

# **What is your project about, who is it for, what does it do + Justifications for my design (Research, feedback, best practices):**

## **What is my project? And how will I ensure it works functionally and correctly?**

I want to create a website that others can use to search for exercises and find out more information about them. Because I plan on others using this interface, I will need to ensure it will not be responsible for injuries or unnecessary danger, such as users accidentally using the wrong exercise form and injuring themselves. To do this effectively I will need to ensure I collect correct, safe and up to date information for my exercises. To ensure that my information about exercise is correct, I will use sources like: [Nutrition.gov](#), [Acefitness.org](#), [Harvard.edu](#), which are more reliable than a website with a less official domain name eg: ".com". By focusing on reliability and safety, I ensure that I minimise injury, and that users are satisfied with my product. Even though majority information will be from a reliable url, other websites that use a less trustworthy url may still have good information, and so if needed I may include this information if I am able to research it further and ensure that it is safe.

Some personal trainers maintain that personal evidence is more effective than scientific evidence, and so I may need to take this into account when choosing what information to include in my database. I could use both my own experience, my friends, or could talk to personal trainers at my gym if needed to ensure that I gather multiple different perspectives on the information, as well as real world application.

I will need to ensure that the programming is robust and functional to avoid any unnecessary errors. And by following the standards of the programs that I am going to use, I ensure that my code is likely to work as intended, as well as making it easier to understand, and to update and change if needed.

To do this, I researched common SQL database conventions (source: [SQL Shack](#)), and made note of how I should name the tables and columns in my database.

- Tables should be named in lowercase letters eg: "id" not "ID" or "Id"
- If a name is more than one word long, use an underscore between words eg: "language\_id"
- Include a primary key named "id" on each table
- When using foreign keys use names like "table\_id" eg: "language\_id".

I also researched standard python coding conventions (source: [PEP8](#)), and made note of how I should type my code such as variable naming conventions, function and loop formatting, and various other standards included in the style guide.

- Use 4 spaces or 'tab' when indenting lines of code.
- Limit all lines to a maximum of 79 characters.
- Imports are always put at the top of the file

- Pick between “” and “ ” when declaring strings and stick to it unless necessary.

Because I am making something that other people are going to use, I need to ensure that the programming is robust and functional to avoid any unnecessary errors, by following the standards of the programs that I am going to use, I ensure that my code is likely to work as intended, as well as making it easier to understand, and to update and change if needed. However, using these rigid conventions could make code harder to interact with, and could involve a learning curve that would result in extra time spent on something that may not be entirely beneficial. I will need to balance following conventions with accessible code, and make decisions about if it's worth trying to learn a new method for something I may already understand.

It can be argued that readability and flexibility is more important than following strict design conventions, and so I will try to take these alternative perspectives into consideration when deciding on variable names and other formatting. For example, while it is said to be necessary to limit lines to 79 characters, it is more important that the code is functional, and so if by shortening a line, the code becomes slower, doesn't work at all, or involves me spending more time learning how to fix it then it may not be worth it to follow these conventions.

## **Who is it for?**

My interface will be majority made for gym goers, or any other people who want to learn more about exercise physiology. Based on my research, the average age for gym goers in the UK is 18-34 (according to [IHRSA Health club](#)), so my main target audience will be somewhere around this age range, eg: 15 - 40, and while there will be extremes on either end of that spectrum, more commonly in the upper end, I will try to make my interface as easy to use as possible to ensure that elderly, impaired and my general user base will be able to interact correctly. Although a simpler interface could deter more advanced and experienced gym goers, I think after considering the scope of my project, as well as who my target audience is, I think that focusing on accessibility is more important.

What I found during my research:

- Most common gym going age is 18-34 years old
- Ages 35 to 44, prefer free weights (dumbbells/hand weights), running, and elliptical machines over other forms of exercise.
- Ages 55 to 64 favor walking, stationary cycling, and the use of dumbbells and hand weights.
- Ages 28 and 44, bias yoga, HIIT, and running

Based on these above findings, I think that 16 to 40 years old is a good target age range for my product. I will try to include more dumbbell exercises in my database, as it seems that many prefer using dumbbells, and also especially due to the fact that dumbbell form is much easier to mess up than machines are. I don't think that I need to include anything about walking, or stationary running, as this is just cardio so form does not matter as much, and injury is rare, and no muscles are actually being targeted other than stabilising muscles. I likely also won't include yoga either, as I have little to no experience in that area, and I want my project to be less about cardio and stretching and more about the muscles being used in

exercises, and form. Although this does lower the inclusivity of my project, yoga does not actually work any muscles, and so doesn't actually fit within my database. By understanding who my target audience is, I am able to more effectively design my product to appeal to these users, as well as giving me a better understanding of what users will want me to include, which should help me create a more satisfying and positive experience for users.

### **Why did I choose this project? What problem am I solving?**

It is time consuming to individually search multiple muscle groups manually into google to find exercises that target them, the results may also be biased or unreliable, as websites could pay to boost the engagement on incorrect or inaccurate information. I want to create an unbiased source of information, as well as to make it easier to search through exercises when designing a split by grouping them into one website, where people can quickly and easily find new exercises, or learn more about exercises they already know about. It can also be costly if gym goers need to pay for a personal trainer to recommend them exercises/form, and although I will never be able to fulfill their in person role, I want to help people know where to start when asking for help, or if I'm lucky, they may be able to use my program in place of a personal trainer if what they need is simplistic, which could save money and time.

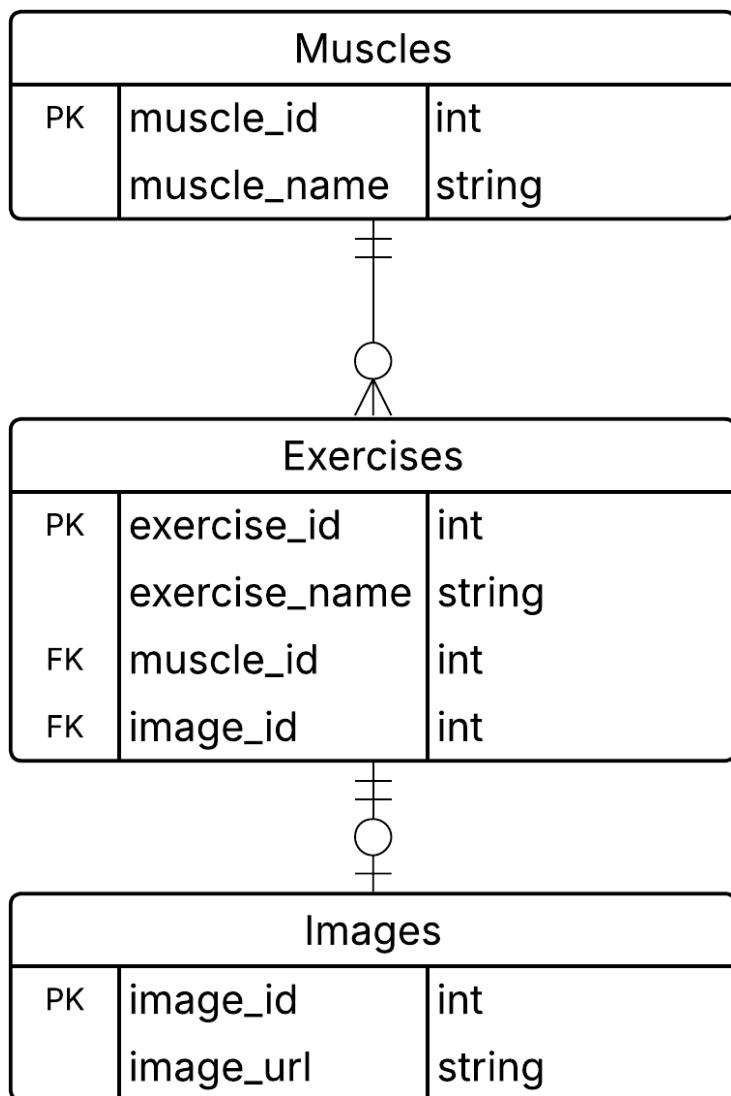
Although I don't want my database to completely replace a personal trainer, and more so offer assistance, some users could choose to follow the advice over a trainer, which could lead to injury. To help minimise this, I will include a disclaimer on my page to ensure that no users make this mistake, and even if they do, I have tried my best to help them avoid injury. By considering all the possible outcomes, although unlikely, I am able to prepare for the worst, and ensure safety for my users.

### **Are there any existing alternatives?**

There are similar solutions already available, such as [ExRx.net](http://ExRx.net), or [functionalmovement.com](http://functionalmovement.com). ExRx is extremely conclusive and takes a very scientific approach, listing the exercises by specific muscles targeted eg: "[Triceps Brachii](#)", rather than just "arms" or "triceps". There are also quite a lot of different links to click, which could seem overwhelming. It does a very good job of explaining information once you navigate to the exercise though, including gif/video and written instructions. Overall, ExRx has a lot of extremely conclusive information, as well as clear instructions, however it isn't very intuitive and by using scientific names, as well as including a large amount of links, it may seem overwhelming to new users, appealing more to advanced gym goers. Functionalmovement takes a different approach, opting to focus more on bodyweight/kettlebell workouts. It has a very aesthetically pleasing and intuitive UI, as well as accompanying thumbnail photos for each exercise. However the information is quite limited, and it also focuses more on stretching exercises rather than muscle training, so it fulfills a different niche that I don't think I will focus on. Overall, I like the amount of information ExRx stores, as well as its clear instructions and explanations, however I dislike the inaccessibility due to advanced terminology and overwhelming UI, in comparison, I like the UI and layout of Functionalmovements as well its images and accessibility, however it doesn't include muscular information. I think that I will take inspiration from ExRx's information and instructions, and take inspiration from the aesthetic design of Functionalmovements, and try to avoid including their negatives.

# Relationship, Diagram, and Flowchart:

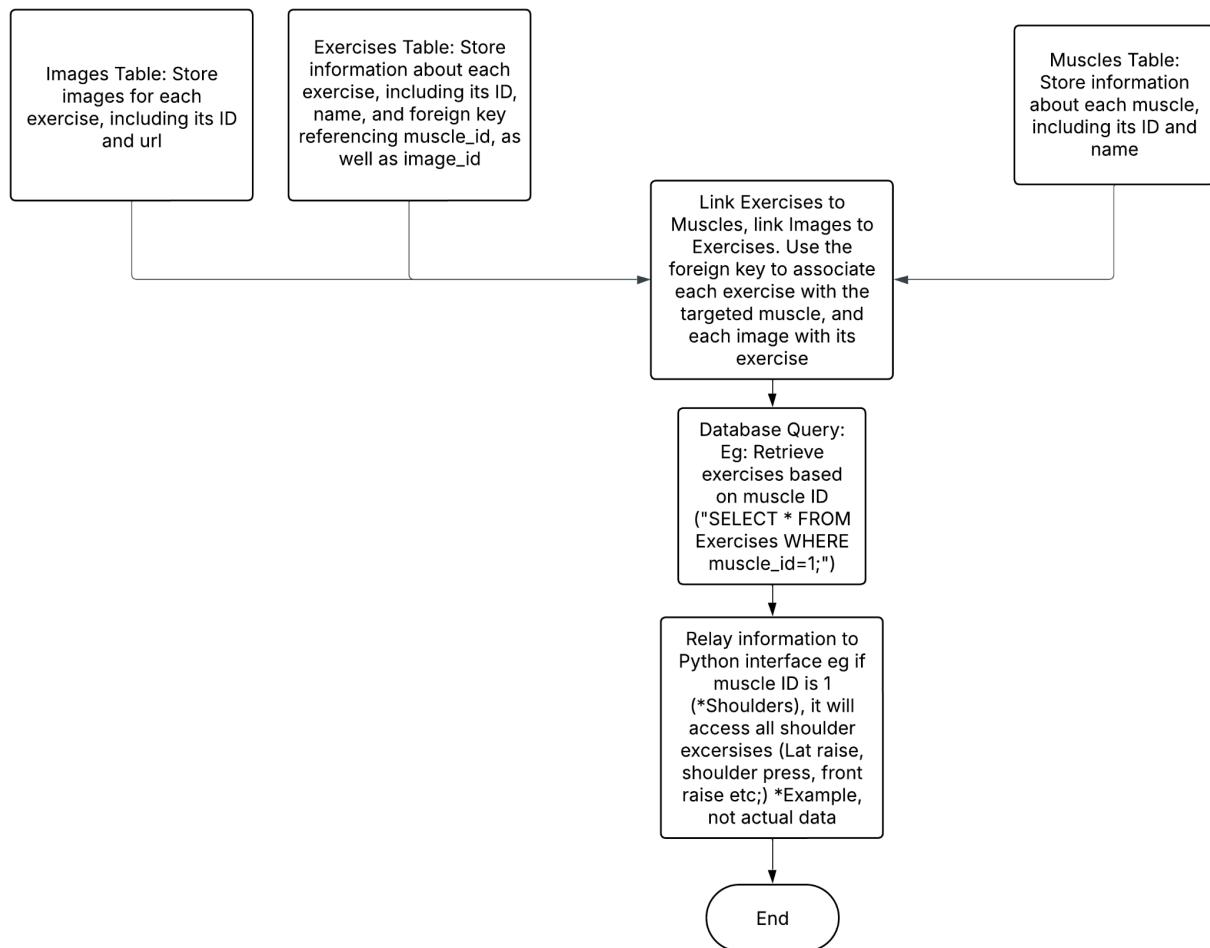
## Entity Relationship Diagram:



## Muscle and Exercise Relationship:

The database structure consists of three tables: Muscles, Exercises, and Images. Each exercise is linked to a specific muscle through a foreign key relationship, indicating which muscle is targeted by the exercise, each exercise is linked to an image through an ID foreign key.

