Project Report: Fitfusion

Adish Shah (200020012), Padakanti Akshay Kiran (200050093),
Vivek Veer (200050158), Arya Amin (200050014)

May 1, 2023

1 Link of Final Repo

https://github.com/aryaamin/FitFusion

2 Overview of Application

Fitfusion is a fitness website that allows users to connect with fitness trainers, and track their fitness progress. The website will feature a user-friendly interface and will be accessible on web platform. The application will allow users to create their fitness goals and find fitness trainers who can help them achieve these goals. Trainers can create workout plans and track the progress of their trainees.

3 Users

3.1 Trainers

Fitness trainers will be able to create and manage their profiles, including their personal details, etc. They will also be able to create customized workout plans for their trainees and track their progress.

3.2 Dieticians

Fitness dieticians will be able to create and manage their profiles, including their personal details, etc. They will also be able to create customized diet plans for their trainees and track their progress.

3.3 Trainees

Trainees will be able to create and manage their profiles, including their fitness goals and progress. They will be able to connect with fitness trainers who can help them achieve their goals and track their progress toward their fitness goals.

4 Data to be Stored

The following entities will be stored in the Fitfusion database:

4.1 User

The user entity will contain information about each user, including their name, email address, and password. It will also include their fitness goals and chosen trainer.

4.2 Trainer

The fitness trainer entity will contain information about each fitness trainer, including their personal details and trainees.

4.3 Dietician

The fitness dietician entity will contain information about each fitness dietician, including their personal details and trainees.

4.4 Workout Plan

The workout plan entity will contain information about the customized workout plans created by fitness trainers for their trainees.

4.5 Diet Plan

The diet plan entity will contain information about the customized diet plans created by fitness dieticians for their trainees.

4.6 Progress

The progress entity will contain information about the progress of each trainee towards their fitness goals.

5 Application Structure

5.1 Backend

The backend of the Fitfusion website will be built using NodeJS along with PostgreSQL and Express for managing sessions. The main API functionality will include:

- User registration
  - Endpoint : /register
  - Method : POST
  - Request Parameters:
    - email (string): The email address of the user.
    - password (string): The password of the user.
    - role (string): Trainee or trainer.
    - Other relevant information like height, weight, fitness goal and current activity level if role is trainee.
  - Return value: Success code if request is successful, error message if not.
- User login
  - Endpoint : /login
  - Method : POST
  - Request Parameters:
    - email (string): The email address of the user.
    - password (string): The password of the user.
  - Return value: Success code if login is successful, error message if not.

Trainee specific

- User account management
- Fitness trainer management
- Trainee management
- Workout plan management
- Progress tracking

5.2 API calls

The following API calls will be implemented in the backend

- **Login**:
  - **Endpoint**: /login
  - **Method**: POST
  - **Request Body**: username, password
  - **Response Body**: access_token, token_type
- **Logout**:
  - **Endpoint**: /logout
  - **Method**: GET
  - **Request Headers**: Authorization: Bearer access_token
  - **Response Body**: message
- **Create User**:
  - **Endpoint**: /register
  - **Method**: POST
  - **Request Body**: name, email, password, age, gender
  - **Response Body**: status codes
- **Get User Profile**:
  - **Endpoint**: /getuserinfo
  - **Method**: GET
  - **Request Headers**: Authorization: Bearer access_token
  - **Response Body**: id, username, email, first_name, last_name, date_of_birth, profile_picture_url, ...
- **Information about Trainer or Dietician**:
  - **Endpoint**: /trainerinfo or /dieticianinfo
  - **Method**: GET
  - **Request Body**: trainee_id
  - **Response Body**: Flags like available, assigned and corresponding info like name,age,gender
- **List of Available Trainers or Dieticians**:
  - **Endpoint**: /availabletrainers or /availabledieticians
  - **Method**: GET
  - **Request Body**: trainee_id
  - **Response Body**: trainers or dieticians
- **List of Current Trainees** :
  - **Endpoint**: /gettrainees
  - **Accessible for**: Trainer or Dietician
  - **Method**: GET
  - **Request Body**: userid
  - **Response Body**: status codes
- **Update Trainer or Update Dietician**:

  - **Endpoint**: /update_trainer or /update_dietician
  - **Method**: POST
  - **Request Body**: userid,trainer_id or dietician_id
  - **Response Body**: status codes

