FITNESS TRACKER WEB APPLICATION

**Group Members – Group 20**

|  |  |  |
| --- | --- | --- |
| Sr.no | Name | Email |
| 1 | Mayur Gorakh Patil | 2023mt93141@wilp.bits-pilani.ac.in |

Division of Work

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Marks** | **Responsibility** | **Tasks done** |
| Design: | 10% |  |  |
| Backend: | 10% | Mayur Gorakh Patil | done |
| Frontend: | 10% | Mayur Gorakh Patil | done |
| Integration: | 10% | Mayur Gorakh Patil | done |

# **Introduction**

# A fitness tracking website is a web application that enables users to monitor their fitness progress based on their daily activities and view their workout goals. These applications have gained popularity due to their convenience and accessibility in helping users achieve their fitness objectives.

# Fitness tracking websites provide users with the flexibility to choose and customize fitness plans to meet their specific goals, anytime and from anywhere, saving time and resources otherwise spent on traveling to physical training locations. Additionally, these platforms offer users the advantage of easily reviewing and replaying recorded exercises hassle-free.

# By leveraging technology, fitness tracking websites empower users to take control of their fitness journeys, allowing them to monitor their activities, set achievable goals, and track progress effectively. This accessibility and ease of use contribute significantly to the increasing adoption and success of fitness tracking websites in helping users lead healthier lifestyles.

# **Description Of Application**

Fitness Tracking web app has the following components.

* **Entities**
  + Users
  + Application Admin
* **Features**
  + Registration of Users
  + Updating of user details
  + Fitness records tracking
  + user support
  + Payments
  + ~~Rewards and Coupons~~
  + ~~Push Notifications~~

# **User Stories and Acceptance criteria**

## **EPIC 1: Login and Registration**

* 1. **Registration:**

**User Story:** As a new user, I should be able to register

**Acceptance Criteria:**

* Input data fields to enter - 1) Username [email id] 2) Password 3) Basic Details such as Name and contact number.
* Password criteria should be displayed to the user when user clicks inside the password data field.
* If 'Password' entry does not match criteria specified and user hits Submit, show error alert "Password entry does not meet criteria"
* After successful validation of all entered fields and on clicking Submit, show message indicating successful account creation
  1. **Login:**

**User Story:** As a user, I should be able to login with the username and password to the portal.

**Acceptance Criteria:**

* Input data fields to enter - 1) Username[email id] 2) Password
* Indicate invalid usernames and passwords as alerts to the user.
* After successful validation of all entered fields and on clicking Submit, show message “Login Successful” and redirect to the Dashboard page of user.

**Assumptions for EPIC 1[User Stories 1.1 and 1.2]**

The user interfaces for login and signup

It can be assumed that the users are successfully created/logged in when the entered values are valid or Mock APIs can be used.

Any other assumptions made can be listed in the documentation

## **EPIC 2: User Fitness Tracking**

## **User Story 2.1:** As a User, I should be presented with a dashboard page on successful login.

**Description:** The dashboard page should contain my Fitness Track record options to add new record for current day and by Fitness Plan, view previous order details.

**Acceptance Criteria:**

Dashboard with graph of total fitness excursive perform.

**User Story 2.2:** As a user, I should be able to view the available Exercise Plan with its price.

**Description:** On the user selecting plan the required payment details like mode of payment should be displayed. Filtering options to be present.

**Acceptance Criteria:**

The Exercise plan details to be displayed

Pagination to be applied.

# **Design and Architecture, System operations, and services identified.**

The Design and Architecture of the Fitness Tracking system is as follows:

1. **Frontend:**
   * The frontend of the application will be responsible for providing a user-friendly interface to user.
   * The frontend will communicate with the backend server through RESTful APIs to perform actions like display historical record, add details of current exercise perform, placing orders for exercise plan, etc.
   * The UI should be responsive, allowing user to access the application from both desktop and mobile devices.
2. **Backend:**
   * The backend will handle all the business logic, process requests from the frontend, interact with the database, and coordinate with external services for payment processing and order status updates.
   * It can be designed using a microservices architecture to improve modularity and scalability. Microservices will enable independent development and deployment of different application modules.
   * The backend will expose RESTful APIs to be consumed by the frontend and any other potential clients.
3. **Database:**
   * The application's data will be stored in a reliable and scalable database system.
   * For structured data, a relational database like SQL can be used to store information about user, course details, orders, etc.
4. **Authentication:**
   * Implement a robust authentication system to secure customer and restaurant logins using Spring JWT authentication techniques.
   * OAuth 2.0 or JWT (JSON Web Tokens) can be used to handle authentication and authorization.
5. **Fitness Tracking:**
   * Users can log their daily exercise activities, including workouts, cardio sessions, and other physical activities.
   * Each exercise entry can include details such as exercise type, duration, intensity, and any additional notes.
   * Users can view graphical representations (e.g., charts, graphs) of their fitness progress over time.
   * Historical data allows users to analyze trends and adjust their routines accordingly.
6. **Fitness work-out Plan:**
   * Users can subscribe to predefined fitness plans offered by the application.
   * Each fitness plan includes a structured set of exercises and routines tailored to specific fitness goals (e.g., weight loss, muscle gain, endurance).
   * Pricing details and plan descriptions are provided to users for selection.
7. **ARCHITECTURE:**

* Architecture Diagram

1. **REST API endpoints with URI and HTTP methods**

( Swagger Documentation can also be used)

|  |  |  |  |
| --- | --- | --- | --- |
| **Serial No** | **Entity** | **Http Method** | **URI** |
| 1 | User | GET | /profile |
| POST | /register |
| POST | /login |
| PUT | /update |
| 2 | Activity | GET |  |
| POST | **/create** |
| POST | **/listActivities** |
| PUT | **/activity** |
| DELETE | **/delete** |
| 3 | Fitness Plan | GET |  |
| POST |  |
| 4 | Food Menu | GET | **/menus**  **/menus/{menu\_id}** |

**Representation design in request and response**

Sample Request and Response header for the above-mentioned HTTP method:

1. **GET /profile**

* Request:
  + **Method: GET**
  + **URI: /profile**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json**
  + **Body:**

**{**

**"customers": [**

**{**

**"customer\_id": "customer\_123",**

**"name": "Jane smith",**

**"email": "jane.smith@example.com",**

**"phone": "+1234567890",**

**"address": “main road, Mumbai"**

**}**

**]**

**}**

1. **GET /customers/{customer\_id}**

* Request:
  + **Method: GET**
  + **URI: /customers/customer\_123**
* Response:
  + **Status Code: 200 OK**
  + **Headers: Content-Type: application/json**
  + **Body:**

**{**

**"customer\_id": "customer\_123",**

**"name": "John Doe",**

**"email": "john.doe@example.com",**

**"phone": "+1234567890",**

**"address": "123 Main Street, City, Country"**

1. **Frontend UI Design**
2. **Execution Instructions and Assumptions Made (If any)**
3. **GITHUB Repositories**

Frontend:

Backend:

1. **Demonstration Video**