**Title:** Gym Bros **Who:**

Breanna Harris
Idaly Gomez-Pena
Joanne Li
Joe Voirol
John Kim
Nicole Paraschiv

**Project Description:**

Introducing Gym Bros, an all-in-one fitness companion application designed to help the user track and achieve consistency. Our application is packed with a multitude of features that can seamlessly integrate itself into the user's fitness journey. With Gym Bros, you can track and log up to nine pre-selected workouts that encapsulate every main part of the body, timing each session down to the milliseconds, and providing the user with a detailed calorie burned counter dynamic to the type of workout. Stay motivated by recording user metrics such as body weight, height, and other fitness details in our comprehensive registration process that stores your details in our expansive database for quite possibly forever.

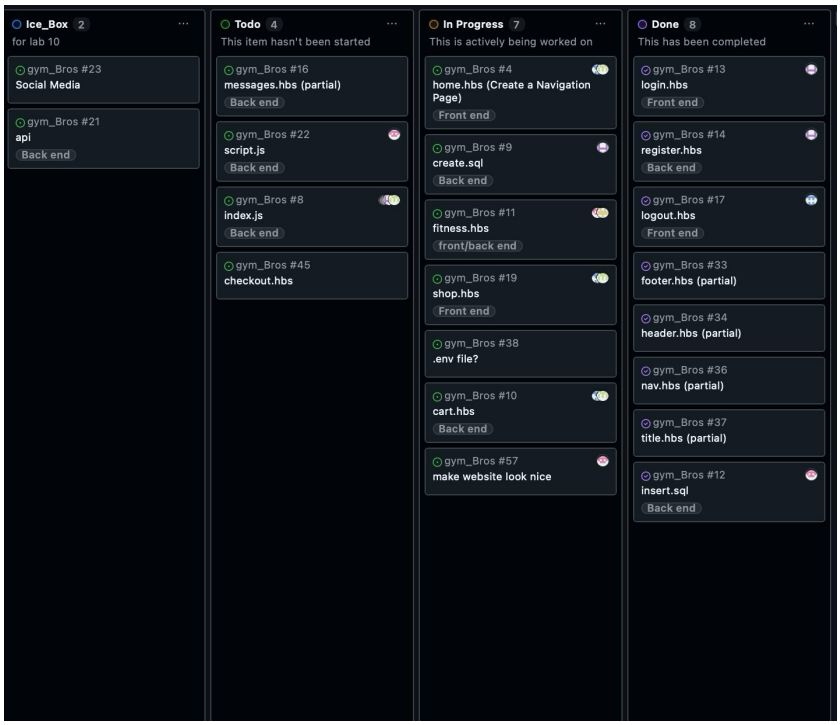
Competition is also heavily supported with a leaderboard that tracks and ranks users based on an internal point system, allowing our users to stay at the top by maintaining their fitness.

Additionally, users who want to indulge their capitalistic desires are encouraged to check out the Gym Bros shop, providing users a method to purchase gear and supplements from popular online shopping centers.

Whether you are a beginner or a seasoned athlete, the Gym Bros application provides all the tools to monitor your personal fitness.

**Project Tracker - GitHub project board:**

Link to Project Tracker: https://github.com/users/BreHarris0917/projects/2




**Video:** 5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.

**VCS:** https://github.com/BreHarris0917/gym_Bros.git

**Contributions:**

Breanna's Contribution:
For my contribution I worked on the cart file and shopping file, it was something we ended up removing to do something else, but in the beginning it was my main focus. I also worked on a workout screen that linked to some at home workouts for people to do. The home workout coincides with the 9 workouts that are on the workout page along with other workouts. The workouts range from beginner to more advanced, where some need equipment and others don't. The workout page provides a variety of workouts for people at any point in their workout journey.