- **Edit personal information**:
  - **Endpoint**: /editinfo
  - **Method**: POST
  - **Request Body**: name, password, email, age, gender
  - **Response Body**: status codes

- **Create Workout Plan**:
  - **Endpoint**: /addexerciseplan
  - **Method**: POST
  - **Request Body**: startdate, enddate, description
  - **Response Body**: status codes

- **Update exercise**:
  - **Endpoint**: /updateexercise
  - **Method**: POST
  - **Request Body**: inputexercise, dateexercise, duration
  - **Response Body**: status codes

- **Add meal**:
  - **Endpoint**: /addmeal
  - **Method**: POST
  - **Request Body**: startDate, endDate, description
  - **Response Body**: status codes

- **Update Calorie Intake**:
  - **Endpoint**: /updatecalorie
  - **Method**: POST
  - **Request Body**: date, meal, calories
  - **Response Body**: status codes

- **Get exercise or diet plan**:
  - **Endpoint**: /getexerciseplan or /getdietplan
  - **Method**: POST
  - **Request Body**: trainee_id
  - **Response Body**: active flag and plan

- **Track Progress**:
  - **Endpoint**: /gettt/:id
  - **Method**: POST
  - **Request Body**: weight, body_measurements, workout_performance
  - **Response Body**: results, notes

5.3 UI

The web interface for for Fitfusion will be built using ReactJS. The key user interfaces that will be supported include:

- User registration
  - Form elements : Registration form for registration of a user.
  - Links : A link to the login page in case the user has already registered.
- User login
  - Form elements : Login form for logging in.
  - Links : A link to the registration page in case the user has already registered.

Trainee specific

- Trainee dashboard
  - Data displayed: Links to all the important interfaces in the application, such as user profile, trainer profile, workout plan, progress tracking.
  - Form elements: None - the dashboard consists of a grid or list of clickable icons or buttons representing each interface, with a brief label or description of each interface.
  - Links: None - this interface serves as a centralized location for accessing other interfaces.
- Trainee profile
  - Data displayed: User's name, profile picture, fitness goals, and any other relevant user information.
  - Form elements: Ability for the user to edit their profile information, upload a profile picture.
  - Links: Link to the user's dashboard.
- Progress tracking
  - Data displayed: A visual representation of progress over time in terms of weight, reps, and sets for a specific exercise or workout plan.
  - Form elements: Ability for the user to input their progress for specific exercises or workout plans and view their progress over time.
  - Links: Links to the user's dashboard and their assigned workout plans.
- Workout plan
  - Data displayed: Exercises, sets, reps, and any other relevant workout information.
  - Form elements: Ability for the trainer to create and edit workout plans for their trainees, and for the trainee to view their assigned workout plans.
  - Links: Links to the trainee's profile and progress tracking.
- Trainer search
  - Data displayed: A list of trainers who match the trainee's search criteria, along with their profiles and areas of expertise.
  - Form elements: A search bar and filters (such as location, price, specialty, etc.) to allow trainees to narrow down their search. Clickable trainer profiles with information such as name, photo, bio, reviews, and specialties.
  - Links: Links to trainer profiles and the option to contact a trainer to inquire about their services.
- Meal tracking
  - Data displayed: A record of the trainee's daily food intake, including details such as meal times, calorie and nutrient intake, and notes about meal choices and preferences.

  - Form elements: A calendar or diary-style interface where trainees can log their meals and track their nutritional intake. Options to add notes or photos to meals, and to search for foods and recipes to add to the log.
  - Links: Links to progress tracking and trainer profiles, as well as options to export data or generate reports on nutritional intake and progress towards fitness goals.