## Muscle and Exercise Database Flowchart:



## Routes, Function Signatures and SQL Queries:

### Route:

“@app.route(“/”)” will be the homepage, which will allow you to access the search page, a list of all included exercises, and a list of all included muscles.

### SQL Query:

N/A. No SQL at this stage, will just serve as a hub to access the other pages. (May change in future, could show a summary of all information or glossary)

**Function Signature:**

```
def home():
```

**Route:**

@app.route("/all\_exercises") will be the page linking all exercises, which will lead to my next page I'm mentioning.

**SQL Query:**

```
"SELECT * FROM Exercises"
```

**Function Signature:**

```
def all_exercises():
```

**Expected results:**

A list of all the names of each exercise, formatted as links to pages showing more detailed information.

Eg:

"Exercises:

Face Pulls

Shrugs

Leg Raise

Russian Twist

Crunch

Plank

Calf Raise

Hip Thrust

Glute Bridge

Deadlift

Romanian Deadlift

Leg Curl

Lunges

Leg Press

Squat

Bent Over Row

Pull Up

Lat Pulldown

Chin Up

Bicep Curl

Tricep Pushdown

Tricep Dips

Incline Bench Press

Push Up

Bench Press

Front Raise

Shoulder Press

Lateral Raise"

**Route:**

@app.route("/exercise/<int:id>") will be a detailed page for each exercise that the user clicks on from the previous page.

**SQL Query:**

```
"SELECT exercise_id, exercise_name, muscle_name FROM Exercises  
INNER JOIN Muscles ON muscle_id = muscle_id WHERE exercise_id = ?"
```

**Function Signature:**

```
def exercise(id):
```

**Expected results:**

Exercise name, and the name of the muscle it targets based on an ID number.

Eg:

“Exercise:

Leg Raise

Muscle:

Abs”

**Route:**

@app.route("/all\_muscles") will be the page linking all muscle groups, which will lead to my next page I'm mentioning.

**SQL Query:**

```
“SELECT * FROM Muscles”
```

**Function Signature:**

```
def all_muscles():
```

**Expected results:**

A list of all the names of each muscle, formatted as links to pages showing more detailed information.

Eg:

“Muscles:

Abs

Forearms

Calves

Trapezius

Lats

Back

Glutes

Hamstrings

Quadriceps

Biceps

Triceps

Chest

Shoulders”

**Route**

@app.route("/muscle/<int:id>") will be a detailed page for each muscle that the user clicks on from the previous page.

## Function Signature

```
def muscle(id):
```

## SQL Query

```
"SELECT muscle_id, muscle_name, exercise_name FROM Muscles  
INNER JOIN Exercises ON exercise_id = exercise_id WHERE muscle_id = ?"
```

## Expected results:

Muscle name, and a list of exercises that target it based on an ID number.

Eg:

"Muscle:

Abs

Exercises:

Leg raise, russian twist, crunch, plank"

## Requirements & Specifications

Requirement	Specifications	Relevant Implication
Must be usable and accessible	<ul style="list-style-type: none"><li>Will be a Python interface</li><li>Code will follow PEP8 Python Style conventions</li><li>Code will be robust and with minimal bugs/errors</li><li>Will include code comments to explain functions</li><li>Interactions will be intuitive and accessible.</li></ul>	Ensures easy readability, maintenance, and consistency while minimising bugs and making my code easier to understand and modify.
Must be informative	<ul style="list-style-type: none"><li>Will be an SQL database</li><li>Database will follow SQL naming conventions</li><li>Database will have correct information</li><li>Database will include exercises and targeted muscles.</li></ul>	Maintains structured and readable data, as well as ensuring accuracy and reliability, and allows efficient retrieval of information.
Must look good	<ul style="list-style-type: none"><li>Will be an HTML and CSS website</li><li>HTML will follow W3Schools recommended design conventions</li><li>Website will have an aesthetically pleasing layout</li><li>Website will be able to be consistently viewable on different resolutions/devices</li><li>Will have a complimentary and aesthetic colour palette.</li></ul>	Ensures aesthetics, accessibility, usability, and responsiveness across various devices, creating a visually appealing and positive experience for my user/s.

# Layout, Fonts, and Colour Palettes + Feedback:

## Website Layout and design prototype:

This is a prototype of what the search page could look like, with an interactable image that allows the user to click on the muscle group they want to search, however if this is outside of my abilities then this could be swapped to a standard search bar. There is a title of the muscle you have searched on the right, and a list of exercises that work that muscle underneath, along with accompanying images. The names of the exercises have been underlined to show that they will be links which will lead to a more detailed description of the selected exercise.

Home Search Exercises



**CHEST**

Bench Press:  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

Chest Press:  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

Chest Flies  
Muscles worked:  
Chest, front deltoids  
(shoulders)

Click muscle to search: Chest



In contrast this image shows the more detailed linked page of the “bench press” exercise, and lists equipment, muscles worked in detail, and technique advice. It also includes further relevant images to demonstrate equipment and form technique.



Image credit: Jun - Getty Images

# BENCH PRESS

## Bench Press Overview

The bench press is a compound exercise that strengthens the chest, shoulders, and triceps. It is widely used in strength training, bodybuilding, and powerlifting.

## Equipment Needed

- Flat Bench – A stable surface to lie on, barbell or dumbbells, weight plates – added resistance for barbells, rack or power cage
- holds the barbell securely, collars/clips – keep weight plates in place, spotter – helps ensure safety when lifting heavy.

## Muscles Worked

- Primary Muscles:
  - Pectoralis Major (Chest)
  - Triceps Brachii (Back of the Arm)
  - Anterior Deltoids (Front Shoulder Muscles)
- Secondary Muscles:
  - Serratus Anterior (Near the Ribs)
  - Latissimus Dorsi (Upper Back)
  - Core (For Stability)

## How to Perform the Bench Press

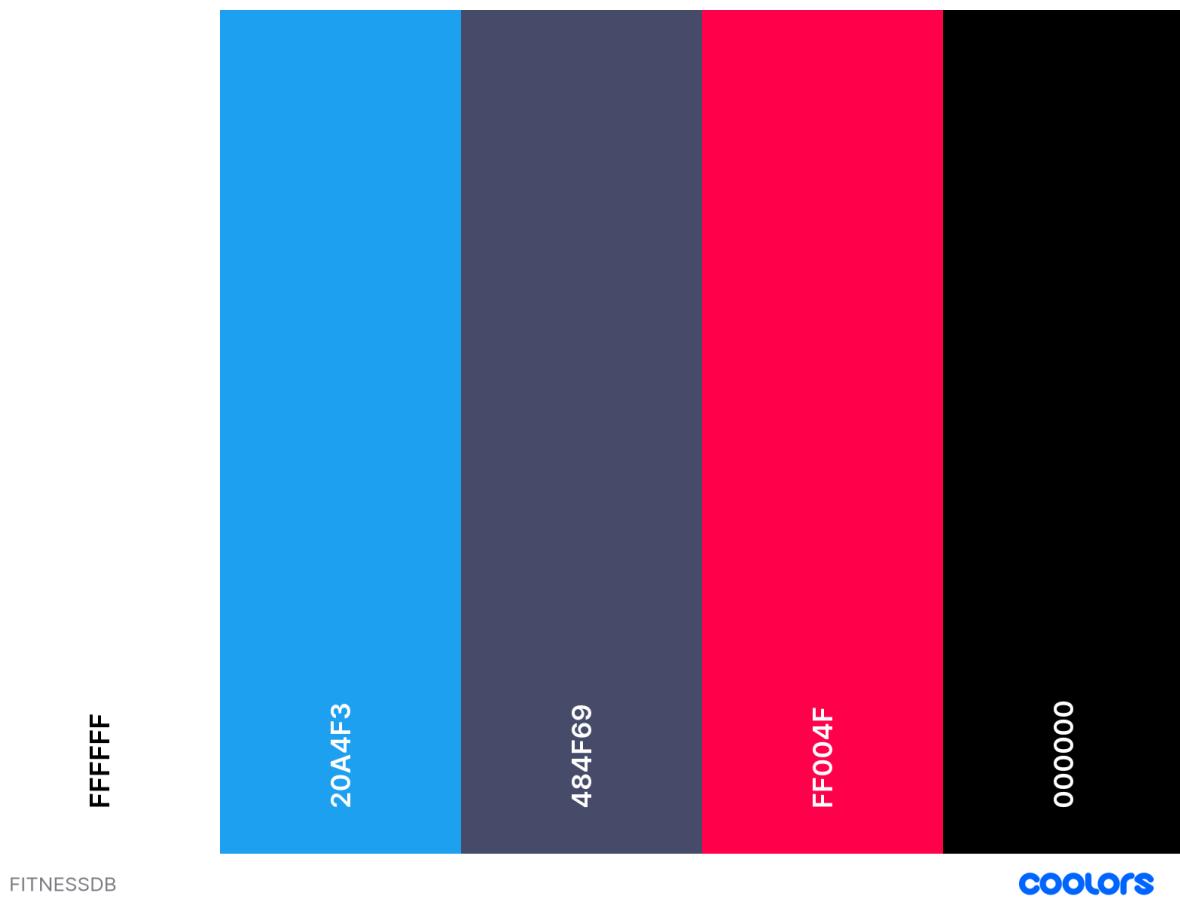
1. Setup: Lie on a flat bench with feet on the ground and back slightly arched.
2. Grip the Bar: Hands slightly wider than shoulder-width.
3. Unrack the Bar: Lift the bar off the rack with arms extended.
4. Lower the Bar: Bring it down to your chest in a controlled motion.
5. Press Up: Push the bar up explosively until arms are fully extended.
6. Lockout & Repeat: Complete the desired reps and re-rack safely.



## Fonts:

As shown in my prototype I think using simple and readable fonts like *LoveLo* for headings and *Calibri* for paragraph text. However this is extremely subject to change, and while it may not be these exact fonts, it will be something similar.

## Colour Palette:



FITNESSDB

COOLORS

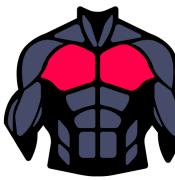
This is a possible colour palette designed using canva and coolors. I want the main colours to be the pink/red, grey, black and white, however I may use the blue as an extra if needed. I could possibly change the white and black to be less absolute (FFFFFF and 000000) in the future as well, although this should be fairly easy to do.

This is a prototype of what the search page could look like, with an interactable image that allows the user to click on the muscle group they want to search, however if this is outside of my abilities then this could be swapped to a standard search bar. There is a title of the muscle you have searched on the right, and a list of exercises that work that muscle underneath, along with accompanying images. The names of the exercises have been underlined to show that they will be links which will lead to a more detailed description of the selected exercise.