Idaly's Contribution:
For my contribution, I worked on the database and its connection between the back and front end. My focus was on features like the login, register, home, and leaderboard pages. On the home page, I added functionality for viewing profiles and changing passwords. The

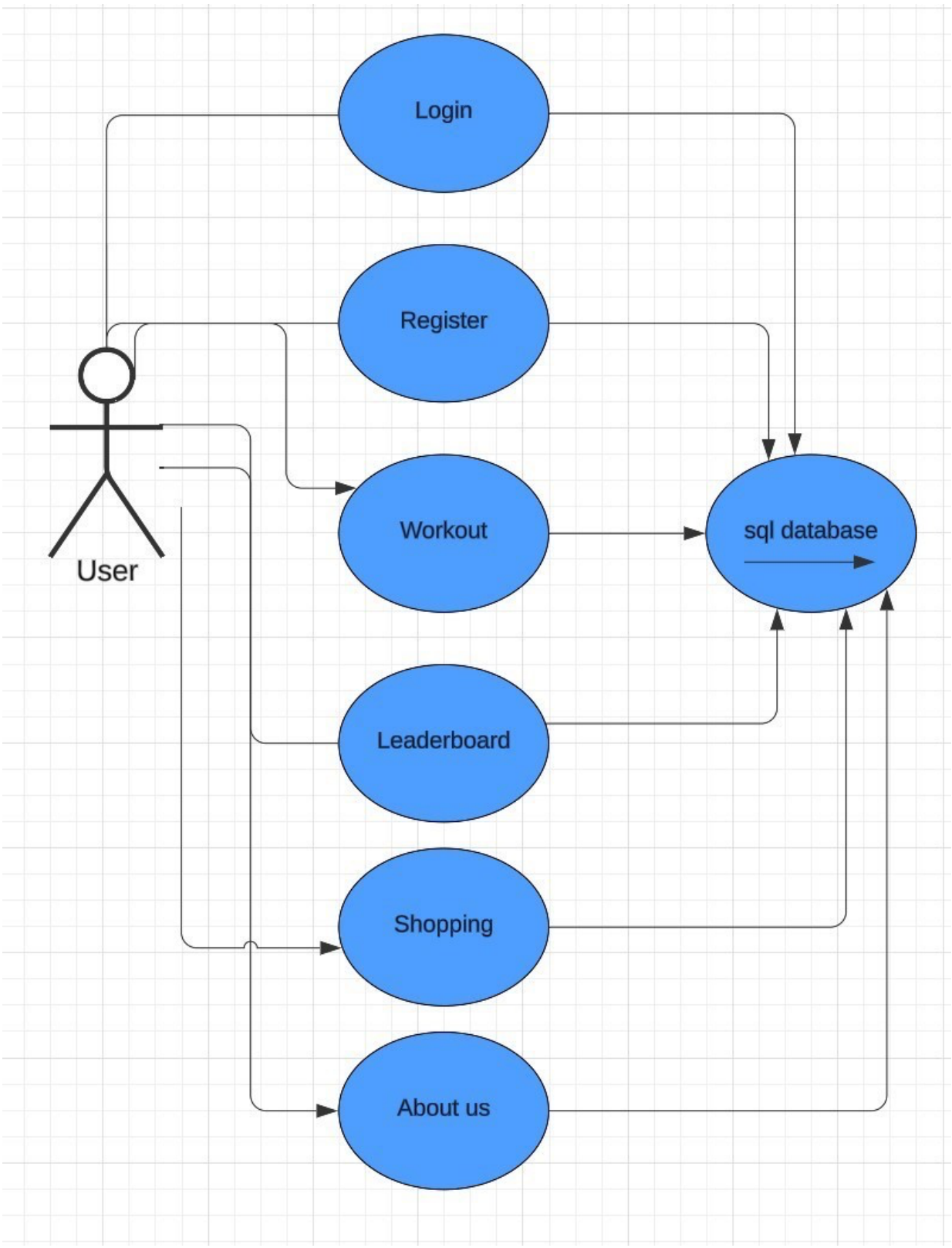
leaderboard dynamically pulls and displays user points from the database. I also resolved issues with storing user data securely in the database.
Joanne's Contribution:

Joe's Contribution:
My contributions to the project include working on nav.hbs as well as title.hbs. These two pages utilize handlebar.js to add functionality to the basic display and navigation around the website in a dynamic manner. All unit testing for our project was completed by me during lab11. Added photos and rewrote the “about us” page, these additions contributed to the completeness of that page. All creation credit for “about us” goes to John.

John's Contribution:
I worked mainly on the fitness tracking part of the application. The timers, the workout modals, and scripts that pertain to the fitness page were developed over time. I also contributed to some overall “beautification” primarily on the login and registration page, as well as the addition of an About Us page.