Trainer specific

- Trainer dashboard
  - Data displayed: Links to all the important interfaces in the application, such as trainer profile, trainee progress view, workout plan creator, and meal tracking.
  - Form elements: None - the dashboard consists of a grid or list of clickable icons or buttons representing each interface, with a brief label or description of each interface.
  - Links: None - this interface serves as a centralized location for accessing other interfaces.
- Trainer profile
  - Data displayed: Trainer's name, profile picture, bio, areas of expertise, qualifications, and reviews.
  - Form elements: Ability for the trainer to edit their profile information, upload a profile picture, and connect with trainees.
  - Links: Link to the trainer dashboard, trainee profiles, and trainee progress tracking.
- Trainee progress view for trainers
  - Data displayed: A visual representation of a trainee's progress towards their fitness goals, including details such as weight loss, strength gains, and other measurements.
  - Form elements: A graph or chart that shows progress over time, with options to adjust the data displayed and compare different metrics. Trainers should be able to view and track the progress of their trainees and provide feedback and recommendations.
  - Links: Links to the workout plan creator interface and the trainee profile of the trainee being tracked.
- Workout plan creator
  - Data displayed: A personalized workout plan for a trainee, including details such as exercises, reps, sets, and rest periods.
  - Form elements: A drag-and-drop interface where trainers can select and customize exercises based on trainees' goals and preferences. Options to adjust the difficulty level, add notes and feedback, and preview the plan before sharing it with a trainee.
  - Links: Links to the trainee progress view interface and the trainee profile of the trainee the workout plan was created for.
- Meal tracking for trainers
  - Data displayed: A log of the meals consumed by a trainee, including details such as calories, macronutrients, and meal types.
  - Form elements: A form where trainers can add and edit meals for their trainees, as well as view a summary of total calorie and macronutrient intake over time.
  - Links: Links to the trainee profile of the trainee being tracked.

6 Fitness App Database Schema

User Table

Column Name	Data Type	Description
`user_id`	integer	Unique ID for each user
`name`	varchar(50)	User's name
`email`	varchar(50)	User's email address
`password`	varchar(50)	User's password
`age`	integer	User's age in years
`gender`	varchar(10)	User's gender (male/female/other)

Trainee Table

Column Name	Data Type	Description
`trainee_id`	integer	Unique ID for each trainee same as user_id from Users
`activity_level`	integer	A value indicating trainee's activity level (1 = sedentary, 2 = lightly active, 3 = moderately
`height`	decimal	User's height in meters
`weight`	decimal	User's weight in kilograms
`goal`	varchar(50)	Trainee's fitness goal (e.g., lose weight, build muscle, maintain weight)

Trainer Table

Column Name	Data Type	Description
`trainer_id`	integer	Unique ID for each trainer
`trainee_id`	integer	ID of the user who is a trainee

Dietician Table

Column Name	Data Type	Description
`dietician_id`	integer	Unique ID for each dietician
`trainee_id`	integer	ID of the user who is a trainee

Calorie Intake Table

Column Name	Data Type	Description
`intake_id`	integer	Unique ID for each calorie intake record
`trainee_id`	integer	ID of the trainee who recorded the intake
`date`	date	Date when the intake was recorded
`meal_type`	varchar(50)	Type of meal (e.g., breakfast, lunch, dinner, snack)
`calories`	integer	Number of calories consumed during the meal

Exercise Table

Column Name	Data Type	Description
`exercise_id`	integer	Unique ID for each exercise record
`trainee_id`	integer	ID of the trainee who recorded the exercise
`date`	date	Date when the exercise was performed
`exercise_type`	varchar(50)	Type of exercise performed (e.g., running, weightlifting, yoga)
`duration`	integer	Duration of the exercise in minutes

Diet Plan Table

Column Name	Data Type
plan_id	integer
trainee_id	integer
start_date	date
end_date	date
plan_description	text

Exercise Plan Table

Column Name	Data Type
plan_id	integer
trainee_id	integer
start_date	date
end_date	date
plan_description	text