### Idea 1

Home Search Exercises

## CHEST



Click muscle to search: [Chest](#)

**Bench Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)



**Chest Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)



**Chest Flies:**  
Muscles worked:  
Chest, front deltoids  
(shoulders)



### Idea 2

Home Muscles Exercises

## CHEST



Click muscle to search: [Chest](#)

**Bench Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)



**Chest Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)



**Chest Flies:**  
Muscles worked:  
Chest, front deltoids  
(shoulders)



### Idea 3

Home Muscles Exercises

## CHEST



Click muscle to search: [Chest](#)

**Bench Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)



**Chest Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)



**Chest Flies:**  
Muscles worked:  
Chest, front deltoids  
(shoulders)



The chest muscles include the pectoralis major, which controls arm movements like pushing, lifting, and rotating, and the pectoralis minor, which stabilizes the shoulder blade. These muscles are key for upper body strength and daily movements. The pectoralis major is responsible for bringing the arm across the body and rotating it inward. The pectoralis minor, lying underneath, pulls the shoulder blade downward and forward, aiding in posture and shoulder stability. Together, they support activities like lifting, pushing, and throwing, and are essential for both athletic performance and everyday tasks.

In contrast this image shows the more detailed linked page of the “bench press” exercise, and lists equipment, muscles worked in detail, and technique advice. It also includes further relevant images to demonstrate equipment and form technique.



Image credit: Jun - Getty Images

# BENCH PRESS

## Bench Press Overview

The bench press is a compound exercise that strengthens the chest, shoulders, and triceps. It is widely used in strength training, bodybuilding, and powerlifting.

## Equipment Needed

- Flat Bench – A stable surface to lie on, barbell or dumbbells, weight plates – added resistance for barbells, rack or power cage
- holds the barbell securely, collars/clips – keep weight plates in place, spotter – helps ensure safety when lifting heavy.

## Muscles Worked

- Primary Muscles:
  - Pectoralis Major (Chest)
  - Triceps Brachii (Back of the Arm)
  - Anterior Deltoids (Front Shoulder Muscles)
- Secondary Muscles:
  - Serratus Anterior (Near the Ribs)
  - Latissimus Dorsi (Upper Back)
  - Core (For Stability)

## How to Perform the Bench Press

1. Setup: Lie on a flat bench with feet on the ground and back slightly arched.
2. Grip the Bar: Hands slightly wider than shoulder-width.
3. Unrack the Bar: Lift the bar off the rack with arms extended.
4. Lower the Bar: Bring it down to your chest in a controlled motion.
5. Press Up: Push the bar up explosively until arms are fully extended.
6. Lockout & Repeat: Complete the desired reps and re-rack safely.



As shown in my prototype I think using simple and readable fonts like *LoveLo* for headings and *Calibri* for paragraph text. However this is extremely subject to change, and while it may not be these exact fonts, it will be something similar.

## Option 1



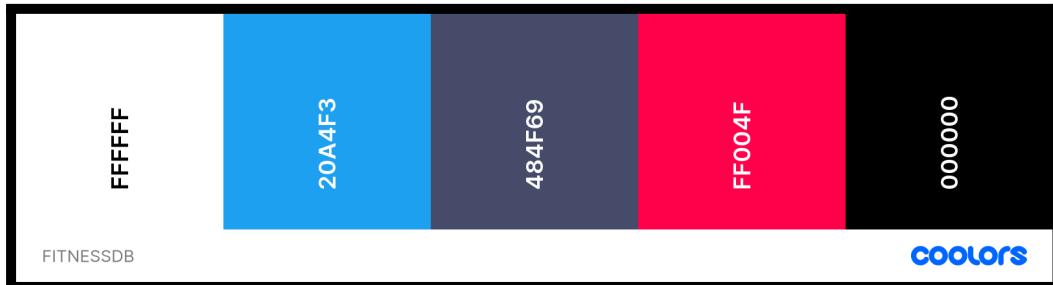
This is a possible colour palette designed using canva and coolors. I want the main colours to be the green on the right, blue, brown and the bright yellow in the middle, however I may use the darker yellow as an extra if needed. I could add a more high contrast colour like white or a darker colour if needed.

## Option 2



This is a possible colour palette designed using canva and coolors. I want the main colours to be the bright green on the right, the blue on the left, the dark green in the middle and the grey colour, however I may use the darker green as an extra if needed. I could add a more high contrast colour like white or a darker colour if needed as well.

### Option 3



This is a possible colour palette designed using canva and coolors. I want the main colours to be the pink/red, grey, black and white, however I may use the blue as an extra if needed. I could possibly change the white and black to be less absolute (FFFFFF and 000000) in the future as well, although this should be fairly easy to do.

## Feedback:

### Who & Role

#### Adi - Friend - Classmate

##### Feedback they gave

"I like the planned interactive features of the website and the functionality (when finished). The colour palette is also good. I don't like the font used, and I think a good idea could be to include video or GIFs to improve understanding. How many exercises are you planning to include?"

##### How does this feedback affect the direction of my design - am I taking it on board and using some of it/all of it or not using and why

I will need to work to ensure that I am able to include the planned features shown in the prototypes in my website. I think that video or gifs could be helpful, but it may take extra work and could be hard to find copyright free high quality video for each exercise. The font is not entirely finalised but I will get further feedback when it is changed in future.

### Who & Role

#### Sean - Classmate

##### Feedback they gave

"I like the interface and think it looks intuitive, but I wonder if the search function will be too complex to make. A good idea would be to remake the layout and prototype different designs."

##### How does this feedback affect the direction of my design - am I taking it on board and using some of it/all of it or not using and why

I'm going to make further variations of the layout with multiple different search designs, and will get further feedback on this in future.

### Who & Role

#### Aine - Classmate

##### Feedback they gave

"The colour palette looks quite extreme which can give the website an electric feel, due to

*the high contrast of complete black and complete white. I think that making another prototype with softer colours could be a good idea. Make sure that the images you use are copyright free and you have the correct permissions to use them."*

How does this feedback affect the direction of my design - am I taking it on board and using some of it/all of it or not using and why

I am going to generate further colour palette options to create multiple options and make choices about my final option. I am also going to research which images I am legally allowed to use.

## Who & Role

**Mrs. Maddaford - Teacher/Design Expert**

Feedback they gave

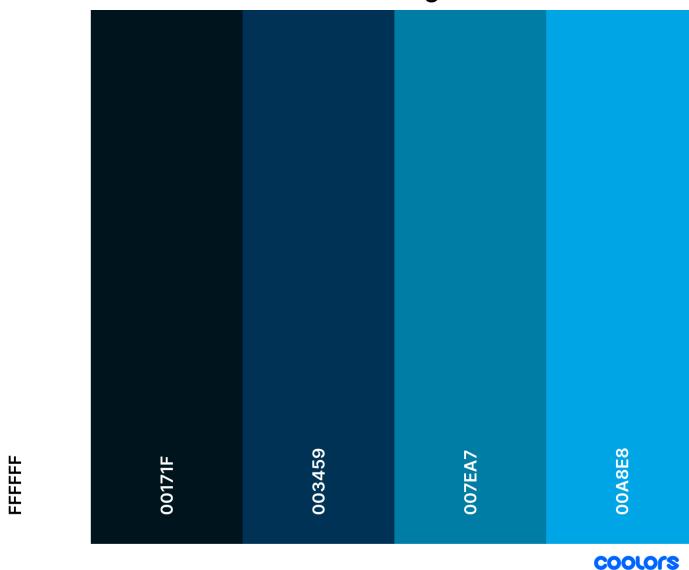
*"I like the idea and layout but I suggest that you research the colour palette and find out what people associate with fitness as well as ensuring that contrast ratios are optimised and that it is accessible to colour blind people."*

How does this feedback affect the direction of my design - am I taking it on board and using some of it/all of it or not using and why

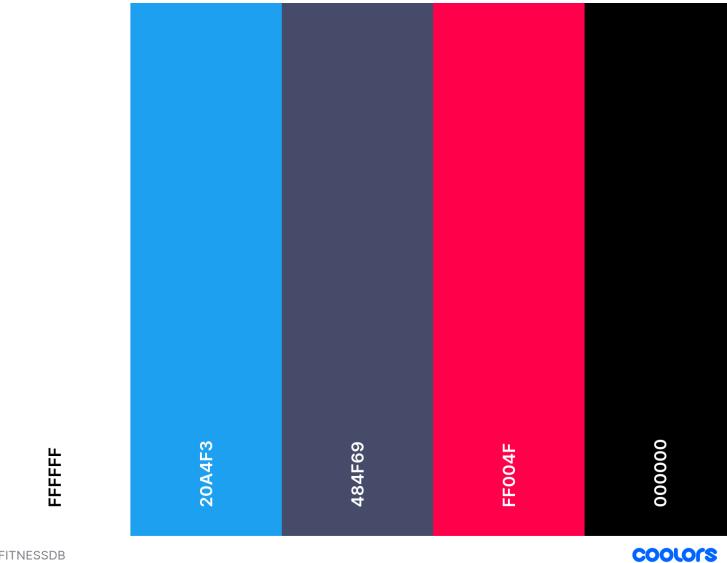
I am researching the current colour palette and its contrast ratios and colour blind testing using Figma, and will do this when I create other colour palettes as well.

## Refinement:

I researched the colours people associate with fitness and energy and found [this website](#), and so based on this research I have generated further colour palette options.



Blue is most often known for representing trust and stability. It also boosts productivity and can encourage people to do more at the gym. Because of this I think that considering a blue based website could be a good idea. Although this does lower the colour contrast of my page, and could make it less accessible for colour blind viewers.



Based on my research, red is a dynamic, powerful color that is very physical. It is energising and can also portray friendliness and strength. It catches attention easily and is very powerful as an accent color. However, research says going all red is not advisable because it can cause aggression and agitation. Mixing red with other colors like white maintains a color balance and can motivate your users while avoiding over-aggressive colours. This colour palette also includes blue which has previously mentioned benefits. It also has high contrast due to varying colours.

This design is what I think best fits my website based on design conventions studied in existing websites like [functionalmovement.com](http://functionalmovement.com), where we see a grid based display of exercises with image previews;

The screenshot shows the homepage of functionalmovement.com. At the top, there is a navigation bar with links for HOME, FMS ACADEMY, THE SYSTEM, ARTICLES, EXERCISES, EDUCATION, PRO APP, and ONLINE COURSES. Below the navigation is a search bar with the placeholder "Find an exercise..." and a "FILTER" button. The main content area is titled "Exercise Library" and features a grid of four exercise cards. Each card contains an image of a person performing an exercise, the name of the exercise, and a brief description. At the bottom of each card is a red "MORE" button. On the far right of the page, there is a small "Privacy + Terms" link.

I aimed to base my website on this one, with the colour palette also being similar. I trialled two different colour palettes and made this initial design, with a grid based design for the text and images.

Home Muscles Exercises



## CHEST

[Bench Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

[Chest Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

[Chest Flies:](#)  
Muscles worked:  
Chest, front deltoids  
(shoulders)

Click muscle to search: [Chest](#)





Home Search Exercises



## CHEST

[Bench Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

[Chest Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

[Chest Flies:](#)  
Muscles worked:  
Chest, front deltoids  
(shoulders)

Click muscle to search: [Chest](#)





I made a second design that's more text based however I do not think that it is the best so I still prefer the next design.

Home Muscles Exercises

## CHEST

### EXERCISES:

The chest muscles include the pectoralis major, which controls arm movements like pushing, lifting, and rotating, and the pectoralis minor, which stabilizes the shoulder blade. These muscles are key for upper body strength and daily movements. The pectoralis major is responsible for bringing the arm across the body and rotating it inward. The pectoralis minor, lying underneath, pulls the shoulder blade downward and forward, aiding in posture and shoulder stability. Together, they support activities like lifting, pushing, and throwing, and are essential for both athletic performance and everyday tasks.

[Chest Flies:](#)  
Muscles worked:  
Chest, front deltoids  
(shoulders)

[Chest Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

[Bench Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)





Home Muscles Exercises

## CHEST

### EXERCISES:

The chest muscles include the pectoralis major, which controls arm movements like pushing, lifting, and rotating, and the pectoralis minor, which stabilizes the shoulder blade. These muscles are key for upper body strength and daily movements. The pectoralis major is responsible for bringing the arm across the body and rotating it inward. The pectoralis minor, lying underneath, pulls the shoulder blade downward and forward, aiding in posture and shoulder stability. Together, they support activities like lifting, pushing, and throwing, and are essential for both athletic performance and everyday tasks.