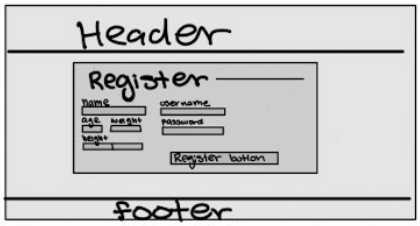
Nicole's Contribution:
I primarily focused on developing the storefront for our project, ensuring it was functional and user-friendly. I experimented with diberent ways to display products and problemsolved issues related to user payments and interactions. To simplify the process, I decided to link users directly to the sellers’ websites for purchases. I also added a search bar to help users easily locate specific items available in our shop. Most of my work involved building and refining the HTML components, as the shop’s structure heavily relied on HTML. This approach allowed me to focus on creating an intuitive and visually appealing storefront.

**Use Case Diagram:**

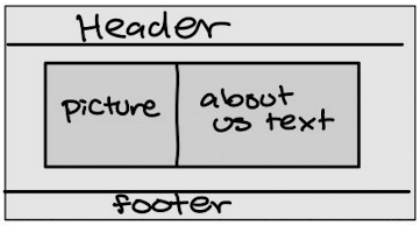

Login
Register
Workout
Leaderboard
Shopping
About us
sql database
User

**Wireframes:**

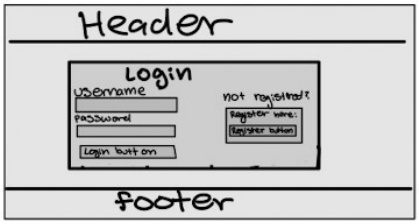
## Register



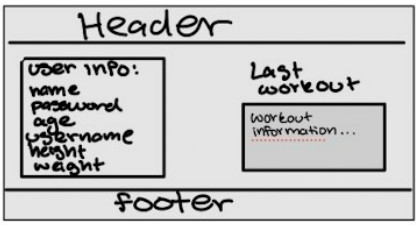

## about us



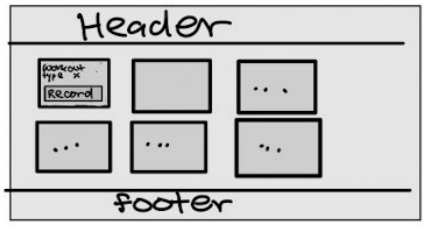

## Login



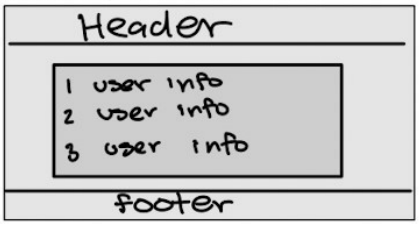

## Home



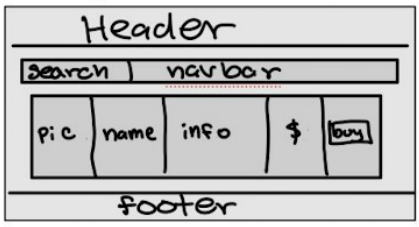

## workout



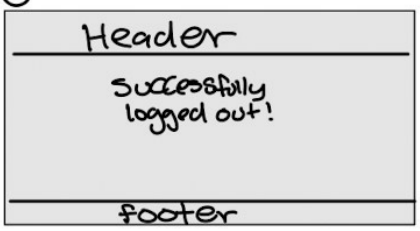

## leaderboard

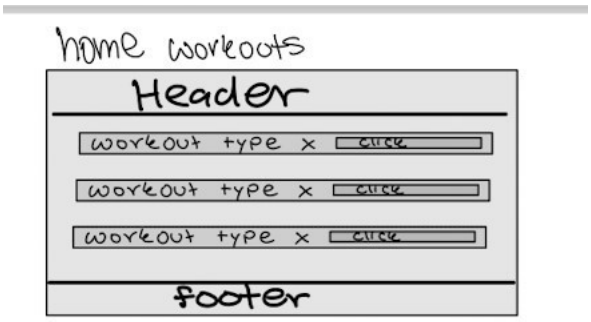



## shop




## logout






**Test results:** I

```
db-1   | The files belonging to this database system will be owned by user "postgres".
db-1   | This user must also own the server process.
db-1   |
db-1   | The database cluster will be initialized with locale "en_US.utf8".
db-1   | The default database encoding has accordingly been set to "UTF8".
db-1   | The default text search configuration will be set to "english".
db-1   |
db-1   | Data page checksums are disabled.
db-1   |
db-1   | fixing permissions on existing directory /var/lib/postgresql/data ... ok
db-1   | creating subdirectories ... ok
db-1   | selecting dynamic shared memory implementation ... posix
db-1   | selecting default max_connections ... 100
db-1   | selecting default shared_buffers ... 128MB
db-1   | selecting default time zone ... Etc/UTC
db-1   | creating configuration files ... ok
db-1   | running bootstrap script ... ok
web-1  |
web-1  | > testandrun
web-1  | > npm run prestart && npm run test && npm start
web-1  |
db-1   | performing post-bootstrap initialization ... ok
web-1  |
web-1  | > prestart
web-1  | > npm install
web-1  |
db-1   | syncing data to disk ... ok
db-1   |
```

```
db-1  |
db-1  | Success. You can now start the database server using:
db-1  |
db-1  |     pg_ctl -D /var/lib/postgresql/data -l logfile start
db-1  |
db-1  | initdb: warning: enabling "trust" authentication for local connections
