Shoaib Valani
ITP 303

1. Topic and Purpose:

The NuFit Workout Log is an interactive fitness application that serves as a comprehensive guide for individuals aiming to tailor their workout experiences. It supports users ranging from gym novices to seasoned fitness enthusiasts by allowing them to explore, customize, and track gym workouts. The website emphasizes proper exercise form and routine customization, enhancing the user's ability to optimize their fitness gains.

2. Site Functionality and Instructions:

The website features a simple navigation bar with access to 'Home', 'My Workout', and 'Explore' sections. The 'Explore' page provides a search function for exercises categorized by muscle groups, equipment, and difficulty level. Exercise entries, like the Incline Hammer Curl, come with detailed information and visual aids. The 'My Workout' section allows users to create and manage workout routines with options like 'Push Workout', 'Load Workout', and 'Delete Workout', giving users full control over their fitness plans. The site does not detail admin credentials or password-protected pages, indicating a focus on user accessibility.

Instructions:

My Workout Page

- **Loading a Workout**: To load a workout in the 'My Workout' section, users must first select a workout from a list.
- **Managing Workouts**: Users can manage their workouts by:
    - **Adding New Workouts**: Through a form provided in the section.
    - **Deleting Workouts**: Users have the option to delete entire workouts or individual exercises within a workout.
- **Viewing Changes**: After making changes like adding exercises, users can return to 'My Workout' to load the workout and see the updates.

Explore Page

- **Searching for Exercises**: The 'Explore' section allows users to search for exercises using a dropdown menu with four options: Name, Type, Muscle, Difficulty.
    - **Name**:
        - Users can search for exercises by their names.

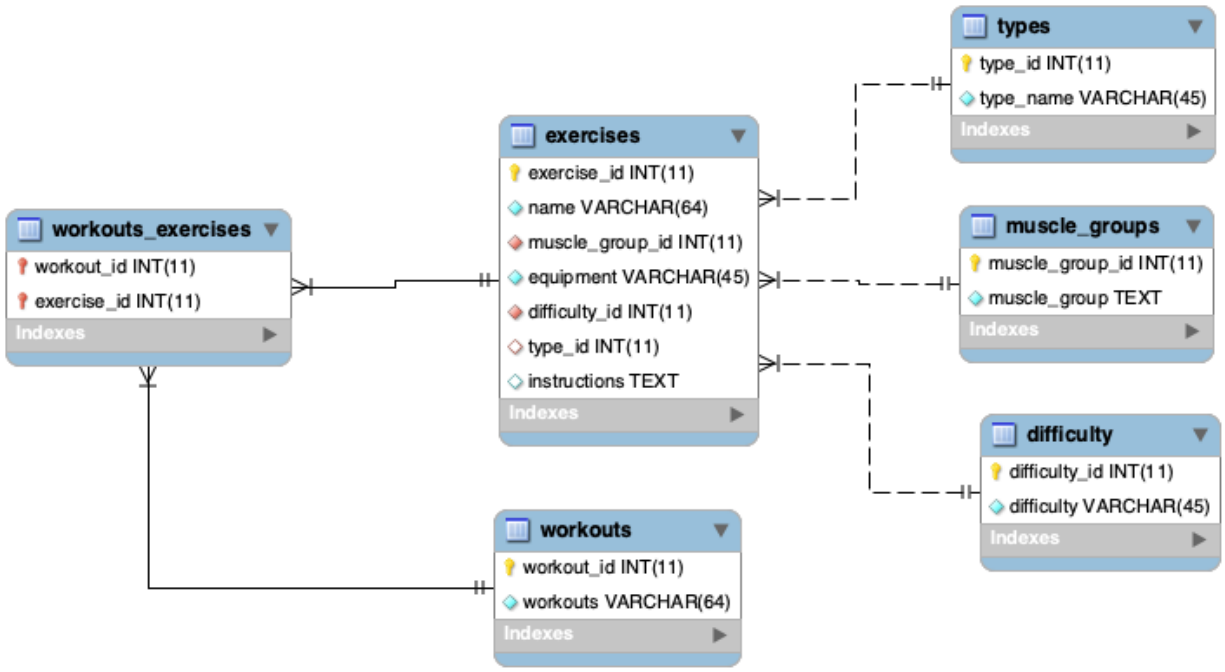
    - The search allows for partial name entries. For example, typing "press" might bring up exercises like "Dumbbell Bench Press".
- **Type**:
    - This refers to the exercise type or category. Users can choose from several types:
        - Cardio
        - Olympic Weightlifting
        - Plyometrics
        - Powerlifting
        - Strength
        - Stretching
        - Strongman
- **Muscle**:
    - Users can search for exercises based on the muscle group they target. The options include:
        - Abdominals
        - Abductors
        - Adductors
        - Biceps
        - Calves
        - Chest
        - Forearms
        - Glutes
        - Hamstrings
        - Lats (Latissimus Dorsi)
        - Lower Back
        - Middle Back
        - Neck
        - Quadriceps
        - Traps (Trapezius)
        - Triceps
- **Difficulty**:
    - This allows users to filter exercises based on their difficulty level. The possible values are:
        - Beginner
        - Intermediate
        - Expert

- **Adding Exercises to a Workout**: Once an exercise is found:
    - Users must select a workout from a dropdown menu in the workout selector.
    - Click the 'Add' button on the desired exercise to include it in their selected workout.

3. Data Source:

Exercise information is sourced from the API-Ninjas Exercises API, offering a vast database of exercise details that users can leverage to inform their workout choices.

4. Database Diagram



5. Extras:

Extra #3: My Integration of the Exercise DB API

- **API Utilization**: I used the Exercise DB API from API Ninjas to retrieve exercise data.
- **Populating Data**: I populated this data using JavaScript (JS) and JSON, effectively integrating it into the website's interface.

Extra #9: Functionality for the 'Add' Button

- **Frontend-Backend Interaction**: To add functionality to the 'Add' button on my 'Explore' page, which is used to select exercises from the API and send them to the SQL database, I implemented a specific process:

- **Data Handling**: I stored the data in a local variable.
- **Sending Data to Backend**: This JSON data is then sent to my backend add_exercises.php file using AJAX from the explore.php file.
- **Backend Processing**: My add_exercises.php file receives the JSON, stores the exercise data in the database, and returns a status code.
- **User Feedback**: Finally, the frontend informs the user whether the addition of the exercise to their workout was successful.

Extra #10: Utilization of Many-to-Many Table Relationship in Database

- **Database Modeling**: In this project, I utilized a many-to-many table relationship, as evident in the DB model diagram.
- **SQL Table Structure**: The diagram indicates a many-to-many relationship between two tables: 'workouts' and 'exercises'.
- **Junction Table**: This relationship is facilitated through the 'workouts_exercises' junction table (also known as a linking table or associative entity), which contains foreign keys referencing the primary keys in both the 'workouts' and 'exercises' tables.

These extras illustrate the technical complexities and backend functionalities I incorporated into the NuFit Workout Log, enhancing the user experience and the efficiency of the application's data handling and storage.

6. API Documentation:

Data for the exercises is obtained from the API-Ninjas Exercises API, which presumably provides comprehensive documentation for accessing and utilizing the exercise data.

7. CSS Frameworks and Templates:

The application uses Bootstrap 4.0 for responsive design and layout, and custom styles are defined within the HTML, including Google Fonts for typography.

For my Bootstrap template, I used "https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"