[Chest Flies:](#)  
Muscles worked:  
Chest, front deltoids  
(shoulders)

[Chest Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

[Bench Press:](#)  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)





I also created a third option for the layout, presenting a separate option to my original. I think that this design is the best because it is most similar to existing alternatives like [functionalmovement.com](http://functionalmovement.com), with its grid based design.

Home Muscles Exercises

# CHEST

**EXERCISES:**

**Bench Press:**  
Muscles worked:  
Chest, triceps, front deltoids (shoulders)

**Chest Press:**  
Muscles worked:  
Chest, triceps, front deltoids (shoulders)

**Chest Flies:**  
Muscles worked:  
Chest, front deltoids (shoulders)

Click muscle to search:

The chest muscles include the pectoralis major, which controls arm movements like pushing, lifting, and rotating, and the pectoralis minor, which stabilizes the shoulder blade. These muscles are key for upper body strength and daily movements.

Home Search Exercises

# CHEST

**EXERCISES:**

**Bench Press:**  
Muscles worked:  
Chest, triceps, front deltoids (shoulders)

**Chest Press:**  
Muscles worked:  
Chest, triceps, front deltoids (shoulders)

**Chest Flies:**  
Muscles worked:  
Chest, front deltoids (shoulders)

Click muscle to search:

The chest muscles include the pectoralis major, which controls arm movements like pushing, lifting, and rotating, and the pectoralis minor, which stabilizes the shoulder blade. These muscles are key for upper body strength and daily movements.

## Feedback:

### Who & Role

#### Adi - Friend

##### Feedback they gave

*"I don't like the font used, and prefer the varied colour palette rather than the blue one. I also prefer the first layout rather than the second."*

*How does this feedback affect the direction of my design - am I taking it on board and using some of it/all of it or not using and why*

I am going to generate some different font choices so that I can decide on a final choice/s. I think that the second design is more suitable for my end vision of the design, so I will not choose to follow this feedback.

### Who & Role

#### Xavier - Classmate

##### Feedback they gave

*"I prefer the varied colour palette and the second layout design. The blue palette is lacking contrast so I don't think it is the best choice."*

*How does this feedback affect the direction of my design - am I taking it on board and using some of it/all of it or not using and why*

I think that I am going to use the second design, due to the majority of feedback, however I may make some changes or make a third design. I am planning on using the more varied colour palette.

### Who & Role

#### Aine - Classmate

##### Feedback they gave

*"The blue palette is very monochromatic and would lower contrast if used. I prefer the*

*second layout in design, but the first would be more effective if you decide to add extra exercises."*

*How does this feedback affect the direction of my design - am I taking it on board and using some of it/all of it or not using and why*

I plan on not using the blue palette as its contrast is too low, and I am planning on using the second layout, although I may make some changes.

## Appropriateness of Design:

I think that I have fulfilled this in my design as I have included a top navigation bar with my three main pages (although this may be changed if needed), and even if I change my design I plan on not including any more than 5.

[This website](#) says that colors should be determined based on your company's brand standards. However they recommended that if your company does not have brand standards to use 2-3 colors and 1-3 neutrals (white, black, gray). They say that if you need more color then use tints and shades of the colors already chosen. This is an important website convention because it will keep my website cohesive and connected.

I think that I have fulfilled this as I have a white black and grey colour as well as one red and one blue colour. I plan on using different tints of the red and blue if needed during development to help create contrast and ensure variation.

Another website convention they recommend to follow is link styling. Cross linking between pages and linking to other sources is a common practice but they say in order to differentiate link text from page text, I will need to use a different color for each. They note that page text is usually some shade of black, whereas link text is typically a brighter color and, sometimes, bold or underlined. They typically recommend using the company's primary or secondary color for links.

I think I have fulfilled these conventions in my design as I considered multiple colour palettes and made decisions about contrast and which colours should be used in my website. I decided on using the red colour for my links, as well as including underlining to help them stand out. Since the colour palette included a black colour I text is also black, which creates contrast with the background and links. I think that this provides suitable contrast between the text and links to ensure that my website use will be as clear as possible.

One of my relevant implications was an aesthetic and minimalist design, and I aimed to fulfill this by gathering feedback from potential end users on my layout, colour palette, and fonts. By making sure to account for my users feedback, I can ensure that my final product will have an aesthetic and minimalist design that best fulfills my end users needs.

I think that I was able to fulfill this as I received feedback multiple times from potential users and made changes accordingly, leading to a preferred design and colour palette.

Such as Adi's feedback where he said "*I don't like the font used, and prefer the varied colour palette rather than the blue one. I also prefer the first layout rather than the second.*" By choosing to use the less text based layout based on his feedback, as well as making choices about the colour palette to make sure that my potential end users (such as Adi), will find using my end product as aesthetic and minimal as possible. Similarly Xavier said "*I prefer the varied colour palette and the second layout design. The blue palette is lacking contrast so I don't think it is the best choice.*" My choice to use the more varied colour palette is reinforced again by Xavier's feedback, and he also prefers less text based layouts, like the second design. Finally, Aine said that "*The blue palette is very monochromatic and would lower contrast if used. I prefer the second layout in design, but the first would be more effective if you decide to add extra exercises.*", this reinforces the varied colour palette even further, further ensuring that I have created the best possible design for my end users.

Because my end users include such a broad range of people, with it being anyone looking to begin going to the gym, inexperienced gym goers or anyone that wants to learn more about exercises and their physiology. I hope to create a website that anyone can access and use as a resource, giving me potentially unlimited end users, however I think I will target my website towards people aged 16-40 years old, as based on my research, the majority of gym-goers are in this range. They require my website to have minimal errors, be accessible and intuitive, and to have aesthetically pleasing design, while also clearly and concisely communicating my information.

I think that I have fulfilled this as I deliberately used less wordy and less text based designs to improve accessibility, and make information more clear and concise to ensure it was as understandable as possible for both new users and more experienced users.

The purpose of my final outcome was to help alleviate some of these risks involved with weight training, as well as spread awareness and information about exercises and correct form, hopefully helping a wider population enter and thrive in the gym, and helping them avoid some of the difficulties that I faced when I first started. Based on my research, injury and incorrect/incomplete knowledge are some of the main barriers to people both enjoying the gym and making progress.

I think that I have fulfilled this purpose in my design, such as how I have included information like images and form advice, providing both basic and advanced info to ensure that my website can spread information as clearly as possible and help my users best reduce the risks involved with weight training without experience and also included a taskbar to help users navigate the website as easily as possible, making information more accessible and helping new users find the information they are looking for.

## Relevant Implications

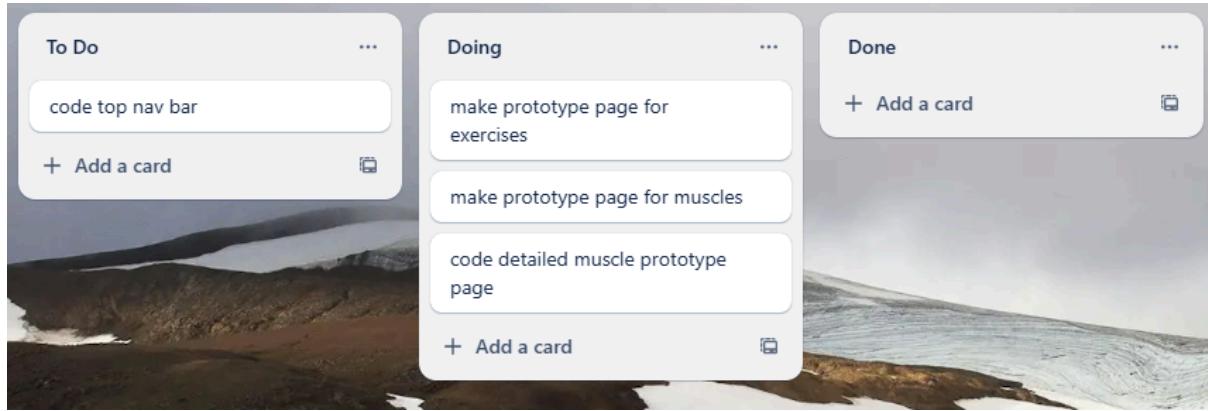
Implication	Explain the implication
Social	Because of my intended audience being targeted towards as many people as possible, I will need to ensure that I use accessible language, making information as easy and intuitive to access as possible.
Cultural	I want my website to be inclusive to as many cultures as possible, however I also need to think within the scope of the project and understand that including each language is a tall task. I think that by making my website in english, users who speak other languages can use the built in google translate feature to translate the website into their language. I will also try to use icons if needed to help these users understand the functions even without translation.
Legal	I will be placing a copyright under my name and the year at the bottom of the website, saying "©Rory Collins 2025". I will also be placing warnings on dangerous exercises, as well as just in general to ensure that my website's information doesn't endanger anyone and cause injury which could lead to legal prosecution.
Ethical	By including warnings on dangerous exercises, as well as including a generalised disclaimer saying something along the lines of "Warning: exercise may lead to injury,

	consult with a doctor or qualified personal trainer if you are unsure about whether or not it is safe for you”, I can hopefully help minimise or prevent injury, improving my users experience significantly.
Intellectual Property	The website will give credits to any images, sound, or text etc; that is not my own. I will abide by others legal copyright on their ideas and media, and if needed will contact the original owners and ask for permission.
Privacy	My users will not have to submit any personal data to my website, and the only thing that will be stored is what they have searched temporarily while it is searched into my database. This should mean that there are no security risks involved with my outcome, and that users don't have to worry about data leaks or other risks.
Accessibility	By including alt text on my images used in my website, I help it become more accessible to visually impaired users, widening my target user base. I can also use icons instead of solely colourful buttons to help improve accessibility for users who speak other languages. By using common design conventions like a task bar I lower the learning curve and make it more intuitive for new users.
Usability	I ensure that my website is usable by making sure that there are as little bugs as possible, as well as intuitive interaction with the website eg: the top nav bar.
Functionality	I will also need to ensure that the website is completely functional and that the user is able to access and view all the pages correctly, no matter which devices they are on, or other variations.
Aesthetics	By gathering feedback from potential end users on my layout, colour palette, and fonts, I ensure that my final product will have an aesthetic and minimalist design that fulfills my end users needs.
Sustainability and Future Proofing	I will be able to update the code and database in future if needed, which should effectively let me future proof (at least while I'm actively working on its upkeep). My website should stay relevant, although AI could end up doing its job for it in future, which is an inevitable and unavoidable event.
End-user considerations	By understanding who my end users are, as well as understanding what they want through feedback and testing, I can ensure that my outcome correctly targets my end users, and provides a satisfactory and positive experience.
Health and Safety	By providing warnings for dangerous exercises and in general about my information, as well as not including flashing lights or screens. By choosing inoffensive colours I can also try to reduce eye strain for my users.

## **Sprint 1 - MVP (minimum viable product)**

### **Basic Framework / Functionality:**

#### **Plan**



## Development:

### ExDB

[Home](#) [Muscles](#) [Exercises](#)

#### Links to other pages:

[View All Muscles](#)

[View All Exercises](#)

Rory Collins ©2023

This is my planned design layout, or at least something similar to this.

Home Muscles Exercises

# SEARCH:



Click muscle to search:

Muscles:	Exercises:
<a href="#">Abs</a>	<a href="#">Face Pulls</a>
<a href="#">Forearms</a>	<a href="#">Shrugs</a>
<a href="#">Calves</a>	<a href="#">Leg Raise</a>
<a href="#">Trapezius</a>	<a href="#">Russian Twist</a>
<a href="#">Lats</a>	<a href="#">Crunch</a>
<a href="#">Back</a>	<a href="#">Plank</a>
<a href="#">Glutes</a>	<a href="#">Calf Raise</a>
<a href="#">Hamstrings</a>	<a href="#">Hip Thrust</a>
<a href="#">Quadriceps</a>	<a href="#">Glute Bridge</a>
<a href="#">Biceps</a>	<a href="#">Deadlift</a>
<a href="#">Triceps</a>	<a href="#">Romanian Deadlift</a>
<a href="#">Chest</a>	<a href="#">Leg Curl</a>
<a href="#">Shoulders</a>	<a href="#">Lunges</a>

Home Muscles Exercises

## CHEST EXERCISES:



Click muscle to search:

**Bench Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

**Chest Press:**  
Muscles worked:  
Chest, triceps, front deltoids  
(shoulders)

**Chest Flies:**  
Muscles worked:  
Chest, front deltoids  
(shoulders)

The chest muscles include the pectoralis major, which controls arm movements like pushing, lifting, and rotating, and the pectoralis minor, which stabilizes the shoulder blade. These muscles are key for upper body strength and daily movements.

