
Flexforge : Real Time Functionalities

1. Member Registration and Login

The member registration and login feature allows users to create an account and access their dashboard securely. During registration, users provide their name, email, and password, which is hashed and stored in the database. The login process verifies the user's credentials, checks for valid payments, and assigns a personal trainer if eligible.

How It Works:

1. **Registration:** Users submit their information; the system validates and stores it in the database.
2. **Login Validation:** Upon login, the system checks the credentials and payment status before granting access.
3. **Trainer Assignment:** If valid, a personal trainer is randomly assigned based on the user's training type.

2.Trainer Registration and Login

The trainer registration and login functionality allows fitness trainers to create accounts and securely access their dashboard. During registration, trainers provide their personal details, expertise, and certifications. The login process validates their credentials, setting up a session to track logged-in trainers.

How It Works:

1. **Registration:** Trainers fill out a form with their details. The system checks for existing trainers and expertise before storing the information securely.
2. **Login Validation:** Upon submitting their email and password, the system verifies their credentials and retrieves their data if valid.
3. **Session Management:** After successful login, the trainer's ID and name are stored in the session for access to restricted areas of the site.

3.User Profile Editing Feature

The user profile editing feature enables users to update their personal information securely, including their name, email, and password. The system checks if the user is logged in before allowing access to the edit profile page. Upon submission, the changes are processed and updated in the database.

How It Works:

1. **Access Profile:** The user accesses the edit profile page, which retrieves their current information from the database based on their session ID.
2. **Submit Changes:** When the user submits the form, the system checks if a password is provided. If so, it hashes the new password before updating the database.
3. **Database Update:** The updated information is committed to the database, and the user is redirected to their personal information page.

4. Progress Tracking Feature

The progress tracking feature allows users to submit their workout progress and enables trainers to view these submissions. Users can log details about their workouts and any issues they encountered, while trainers can access a summary of their trainees' progress.

How It Works:

1. **User Submission:** When a user submits their progress through a form, the system checks if they are logged in and retrieves their assigned trainer's ID.
2. **Data Insertion:** The submitted progress data (date, workout details, issues) is then inserted into the `progress_tracking` table in the database.
3. **Trainer View:** Trainers can access the `/trainer-progress` route, which fetches all progress entries associated with their ID and displays them on the trainer's progress page.

5. Forgot Password Feature

The "Forgot Password" feature allows users to request a password reset link via email. Upon receiving the link, users can set a new password securely.

How It Works:

1. **Email Verification:** Users submit their email on the `/forgot_password` page. The system checks if the email exists in the database. If found, a unique token is generated and stored in the `password_reset` table. An email containing the reset link is sent to the user.
2. **Reset Link Access:** Users click the link in the email, which directs them to the `/reset_password/<token>` page. This token verifies their identity and allows them to set a new password.
3. **Password Update:** After submitting the new password, the system updates the user's password in the database, removes the token from the `password_reset` table, and redirects them to the login page.

6.Feedback Feature

The feedback feature allows users to provide ratings and comments on their experiences, helping improve the service.

How It Works:

1. **Feedback Form Access:** Users must be logged in to access the feedback form. If not logged in, they are redirected to the login page.
2. **Feedback Submission:** Upon submitting the form, the user's ID, rating, category, feedback, and current timestamp are collected and stored in the feedback table in the database.
3. **Thank You Page:** After successful submission, users are redirected to a thank-you page confirming that their feedback has been received.

7.Payment Processing Feature

This feature handles user email verification, membership access, and UPI payment processing. Here's how it works:

1. Check Email

- **Route:** /check_email
- **Method:** GET, POST
- **Functionality:**
 - Users submit their email.
 - The app checks if the email exists in the user table.
 - If the email exists, it stores it in the session and redirects to the membership page.
 - If the email doesn't exist, it redirects to the registration page.
 - Errors during database operations are logged and shown on an error page.

2. Membership Page

- **Route:** /membership
- **Functionality:**
 - Displays membership options if the user is authenticated via session.

- If the user's email isn't found in the session, it redirects them back to the email check page.

3. UPI Payment Page

- **Route:** /upi_payment
- **Functionality:**
 - Renders the UPI payment form for users to input payment details.

4. Handle UPI Payment

- **Route:** /handle_upi_payment
- **Method:** GET, POST
- **Functionality:**
 - On form submission, it captures payment details such as training_type, tier, upi_id, amount, and email.
 - It inserts these details into the payments table.
 - If the payment is processed successfully, it displays a thank-you page; otherwise, it rolls back the transaction and shows an error message.

8. Workout Reminder Feature

The workout reminder feature automatically sends email notifications to users, reminding them about their workout schedules.

How It Works:

1. **Email Sending:** The send_workout_reminder function constructs an email message and sends it to a user's email address with a personalized reminder.
2. **User Fetching:** The workout_reminder_job function connects to the database, retrieves user names and emails, and calls the send_workout_reminder function for each user.
3. **Scheduling:** The reminders are scheduled to be sent daily at 6 AM using the BackgroundScheduler from the APScheduler library.