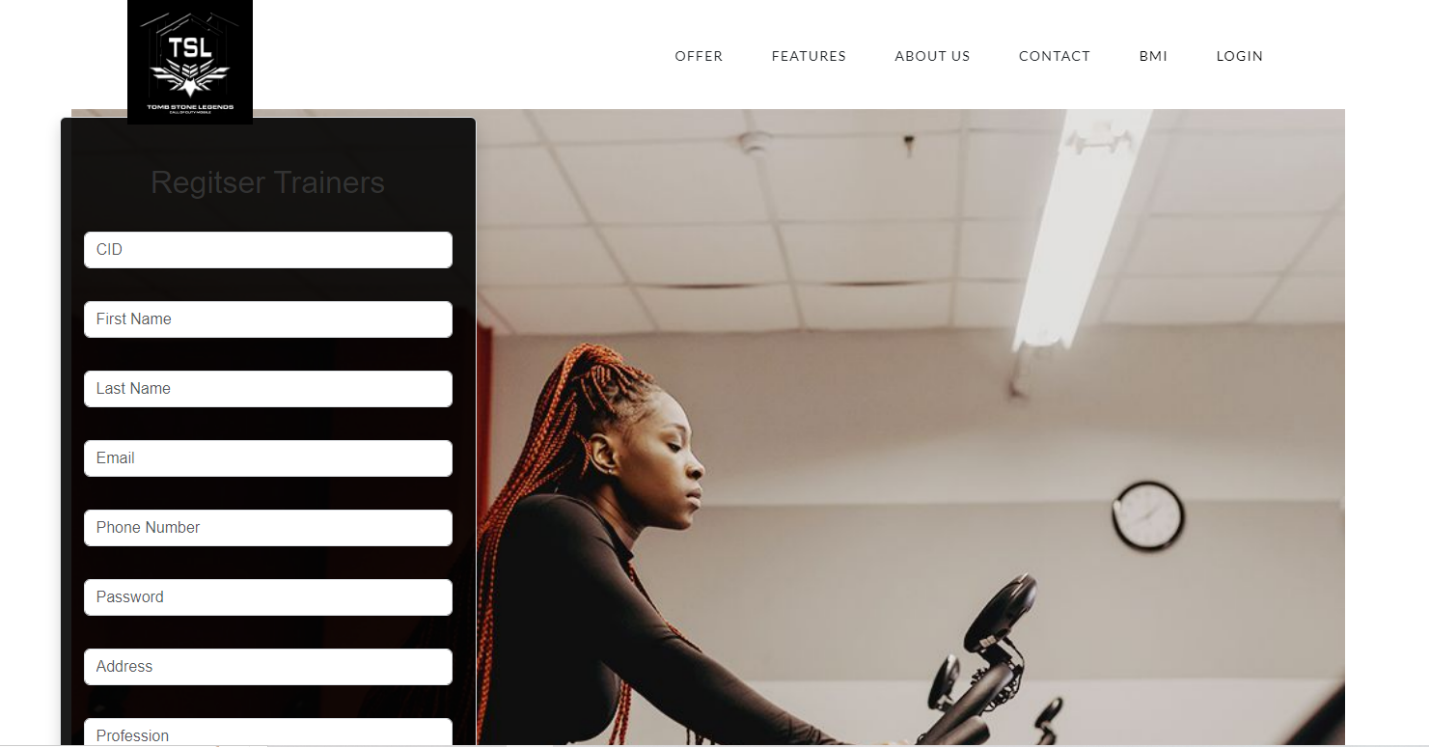


* **Client Sign Up/Register**

****

Here, clients can create a new account.

* **Trainer Sign Up/Register**

****

Here, trainers can create a new account filling all the blanks.

* **Appointment Management:**

- Step-by-step instructions for managing appointments, including creation, viewing, updating, and deletion.

* **Troubleshooting and FAQs:**

- Common issues users may encounter and their resolutions.

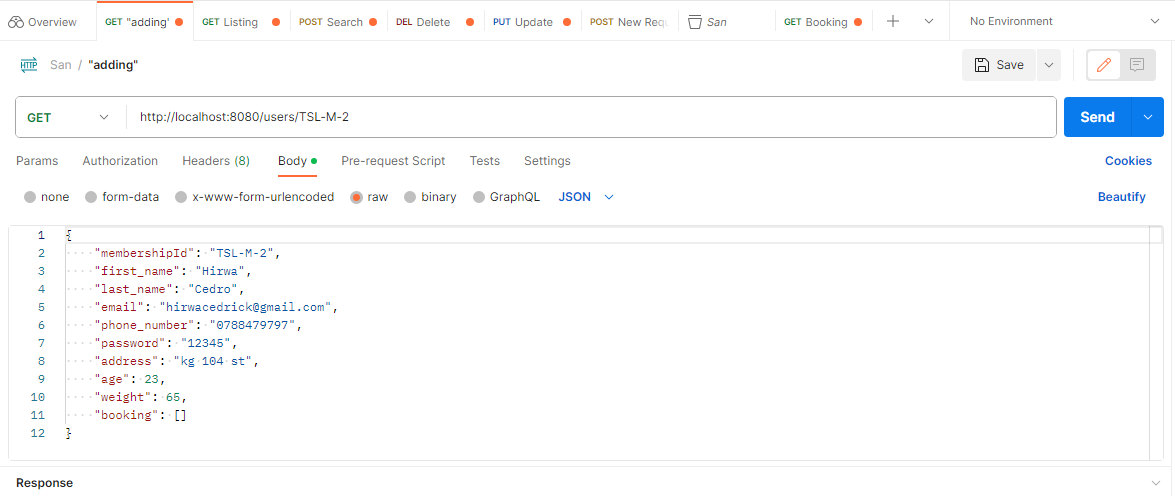
* **Technical Documentation**
* Front-end and Back-end Technologies:

- Overview of technologies used, including React, Node.js, Express, etc.

* **Database Structure:**

- Detailed explanation of table relationships, indexing, and normalization.

* **API Endpoints:**



- http://localhost:8080/users/TSL-M-2: This is used to get the user using membershipId as an example.

* Third-party Libraries and Tools
* **React-router-dom**: Provides the DOM bindings for React Router. It allows you to handle navigation and rendering of components based on the URL.
* **Axios:** Axios simplifies the process of sending asynchronous HTTP requests and handling responses.