Home Muscles Exercises

## BENCH PRESS



Image credit: Jun - Getty Images

**Bench Press Overview**  
The bench press is a compound exercise that strengthens the chest, shoulders, and triceps. It is often used in strength training, bodybuilding, and powerlifting.

**Equipment Needed:**

- Flat Bench - A flat surface to lie on, barbell or dumbbells, weight plates - added resistance by adding weight to the plates - hold the barbell securely, callus tips - keep weight plates in place - grip powder - regular exercise will increase grip strength

**Muscles Worked:**

- Primary Muscles:**
  - Pectoralis Major (Chest)
  - Anterior Deltoid (Front Shoulder Muscle)
  - Secondary Muscles:
  - Anterior Tricep (Upper Arm)
  - Lateral Tricep (Outer Tricep)
  - Posterior Tricep (Inner Tricep)

**How to Perform the Bench Press**

- Setup: Lie on a flat bench with feet on the floor and back slightly arched.
- Grasp the bar with hands wider than the shoulder width.
- Inhale and lift the bar up off the rack with arms extended.
- Lower the bar: Bring it down to your chest in a controlled motion.
- Push the bar up again explosively until arms are fully extended.
- Lockout: Hold the weight at the end of the movement.

Diagram illustrating the bench press movement cycle.

FFFFFF
20A4F3
484F9
FF004F
000000

This is a prototype version of my home page, which may be changed in future, but for now just leads to the other two planned pages. I have also made a prototype of my top navigation bar, however it doesn't include the boxes or housing yet, making it a functional but unaesthetic feature.:)

```

@app.route('/')
def home():
    return render_template('home.html', title='Home')

{% extends 'layout.html' %}
{% block content %}
<div class="topnav">
    <a class="active" href="/">Home</a>
    <a href="/all_muscles">Muscles</a>
    <a href="/all_exercises">Exercises</a>
</div>
<h1>Links to other pages:</h1>
<p><a href="all_muscles">View All Muscles</a></p>
<p><a href="all_exercises">View All Exercises</a></p>
{% endblock %}

```

The screenshot shows a simple web application interface. At the top, there is a header with the text "ExDB". Below the header is a navigation bar containing three items: "Home", "Muscles", and "Exercises". The main content area contains the following text:

**Links to other pages:**

[View All Muscles](#)  
[View All Exercises](#)

At the bottom of the page, there is a small copyright notice: "Rory Collins ©2025".

This is a prototype of the muscles page, where everything is printed as links which will lead to more detailed pages when they are finished.

```

@app.route('/all_muscles')
def all_muscles():
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM Muscles")
    muscles = cur.fetchall()
    conn.close()
    return render_template('all_muscles.html', title='Muscles', muscles=muscles)

```

```

{% extends 'layout.html' %}

{% block content %}

<div class="topnav">
    <a href="/">Home</a>
    <a class="active" href="/all_muscles">Muscles</a>
    <a href="/all_exercises">Exercises</a>
</div>

<h1>Muscles:</h1>

{% for muscle in muscles %}

<p><a href="muscle/{{muscle[0]}}">{{muscle[1]}}</a></p>

{% endfor %}

{% endblock %}

```

## ExDB

[Home](#) [Muscles](#) [Exercises](#)

### Muscles:

- Abs
- Forearms
- Calves
- Trapezius
- Lats
- Back
- Glutes
- Hamstrings
- Quadriceps
- Biceps
- Triceps
- Chest
- Shoulders

Rory Collins ©2025

This is a prototype of the exercise page, where everything is printed as links which will lead to more detailed pages when they are finished.

```

@app.route('/all_exercises')
def all_exercises():
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM Exercises")
    exercises = cur.fetchall()
    conn.close()
    return render_template('all_exercises.html', title='Exercises', exercises=exercises)

```

```

{% extends 'layout.html' %}

{% block content %}

<div class="topnav">
    <a href="/">Home</a>
    <a href="/all_muscles">Muscles</a>
    <a class="active" href="/all_exercises">Exercises</a>
</div>

<h1>Exercises:</h1>

{% for exercise in exercises %}

<p><a href="exercise/{{exercise[0]}}">{{exercise[1]}}</a></p>

{% endfor %}

{% endblock %}

```

## ExDB

Home Muscles Exercises

### Exercises:

[Face Pulls](#)  
[Shrugs](#)  
[Leg Raise](#)  
[Russian Twist](#)  
[Crunch](#)  
[Plank](#)  
[Calf Raise](#)  
[Hip Thrust](#)  
[Glute Bridge](#)  
[Deadlift](#)  
[Romanian Deadlift](#)  
[Leg Curl](#)  
[Lunges](#)  
[Leg Press](#)  
[Squat](#)  
[Pull Up](#)  
[Bent Over Row](#)  
[Chin Up](#)  
[Lat Pulldown](#)  
[Bicep Curl](#)  
[Tricep Pushdown](#)  
[Tricep Dips](#)  
[Incline Bench Press](#)

## What I am testing

# ExDB

[Home](#) [Muscles](#) [Exercises](#)

## Links to other pages:

[View All Muscles](#)

[View All Exercises](#)

Rory Collins ©2025

# ExDB

[Home](#) [Muscles](#) [Exercises](#)

## Links to other pages:

[View All Muscles](#)

[View All Exercises](#)

Rory Collins ©2025

# ExDB

[Home](#) [Muscles](#) [Exercises](#)

## Muscles:

Abs

Forearms

Calves

Trapezius

Lats

Back

Glutes

Hamstrings

Quadriceps

Biceps

Triceps

Chest

Shoulders

Rory Collins ©2025

# ExDB

Home Muscles Exercises

## Exercises:

[Face Pulls](#)  
[Shoulder Press](#)  
[Leg Raise](#)  
[Russian Twist](#)  
[Crunch](#)  
[Plank](#)  
[Calf Raise](#)  
[Hip Thrust](#)  
[Glute Bridge](#)  
[Deadlift](#)  
[Romanian Deadlift](#)  
[Leg Curl](#)  
[Lunges](#)  
[Leg Press](#)  
[Squat](#)  
[Pull Up](#)  
[Bent Over Row](#)  
[Chin Up](#)  
[Lat Pulldown](#)  
[Bicep Curl](#)  
[Tricep Pushdown](#)  
[Tricep Dips](#)  
[Incline Bench Press](#)

```
@app.route('/')
def home():
    return render_template('home.html', title='Home')
```

```
{% extends 'layout.html' %}
{% block content %}
<div class="topnav">
    <a class="active" href="/">Home</a>
    <a href="/all_muscles">Muscles</a>
    <a href="/all_exercises">Exercises</a>
</div>
<h1>Links to other pages:</h1>
<p><a href="all_muscles">View All Muscles</a></p>
<p><a href="all_exercises">View All Exercises</a></p>
{% endblock %}
```

```
@app.route('/all_muscles')
def all_muscles():
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM Muscles")
    muscles = cur.fetchall()
    conn.close()
    return render_template('all_muscles.html', title='Muscles', muscles=muscles)
```

```
{% extends 'layout.html' %}  
{% block content %}  
<div class="topnav">  
    <a href="/">Home</a>  
    <a class="active" href="/all_muscles">Muscles</a>  
    <a href="/all_exercises">Exercises</a>  
</div>  
<h1>Muscles:</h1>  
{% for muscle in muscles %}  
<p><a href="muscle/{{muscle[0]}}">{{muscle[1]}}</a></p>  
{% endfor %}  
{% endblock %}
```

```
@app.route('/all_exercises')  
def all_exercises():  
    conn = sqlite3.connect('fitness.db')  
    cur = conn.cursor()  
    cur.execute("SELECT * FROM Exercises")  
    exercises = cur.fetchall()  
    conn.close()  
    return render_template('all_exercises.html', title='Exercises', exercises=exercises)
```

```

{% extends 'layout.html' %}

{% block content %}

<div class="topnav">
    <a href="/">Home</a>
    <a href="/all_muscles">Muscles</a>
    <a class="active" href="/all_exercises">Exercises</a>
</div>

<h1>Exercises:</h1>

{% for exercise in exercises %}

<p><a href="exercise/{{exercise[0]}}">{{exercise[1]}}</a></p>

{% endfor %}

{% endblock %}

```

## Testing

Test ID	Feature	Interaction	Expected result	Actual result
1	Task bar	Click muscle/exercise	Move to exercise/muscle page	Passed
2	Exercise links	Click link	Move to detailed exercise page	Passed
3	Home page links	Click view all muscles/exercises	Move to list of muscles/exercises	Passed
4	Character set was initialised correctly	Test if language is correct	Displays english and full text	Passed
5	Viewport size was initialised correctly	Test if viewport size is correct	Displays correctly	Passed

Overall, I think that the website has passed all the tests, however there was a very limited amount of features to trial.

## Feedback:

Elijah:

Elijah regularly goes to the gym and said that he thinks a website like mine could be helpful for finding exercises once complete. His feedback on my website was “*Adding a way to search for exercises by muscle as well as adding a search window to the nav bar would*

*make finding what you are looking for easier. Make sure to add images and explain the form for exercises.”*

Adi:

*Adi said “I don’t like the layout of the lists. It should be horizontally spaced instead of vertically. I think that adding a search function is important and would be very helpful to new users. I also think that you should include images showing the form of the exercise.”*

Sean:

*Sean said “I think that a search should be added to make the website more easy to navigate. The current orientation of the list should be changed to make the website more compact. I also think adding images would be helpful.”*

Logan:

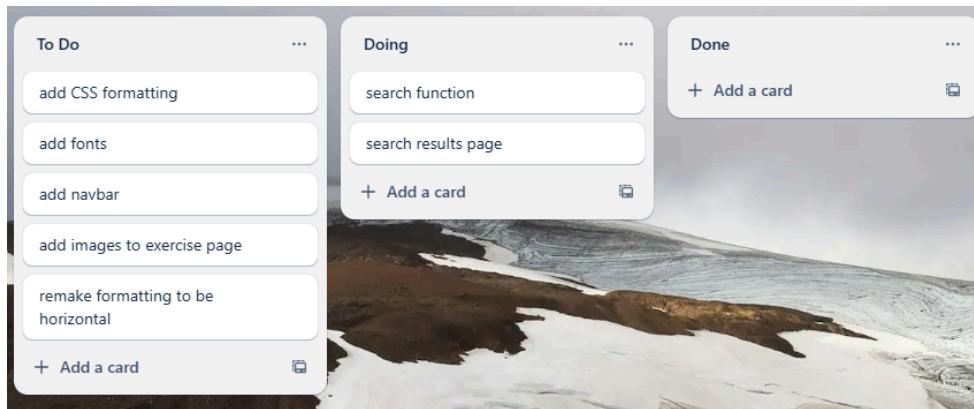
*Logan said “I like the layout but I think that adding images could be helpful. I think that making the taskbar look better will make its function more obvious. I also think that including information about the form is important.”*

## Next Steps

I am going to change the layout of my website to be more horizontal rather than vertical, as well as adding images where necessary. I am also going to add a way to search, and redo the taskbar as well as adding basic CSS formatting to make the website look better.

## Sprint 2 - MMP (minimum marketable product) Adding Content / Aesthetics

### I PLAN TO



### WHAT I HAVE DEVELOPED DURING THIS SPRINT (the trialling)



This is a screen recording of what my website looks like after adding some basic CSS to the taskbar, and adding a search input (Doesn't lead anywhere as the html page has not been made yet).