7 SQL Schema

```
CREATE TABLE Users (
  user_id INTEGER PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  email VARCHAR(50) NOT NULL UNIQUE,
  password VARCHAR(50) NOT NULL,
  age INTEGER NOT NULL CHECK (age >= 0),
  gender VARCHAR(10) NOT NULL CHECK (gender IN (’male’, ’female’, ’other’)),
  user_role VARCHAR(10) NOT NULL CHECK (user_role IN (’trainee’, ’trainer’, ’dietician’))
);

CREATE TABLE Trainee (
  trainee_id INTEGER PRIMARY KEY REFERENCES Users(user_id),
  activity_level INTEGER NOT NULL CHECK (activity_level >= 1 AND activity_level <= 5),
  height DECIMAL NOT NULL CHECK (height >= 0),
  weight DECIMAL NOT NULL CHECK (weight >= 0),
  goal VARCHAR(50) NOT NULL CHECK (goal IN (’lose weight’, ’gain weight’, ’maintain weight’, ’build muscle
);

CREATE TABLE Trainer (
  trainer_id INTEGER,
  trainee_id INTEGER REFERENCES Trainee(trainee_id)
);

CREATE TABLE Dietician (
  dietician_id INTEGER,
  trainee_id INTEGER REFERENCES Trainee(trainee_id)
);

CREATE TABLE Calorie_Intake (
  intake_id INTEGER PRIMARY KEY,
  trainee_id INTEGER REFERENCES Trainee(trainee_id),
  date DATE NOT NULL,
  meal_type VARCHAR(50) NOT NULL CHECK (meal_type IN (’breakfast’, ’lunch’, ’dinner’, ’snack’)),
  calories INTEGER NOT NULL CHECK (calories > 0)
```

```
);

CREATE INDEX intake_idx ON Calorie_Intake (trainee_id, date);

CREATE TABLE Exercise (
  exercise_id INTEGER PRIMARY KEY,
  trainee_id INTEGER REFERENCES Trainee(trainee_id),
  date DATE NOT NULL,
  exercise_type VARCHAR(50) NOT NULL CHECK (exercise_type IN ('running', 'swimming', 'cycling', 'weightlif
  duration INTEGER NOT NULL CHECK (duration > 0)
);

CREATE INDEX exercise_idx ON Exercise (trainee_id, date);

CREATE TABLE Diet_Plan (
  plan_id INTEGER PRIMARY KEY,
  trainee_id INTEGER REFERENCES Trainee(trainee_id),
  start_date DATE NOT NULL,
  end_date DATE NOT NULL,
  plan_description TEXT NOT NULL CHECK (LENGTH(plan_description) > 0)
);

CREATE INDEX plan_idx ON Diet_Plan (trainee_id, start_date, end_date);

CREATE TABLE Exercise_Plan (
  plan_id INTEGER PRIMARY KEY,
  trainee_id INTEGER REFERENCES Trainee(trainee_id),
  start_date DATE NOT NULL,
  end_date DATE NOT NULL,
  plan_description TEXT NOT NULL CHECK (LENGTH(plan_description) > 0)
);

CREATE INDEX ex_plan_idx ON Exercise_Plan (trainee_id, start_date, end_date);

--Roles
CREATE ROLE dietician_role;
CREATE ROLE trainer_role;
CREATE ROLE trainee_role;

GRANT USAGE ON SCHEMA public TO dietician_role, trainer_role, trainee_role;

GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Users TO dietician_role, trainer_role, trainee_role;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Trainee TO dietician_role, trainer_role, trainee_role;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Trainer TO trainer_role;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Dietician TO dietician_role;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Calorie_Intake TO trainee_role;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Exercise TO trainee_role;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Diet_Plan TO dietician_role;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE Exercise_Plan TO trainer_role;
```

8 Other Aspects

8.1 Test Data

Test data will be generated using a combination of real-world data and synthetic data.

8.2 Security

The Fitfusion website will implement several security measures, including session management and XSRF prevention, to ensure the safety and privacy of user data.

9 Finished Tasks

We have finished all of the functionalities (except minor details) that we had stated in our design document.

10 Future Plans

While we have successfully implemented all of the functionalities we planned for this project, there is always room for improvement and expansion. Some potential future work includes:

10.1 Profile pictures

Users can add profile pictures and they are displayed at relevant pages.

10.2 Reminders

Users get reminders about their exercise plan through mail or notifications.

10.3 Adding food items in a meal

Currently we ask the user to input the calories they consumed, instead we can ask them what they ate and calculate the calories ourselves.

10.4 Mobile friendly

We would like to make our website responsive and mobile friendly.

10.5 Website security

Even though we have our website is secure against some basic attacks, we would definitely like to add security against some of the more advanced attacks.

10.6 Frontend design

The current design isn't the most pleasing to the eye, we would like to make the frontend more aesthetic.

11 Our learnings

During the development of this project, we have learned the following:

11.1 Team Collaboration

Working as a team has taught us the importance of communication and collaboration. We have learned to divide tasks and work independently while keeping each other updated on progress.

11.2 Software Development

We have learned various software development skills such as requirement gathering, system design, implementation, testing, deployment and most importantly the use of git for collaboration.

11.3 Web Development

This project gave us hands-on experience in web development technologies such as HTML, CSS, JavaScript and the PERN stack.