After adding more advanced css. Notice that there are still no links on the muscle page (added later)

This is my home page, which may be changed in future, but for now just leads to the other two planned pages. The home page is relatively empty currently, but I plan on making it a preview showcase of exercises featured in my website.

```
# Home page to display previews
@app.route('/')
def home():
    return render_template('home.html', title='Home')
```



This is the muscles page, where everything is printed as links which lead to a list of links to all the exercises that target that muscle.

```
@app.route('/all_muscles')
def all_muscles():
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM Muscles")
    muscles = cur.fetchall()
    conn.close()
    return render_template('all_muscles.html', title='Muscles', muscles=muscles)

{% extends 'layout.html' %}
{% block content %}


Home
Muscles
All Exercises


```



This is the exercises page, where everything is printed as images titled with links which lead to more detailed pages about the exercise. There is also the more detailed specific exercise page which can be accessed by clicking on one of the links.

```
#app.route('/all_exercises')
def all_exercises():
    conn = sqlite3.connect('fitness.db')
    conn.row_factory = sqlite3.Row
    cur = conn.cursor()
    cur.execute("""
        SELECT e.exercise_id, e.exercise_name, m.muscle_name, i.image_url
        FROM Exercises e
        INNER JOIN Muscles m ON e.muscle_id = m.muscle_id
        LEFT JOIN Images i ON i.image_id = e.exercise_id
    """)
    exercises = cur.fetchall()
    conn.close()
    return render_template('all_exercises.html', title='All Exercises', exercises=exercises)

1  {% extends 'layout.html' %}
2  {% block content %}
3  <div class="topnav">
4      <a href="/>Home</a>
5      <a href="/all_muscles">Muscles</a>
6      <a class="active" href="/all_exercises">Exercises</a>
7      <form action="{{ url_for('search') }}" method="get" class="topnav">
8          <input type="text" name="query" placeholder="Search" required />
9      </form>
10 </div>
11 <h1>Exercises</h1>
12 <div class="exercise-grid">
13     {% for exercise in exercises %}
14         <div class="container">
15             
17                 <p><a href="{{url_for('exercise', id=exercise[0])}" style="font-size: 30px;">{{exercise[1]}}</a></p>
18             </div>
19         </div>
20     {% endfor %}
21 </div>
22 {% endblock %}

@app.route("/exercise/<int:id>")
def exercise(id):
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("""
        SELECT e.exercise_id, e.exercise_name, m.muscle_name, i.image_url
        FROM Exercises e
        INNER JOIN Muscles m ON e.muscle_id = m.muscle_id
        LEFT JOIN Images i ON i.image_id = e.exercise_id
        WHERE e.exercise_id = ?
    """, (id,))
    exercise = cur.fetchone()
    title = 'Exercises - ' + str(exercise[1])
    conn.close()
    return render_template('exercise.html', title=title, exercise=exercise)

5  {% extends 'layout.html' %}
6  {% block content %}
7  <div class="topnav">
8      <a href="/>Home</a>
9      <a href="/all_muscles">Muscles</a>
10     <a class="active" href="/all_exercises">Exercises</a>
11     <form action="{{ url_for('search') }}" method="get" class="topnav">
12         <input type="text" name="query" placeholder="Search" required />
13     </form>
14 </div>
15 <h2>{{ exercise[1] }}</h2>
16 <h3>Muscle:</h3>
17 <ul>
18     <li>{{ exercise[2] }}</li>
19 </ul>
20 
21 <p>No image available for this exercise.</p>
22 {% endblock %}
```

This is the search function, accessed in the top right of the navigation bar. It allows you to search a string and will output similar muscle and exercise results.

```
# This is a function to search the database for information similar to entered string
@app.route('/search')
def search():
    query = request.args.get('query', '').strip()
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("SELECT muscle_id, muscle_name FROM Muscles WHERE muscle_name LIKE ?", ('%' + query + '%'))
    muscles = cur.fetchall()
    cur.execute("SELECT exercise_id, exercise_name FROM Exercises WHERE exercise_name LIKE ?", ('%' + query + '%'))
    exercises = cur.fetchall()
    conn.close()
    return render_template('search_results.html', title='Search Results', query=query, muscles=muscles, exercises=exercises)
```

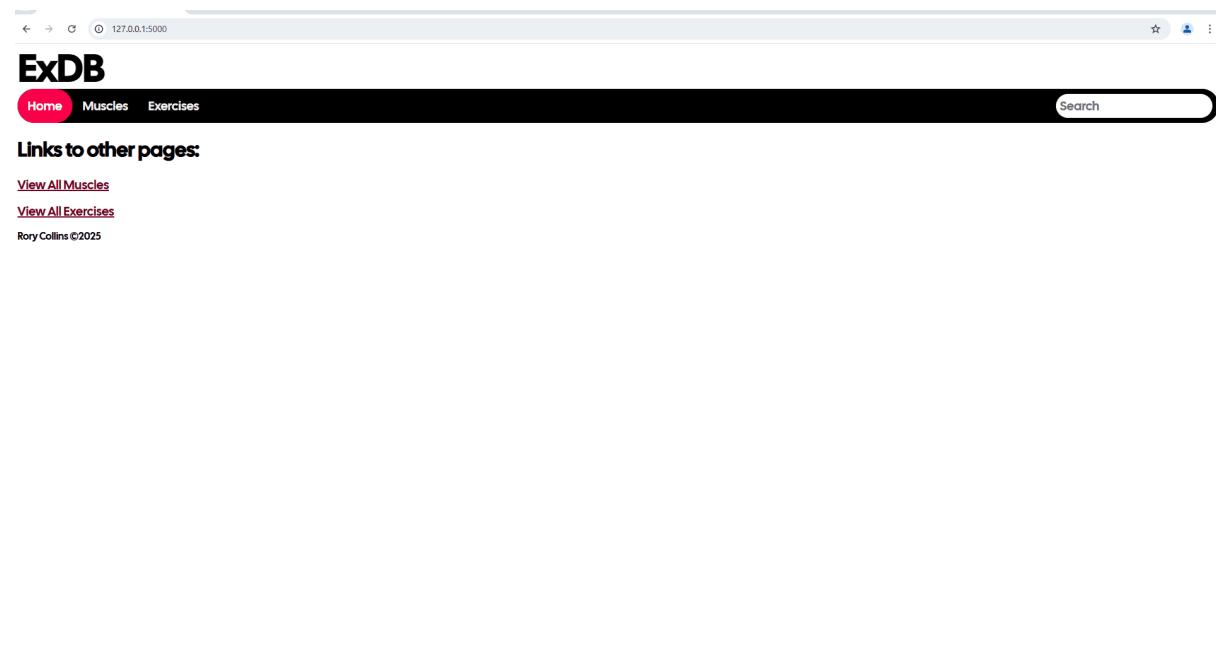
```

<% extends 'layout.html' %>
<% block content %>


Home
Muscles
Exercises
<form action="{{ url_for('search') }}" method="get" class="topnav">
    <input type="text" name="query" placeholder="Search" required />
</form>
</div>
<h1>Search Results for "{{ query }}</h1>
<h2>Muscles</h2>
<% if muscles %>
<ul>
    {% for muscle in muscles %}
        <li><a href="{{ url_for('muscle_exercises', id=muscle[0]) }}>{{ muscle[1] }}</a>
    {% endfor %}
</ul>
<% else %>
    <p>No muscles found.</p>
<% endif %>
<h2>Exercises</h2>
<% if exercises %>
<ul>
    {% for exercise in exercises %}
        <li><a href="{{ url_for('exercise', id=exercise[0]) }}>{{ exercise[1] }}</a></li>
    {% endfor %}
</ul>
<% else %>
    <p>No exercises found.</p>
<% endif %>
<% endifblock %>


```

## END OF SPRINT - WHAT I AM TESTING



## Testing

Test ID	Feature	Interaction	Expected result	Actual result
1	Task bar links	Click muscles/exercises button	Move to muscles/exercises	Passed

2	Home page links	Click view all muscles/exercises link	Move to muscles/exercises link	Passed
3	Search bar	Type e	Show all muscles/exercises containing e	Passed
4	Search bar	Type nothing	Do not change page	Passed
5	Search bar	Enter a letter not contained in anything	Show no results	Passed
6	Exercise title links	Click title text to go to more detailed view of exercise	Changes page to more detailed exercise information	Passed
7	Exercise images + title display	View exercise page	Exercise images displayed in grid	Passed
8	Muscles page	View muscle page	Muscle names displayed in list	Passed
9	Character set was initialised correctly	Test if language is correct	Displays english and full text	Passed
10	Viewport size was initialised correctly	Test if viewport size is correct	Displays correctly	Passed

#### **CLIENT / STAKEHOLDER / EXPERT TESTING & FEEDBACK**

Adi -



I like the exercise page, however adding more content to the home page to reduce whitespace would be helpful. The muscles page could also be changed to be set up more similarly to the exercise page to reduce emptiness.

*Xavier -*

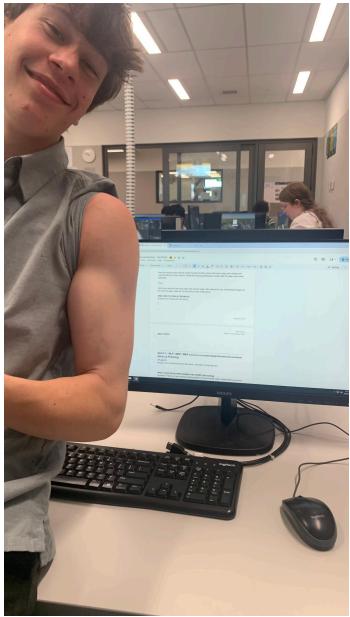
I like the exercise page and the search function but the muscle and home page seem empty and could benefit from more content. I think that reducing whitespace would make the page seem more appealing.

*Sean -*

Add more content to the home page and muscles page. Also reduce the size of thumbnail images on the exercise page. Make the UI and menus more professional.

## **END USER TESTING & FEEDBACK**

*Elijah -*



Elijah goes to the gym frequently and calls himself a “gymrat”. He says that the website is good but it needs more exercises and more information about form.

#### **NEXT STEPS**

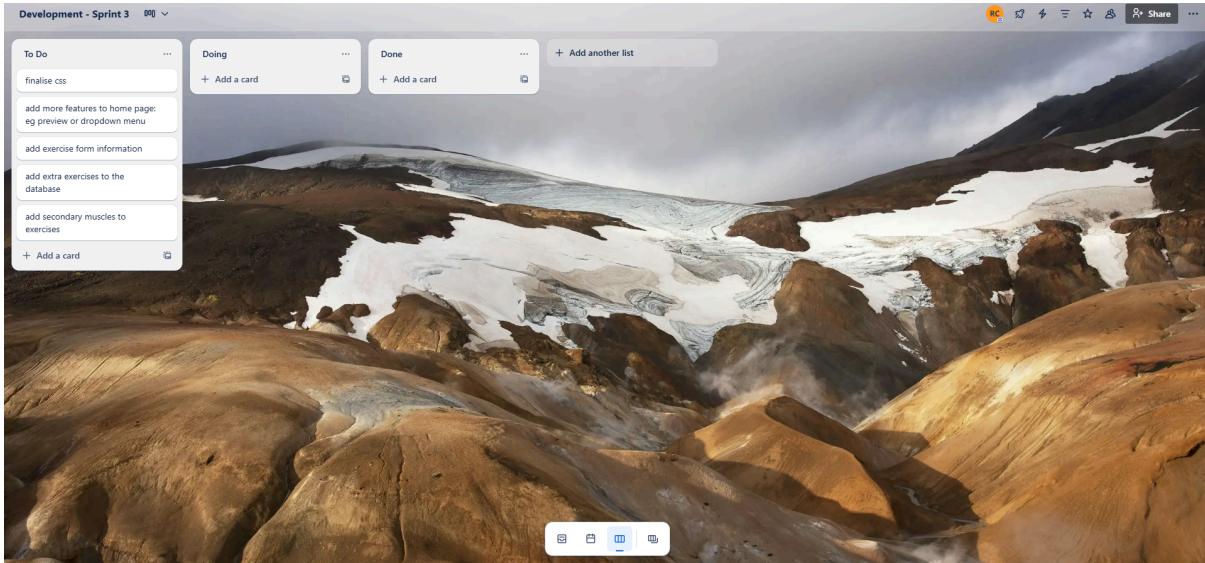
My next step is to add more to the home page, as well as to add more information about the exercises. I will also do some finalised CSS design and make sure it looks as professional as possible.

---

#### **Sprint 3 - MLP / MDP / MAP (minimum loveable/delightful/awesome product)**

#### **Extras & Polishing**

#### **I PLAN TO**



## WHAT I HAVE DEVELOPED DURING THIS SPRINT (the trialling)

Before	After
<p>This is my original code for the home page, which only showed links to the other pages. Notice the layout of the text and the amount of empty space.</p> <pre># Home page to display previews @app.route('/') def home():     return render_template('home.html', title='Home')  &lt;h1&gt;Links to other pages&lt;/h1&gt; &lt;p&gt;&lt;a href="/all_muscles"&gt;View All Muscles&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&lt;a href="/all_exercises"&gt;View All Exercises&lt;/a&gt;&lt;/p&gt;</pre> 	<pre># Home page to display previews @app.route('/') def home():     conn = sqlite3.connect('fitness.db')     conn.row_factory = sqlite3.Row     cur = conn.cursor()     cur.execute("SELECT image_url FROM Images")     image_rows = cur.fetchall()     conn.close()     if not image_rows:         return "No images available."     random_images = random.sample(image_rows, 10)     image_urls = [image['image_url'] for image in random_images]     return render_template('home.html', title="Home", images=image_urls)  {% extends 'layout.html' %}  {% block content %} <div class="topnav">     &lt;a class="active" href="/"&gt;Home&lt;/a&gt;     &lt;a href="/all_muscles"&gt;Muscles&lt;/a&gt;     &lt;a href="/all_exercises"&gt;Exercises&lt;/a&gt;     &lt;form action="{{ url_for('search') }}" method="get" class="topnav"&gt;         &lt;input type="text" name="query" placeholder="Search" required /&gt;     &lt;/form&gt; </div>  &lt;h1&gt;Welcome to the Exercise Database!&lt;/h1&gt; &lt;h2&gt;Image Gallery Preview&lt;/h2&gt; &lt;div class="gallery-preview"&gt;     {% for image in images %}         &lt;div class="gallery-item"&gt;             &lt;img src="{{ url_for('static', filename='images/' + image) }}" alt="Random Image" class="preview" /&gt;         {% endfor %}     &lt;/div&gt; &lt;h1&gt;Links to other pages&lt;/h1&gt; &lt;p&gt;&lt;a href="/all_muscles"&gt;View All Muscles&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&lt;a href="/all_exercises"&gt;View All Exercises&lt;/a&gt;&lt;/p&gt;</pre>

Welcome to the Exercise Database!

Image Gallery Preview

Links to other pages

[View All Muscles](#)

[View All Exercises](#)

Ray Colvin ©2023

This is my code for the muscle page, where the muscles are displayed in a grid that will

```
app.route('/all_muscles')
def all_muscles():
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("SELECT * FROM Muscles")
    muscles = cur.fetchall()
    conn.close()
    return render_template('all_muscles.html', title='Muscles', muscles=muscles)

{% extends 'layout.html' %}
{% block content %}


Home
Muscles
Exercises


<form action="{{ url_for('search') }}" method="get" class="topnav">
    <input type="text" name="query" placeholder="Search" required />
</form>

```

Muscles:

- Ab
- Forearms
- Calves
- Tricep
- Lats
- Back
- Glutes
- Hamstrings
- Quadriceps
- Biceps
- Triceps
- Shoulders

Ray Colvin ©2023

```
# Display exercises
@app.route('/all_exercises')
def all_exercises():
    conn = sqlite3.connect('fitness.db')
    conn.row_factory = sqlite3.Row
    cur = conn.cursor()
    cur.execute("""
        SELECT e.exercise_id, e.exercise_name, m.muscle_name, i.image_url
        FROM Exercises e
        INNER JOIN Muscles m ON e.muscle_id = m.muscle_id
        LEFT JOIN Images i ON i.image_id = e.exercise_id
    """)
    exercises = cur.fetchall()
    conn.close()
    return render_template('all_exercises.html', title='All Exercises', exercises=exercises)

# Extends 'layout.html'
{% block content %}


Home
Muscles
Exercises


<form action="{{ url_for('search') }}" method="get" class="topnav">
    <input type="text" name="query" placeholder="Search" required />
</form>

```

Muscles

Abs	Forearms	Calves	Trapezius
Lats	Back	Glutes	Hamstrings
Quadriceps	Biceps	Triceps	Chest
Shoulders			

Ray Colvin ©2023

This is the code for my exercise page, where they are listed in a grid with images.

Exercises

Face Pulls	Shrugs	Leg Raise	Russian Twist
Pushups	Plank	Situp	Deadlift

```

#app.route('/all_exercises')
def all_exercises():
    conn = sqlite3.connect('fitness.db')
    conn.row_factory = sqlite3.Row
    cur = conn.cursor()
    cur.execute("""
        SELECT e.exercise_id, e.exercise_name, m.muscle_name, i.image_url
        FROM Exercises e
        INNER JOIN Muscles m ON e.muscle_id = m.muscle_id
        LEFT JOIN Images i ON i.image_id = e.exercise_id
    """)
    exercises = cur.fetchall()
    conn.close()
    return render_template('all_exercises.html', title='All Exercises', exercises=exercises)

1  {% extends 'layout.html' %}
2  {% block content %}
3  <div class="topnav">
4      <a href="/">>Home</a>
5      <a href="/all_muscles">Muscles</a>
6      <a class="active" href="/all_exercises">Exercises</a>
7      <form action="{{ url_for('search') }}" method="get" class="topnav">
8          <input type="text" name="query" placeholder="Search" required />
9      </form>
10 </div>
11 <h1>Exercises</h1>
12 <div class="exercise-grid">
13     {% for exercise in exercises %}
14         <div class="container">
15             
17                 <p><a href="#{{exercise[0]}}>{{exercise[1]}}</a></p>
18             </div>
19         </div>
20     {% endfor %}
21 </div>
22 {% endblock %}
23 
```

```

@app.route("/exercise/<int:id>")
def exercise(id):
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    cur.execute("""
        SELECT e.exercise_id, e.exercise_name, m.muscle_name, i.image_url
        FROM Exercises e
        INNER JOIN Muscles m ON e.muscle_id = m.muscle_id
        LEFT JOIN Images i ON i.image_id = e.exercise_id
        WHERE e.exercise_id = ?
    """, (id,))
    exercise = cur.fetchone()
    title = 'Exercises - ' + str(exercise[1])
    conn.close()
    return render_template('exercise.html', title=title, exercise=exercise)

```

```

{% extends 'layout.html' %}
{% block content %}
<div class="topnav">
<a href="/">>Home</a>
<a href="/all_muscles">Muscles</a>
<a class="active" href="/all_exercises">Exercises</a>
<form action="{{ url_for('search') }}" method="get" class="topnav">
<input type="text" name="query" placeholder="Search" required />
</form>
</div>
<h1>{{exercise[1]}}</h1>
<h2>{{ exercise[1] }}</h2>
<h3>{{ muscle }}</h3>
<ul>
    <li>{{ exercise[2] }}</li>
</ul>
<div style="margin-top: 20px;">
    
</div>
<p>No image available for this exercise.</p>
</div>
{% endblock %}

```

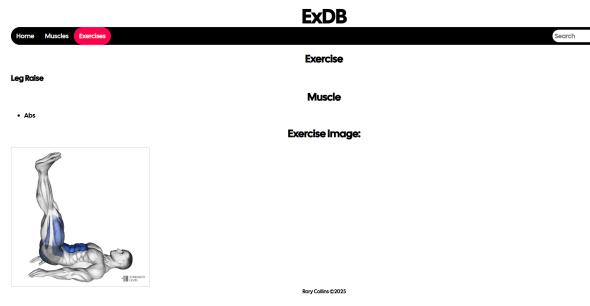
  

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <meta http-equiv="X-UA-Compatible" content="ie=edge" />
        <title>{{ title }}</title>
        <link rel="stylesheet" href="{{ url_for('static', filename='favicon.png') }}" type="image/png" />
        <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}"/>
    </head>
    <body>
        <!-- Header -->
        <header>{{ include('header.html') }}</header>
        <!-- Content -->
        <div>
            {{ content }}
        </div>
        <!-- Footer -->
        <footer>
            <small>Copyright 2025, Roy Collins Roy, 2025</small>
            <small>Disclaimer: The information provided here is for general educational purposes only. It is not a substitute for professional medical advice, diagnosis, or treatment. Always consult with a healthcare professional before starting any new exercise program. Roy Collins Roy, 2025</small>
        </footer>
    </body>
</html>

```

This is my layout.html template used on every page of the website. I tried to keep it simple and included an icon, a short footer for copyright, and a short disclaimer for my users safety. It is also responsible for the title and other necessities like the character set, viewport size, and css.



Roy Collins ©2025

Disclaimer: The information provided here is for general educational purposes only. It is not a substitute for professional medical advice, diagnosis, or treatment. Always consult with a healthcare professional before starting any new exercise program. Roy Collins Roy, 2025

```

templates > header.html > <h1>
1  <h1 style="font-size: 60px; margin: 0; padding: 0 0 3px 0; line-height: 1; text-align: center;">ExDB</h1>
2
3

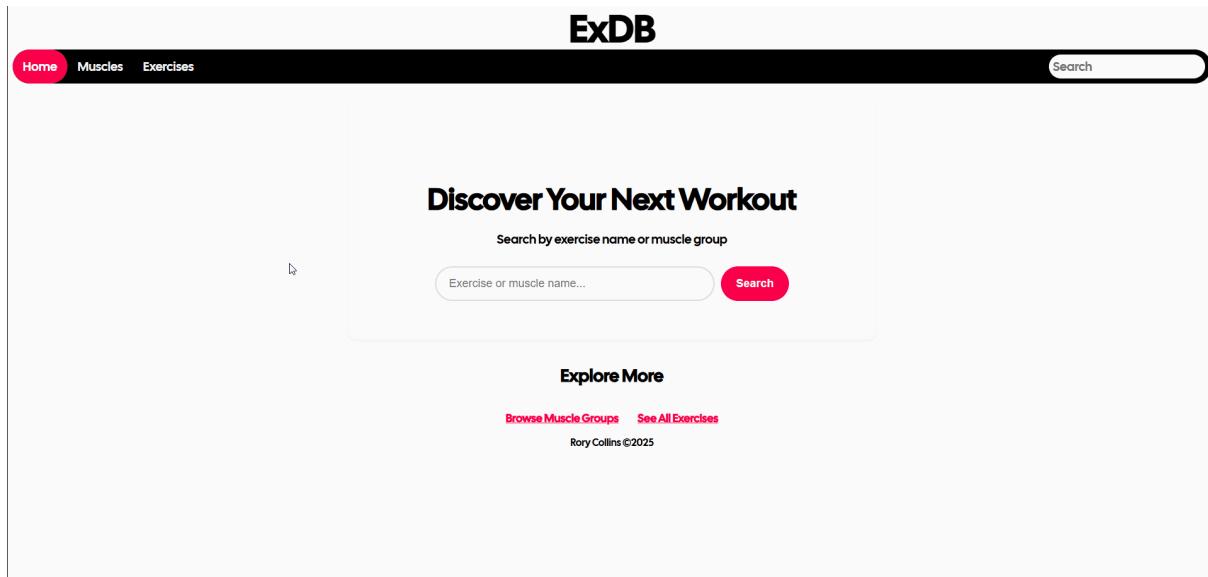
```

This is my header.html template used in the layout (every page of the website), and this is responsible for creating the title shown at the top of the screen.

# ExDB

Option #1		Option #2
	I chose to swap the text to be centered instead of aligned left to give my website more contrast and show that this is meant to be the title and heading.	
	I chose to remake the home page where instead of it being a gallery, it's something more functional that will allow users to search through the database.	

## END OF SPRINT - WHAT I AM TESTING



This is a screen recording of my website, showcasing the home page, where users are able to search through exercises and muscle groups, the muscles page, where users can filter by what muscle they want to target, the exercises page which showcases all of the exercises with their accompanying images and the nav bar that links them all together. Notice how search inputs have been safeguarded by applying a limit to the amount of characters,

helping ensure my website is able to stay safer as well as still be usable for the average user.

## Testing

Test ID	Feature	Interaction	Expected result	Actual result
1	Task bar links	Click muscles/exercises button	Move to muscles/exercises	Passed
2	Home page links	Click view all muscles/exercises link	Move to muscles/exercises	Passed
3	Search bar	Type e	Show all muscles/exercises containing e	Passed
4	Search bar	Type nothing	Does not change page, says required field	Passed
5	Search bar	Enter a letter not contained in anything	Show no results	Passed
6	Exercise title links	Click title text to go to more detailed view of exercise	Changes page to more detailed exercise information	Passed
7	Exercise images + title display	View exercise page	Exercise images displayed in grid	Passed
8	Muscles page	View muscle page	Muscle names displayed in list	Passed
9	Home page search input	Type e	Show all muscles/exercises containing e	Passed
10	Home page search input	Type nothing	Does not change page, says required field	Passed
11	Home page search input	Enter a letter not contained in anything	Shows no results	Passed

12	Home page search input	Enter large amount of characters	Limits characters to 25	Passed
13	Nav bar search input	Enter large amount of characters	Limits characters to 25	Passed
14	Header and footer display	Header and footer are displayed correctly	Displays "ExDB" and copyright + disclaimer	Passed
15	Character set was initialised correctly	Test if language is correct	Displays english and full text	Passed
16	Viewport size was initialised correctly	Test if viewport size is correct	Displays correctly	Passed

## CLIENT / STAKEHOLDER / EXPERT TESTING & FEEDBACK

*Jonty* - Programming Student

I like the aesthetics of the website, it looks very professional with the way that the elements have had CSS used.

*Isaac* - Programming Student

I like the navigation bar and think that it follows external consistency conventions. I also think that making it rounded helps it flow in the website.

*Adi* -

I think that you filled the whitespace well and I think applying a limit to the search inputs will help avoid malicious attacks.

## END USER TESTING & FEEDBACK

*Elijah* -

As mentioned earlier, Elijah regularly exercises, making him a potential end user

I think that the website will be helpful for me to look at new exercises, and I like the search because it makes it easy for me to find what I'm looking for. This will be a helpful tool for me to use when I am making my workout plans.

## ANY MINOR CHANGES

```
<footer style="text-align: center; font-size: 12px; color: #666;">
  Disclaimer: The information provided here is for general educational purposes only. It is not a substitute for professional medical advice, diagnosis, or treatment. Always consult with a healthcare professional before starting any new exercise program.
</footer>
</body>
</html>
```

[Browse Muscle Groups](#) [See All Exercises](#)

Rory Collins ©2025

Disclaimer: The information provided here is for general educational purposes only. It is not a substitute for professional medical advice, diagnosis, or treatment. Always consult with a healthcare professional before starting any new exercise program.

I added a short disclaimer to ensure that my users understand that this information may not be perfect and that they should consult experts or doctors before they do anything risky.

```
# 404 Page for when page not found
@app.errorhandler(404)
def not_found(error):
    return render_template("404.html", title='Page Not Found')
```

```
{% extends "layout.html" %}
{% block content %}
  <h1>Oops! Looks like the page doesn't exist anymore</h1>
  <p>Redirecting you back to the home page in a few seconds...</p>

  <script>
    setTimeout(function() {
      window.location.href = "/";
    }, 3000);
  </script>
{% endblock %}
```

## ExDB

**Oops! Looks like the page doesn't exist anymore**

Redirecting you back to the home page in a few seconds...

Rory Collins ©2025

Disclaimer: The information provided here is for general educational purposes only. It is not a substitute for professional medical advice, diagnosis, or treatment. Always consult with a healthcare professional before starting any new exercise program.

I also added a simple 404 page with CSS formatting that will automatically redirect the user back to the home page after a short amount of time.

[insert images of your final outcome - in the working file (from the software used) and what the user is using/seeing]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Initialise character set, viewport size, title, icon, and css -->
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ExDB - {{ title }}</title>
  <link rel="icon" href="{{ url_for('static', filename='favicon.png') }}" type="image/png">
  <link rel="stylesheet" href="\static\css\style.css">
</head>
<body>
  <!-- Set header and footer -->
  <header>% include 'header.html' %</header>
  <main>% block content %<% endblock %</main>
  <footer style="text-align: center; font-size: 12px; color: #666;">
    Disclaimer: The information provided here is for general educational purposes only. It is not a substitute for professional medical advice, diagnosis, or treatment. Always consult with a healthcare professional before starting any new exercise program.
  </footer>
</body>
</html>
```

This is my html code that will display the footer, header, and initialise character set, viewport size, page title, and icon.

```
# 404 Page for when page not found
@app.errorhandler(404)
def not_found(error):
    return render_template("404.html", title='Page Not Found')

{% extends "layout.html" %}
{% block content %}
<h1>Oops! Looks like the page doesn't exist anymore</h1>
<p>Redirecting you back to the home page in a few seconds...</p>

<script>
    setTimeout(function() {
        window.location.href = "/";
    }, 3000);
</script>
{% endblock %}
```

This is my 404 error page that will display a simple page and redirect the user back to the home page.

```
# Home page with search engine
@app.route('/')
def home():
    return render_template('home.html', title="Home")
```

```

{% extends 'layout.html' %}

{% block content %}

<!-- Navigation Bar -->


Home
Muscles
Exercises
<form action="{{ url_for('search') }}" method="get" class="topnav-search">
| <input type="text" name="query" placeholder="Search" maxlength="25" required />
</form>
</div>

<!-- Search Section -->


<h1>Discover Your Next Workout</h1>
<p>Search by exercise name or muscle group</p>
<form action="{{ url_for('search') }}" method="get" class="hero-search-form">
| <input type="text" name="query" placeholder="Exercise or muscle name..." maxlength="25" required>
| <button type="submit">Search</button>
</form>


</div>

<!-- Links to other pages -->
<section class="links-section container">
<h2>Explore More</h2>
<ul>
| <li><a href="/all_muscles">Browse Muscle Groups</a></li>
| <li><a href="/all_exercises">See All Exercises</a></li>
</ul>
</section>


```

{% endblock %}

This is the code for my home page with the search engine.

```

# Display all exercises in grid with images
@app.route('/all_exercises')
def all_exercises():
    # Connect to database
    conn = sqlite3.connect('fitness.db')
    conn.row_factory = sqlite3.Row
    cur = conn.cursor()
    # Select exercise id and name, muscle name and image url
    cur.execute("""
        SELECT e.exercise_id, e.exercise_name, m.muscle_name, i.image_url
        FROM Exercises e
        INNER JOIN Muscles m ON e.muscle_id = m.muscle_id
        LEFT JOIN Images i ON i.image_id = e.exercise_id
    """)
    exercises = cur.fetchall()
    conn.close()
    return render_template('all_exercises.html', title='All Exercises', exercises=exercises)

```

```

{% extends 'layout.html' %}
{% block content %}
<!--Top navigation bar and search box-->


|   <a href="/">Home</a>
|   <a href="/all_muscles">Muscles</a>
|   <a class="active" href="/all_exercises">Exercises</a>
|   <form action="{{ url_for('search') }}" method="get" class="topnav">
|       <input type="text" name="query" placeholder="Search" maxlength="25" required />
|   </form>


<!--Display exercise images in a grid format with name-->


# Exercises



{% for exercise in exercises %}


![Exercise Image]({{ url_for('static', filename='images/' + exercise[3]) }})

|   <p><a href="exercise/{{exercise[0]}}>{{exercise[1]}}</a></p>


{% endfor %}


{% endblock %}

```

This is the code for the exercises page that displays all the exercises with their images.

```

# Display specific exercise information with muscle name and exercise image
@app.route("/exercise/<int:id>")
def exercise(id):
    # Connect to database
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    # Select exercise id and name, muscle name and specific image url
    cur.execute("""
        SELECT e.exercise_id, e.exercise_name, m.muscle_name, i.image_url
        FROM Exercises e
        INNER JOIN Muscles m ON e.muscle_id = m.muscle_id
        LEFT JOIN Images i ON i.image_id = e.exercise_id
        WHERE e.exercise_id = ?
    """, (id,))
    exercise = cur.fetchone()
    title = 'Exercises - ' + str(exercise[1])
    conn.close()
    return render_template('exercise.html', title=title, exercise=exercise)

```

```

(% extends 'layout.html' %)
{%- block content %}
<!-- Top navigation bar and search box--&gt;
<div class="topnav">
Home
Muscles
Exercises
<form action="{{ url_for('search') }}" method="get" class="topnav">
    <input type="text" name="query" placeholder="Search" maxlength="25" required />
</form>

```

This is the code for my specific exercise page that displays more detailed information.

```

# Display all muscles in grid
@app.route('/all_muscles')
def all_muscles():
    # Connect to database
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    # Select all muscle ids and names
    cur.execute("SELECT * FROM Muscles")
    muscles = cur.fetchall()
    conn.close()
    return render_template('all_muscles.html', title='Muscles', muscles=muscles)

```

```

{% extends 'layout.html' %}
{% block content %}
<!-- Top navigation bar and search box--&gt;
<div class="topnav">
Home
Muscles
Exercises
<form action="{{ url_for('search') }}" method="get" class="topnav">
    <input type="text" name="query" placeholder="Search" maxlength="25" required />
</form>

```

This is the code for my all muscles page that displays all the muscles in a grid.

```

# Display specific muscle information and what exercises target it
@app.route("/muscle/<int:id>")
def muscle_exercises(id):
    # Connect to database
    conn = sqlite3.connect('fitness.db')
    cur = conn.cursor()
    # Select specific muscle name
    cur.execute("SELECT muscle_name FROM Muscles WHERE muscle_id = ?", (id,))
    muscle = cur.fetchone()
    # Select specific exercise id and names
    cur.execute("SELECT exercise_id, exercise_name FROM Exercises WHERE muscle_id = ?", (id,))
    exercises = cur.fetchall()
    title = 'Muscles - ' + str(muscle[0])
    conn.close()
    return render_template('muscle_exercises.html', title=title, muscle=muscle, exercises=exercises)

{% extends 'layout.html' %}
{% block content %}
<!--Top navigation bar and search box--&gt;
<div class="topnav">
Home
Muscles
Exercises
<form action="{{ url_for('search') }}" method="get" class="topnav">
    <input type="text" name="query" placeholder="Search" maxlength="25" required />
</form>

```

```

{% extends 'layout.html' %}

{% block content %}

<!--Top navigation bar and search box-->


<a class="active" href="/">Home</a>
    <a href="/all_muscles">Muscles</a>
    <a href="/all_exercises">Exercises</a>
    <form action="{{ url_for('search') }}" method="get" class="topnav">
        <input type="text" name="query" placeholder="Search" maxlength="25" required />
    </form>


<!--Show all results in list muscle + exercise-->
<h1>Search Results for "{{ query }}</h1>\n
<!--Display muscle results in list-->
<h2>Muscles</h2>
{% if muscles %}
    <ul>
        {% for muscle in muscles %}
            <li><a href="{{ url_for('muscle_exercises', id=muscle[0]) }}>{{ muscle[1] }}</a></li>
        {% endfor %}
    </ul>
{% else %}
    <p>No muscles found.</p>
{% endif %}
<!--Display muscle results in list-->
<h2>Exercises</h2>
{% if exercises %}
    <ul>
        {% for exercise in exercises %}
            <li><a href="{{ url_for('exercise', id=exercise[0]) }}>{{ exercise[1] }}</a></li>
        {% endfor %}
    </ul>
{% else %}
    <p>No exercises found.</p>
{% endif %}
{% endblock %}

```

This is the code for my search function that will find information in my database similar to the input.

```

# Run code on port:5000
if __name__ == '__main__':
    app.run(debug=True, port=5000)

```

This is the part that makes my code run on port:5000.

## FINAL OUTCOME

## Discover your next Workout

Search by exercise name or muscle group

### Explore More

[Browse Muscle Groups](#) [See All Exercises](#)

Rory Collins ©2025

Disclaimer: The information provided here is for general educational purposes only. It is not a substitute for professional medical advice, diagnosis, or treatment. Always consult with a healthcare professional before starting any new exercise program.