db-1  | You can change this by editing pg_hba.conf or using the option -A, or
db-1  | --auth-local and --auth-host, the next time you run initdb.
db-1  | waiting for server to start....2024-12-04 18:33:11.059 UTC [49] LOG:  starting
PostgreSQL 14.13 (Debian 14.13-1.pgdg120+1) on aarch64-unknown-linux-gnu,
compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db-1  | 2024-12-04 18:33:11.059 UTC [49] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
db-1  | 2024-12-04 18:33:11.061 UTC [50] LOG:  database system was shut down at
2024-12-04 18:33:10 UTC
db-1  | 2024-12-04 18:33:11.063 UTC [49] LOG:  database system is ready to accept
connections
db-1  |  done
db-1  | server started
db-1  | CREATE DATABASE
db-1  |
db-1  |
db-1  | /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-
initdb.d/create.sql
db-1  | CREATE TABLE
db-1  | CREATE TABLE
db-1  | CREATE TABLE
db-1  |
db-1  |
db-1  | /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-
initdb.d/insert.sql
db-1  |
db-1  |
db-1  | waiting for server to shut down...2024-12-04 18:33:11.249 UTC [49]
LOG:  received fast shutdown request
db-1  | .2024-12-04 18:33:11.250 UTC [49] LOG:  aborting any active transactions
db-1  | 2024-12-04 18:33:11.251 UTC [49] LOG:  background worker "logical replication
launcher" (PID 56) exited with exit code 1
```

```
db-1   | 2024-12-04 18:33:11.251 UTC [51] LOG:  shutting down
db-1   | 2024-12-04 18:33:11.273 UTC [49] LOG:  database system is shut down
db-1   |  done
db-1   | server stopped
db-1   |
db-1   | PostgreSQL init process complete; ready for start up.
db-1   |
db-1   | 2024-12-04 18:33:11.360 UTC [1] LOG:  starting PostgreSQL 14.13 (Debian
14.13-1.pgdg120+1) on aarch64-unknown-linux-gnu, compiled by gcc (Debian 12.2.0-
14) 12.2.0, 64-bit
db-1   | 2024-12-04 18:33:11.360 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port
5432
db-1   | 2024-12-04 18:33:11.360 UTC [1] LOG:  listening on IPv6 address "::", port 5432
db-1   | 2024-12-04 18:33:11.361 UTC [1] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
db-1   | 2024-12-04 18:33:11.362 UTC [68] LOG:  database system was shut down at
2024-12-04 18:33:11 UTC
db-1   | 2024-12-04 18:33:11.364 UTC [1] LOG:  database system is ready to accept
connections
web-1  |
web-1  | up to date, audited 368 packages in 1s
web-1  |
web-1  | 86 packages are looking for funding
web-1  |   run `npm fund` for details
web-1  |
web-1  | 12 vulnerabilities (3 low, 6 high, 3 critical)
web-1  |
web-1  | To address issues that do not require attention, run:
web-1  |   npm audit fix
web-1  |
web-1  | To address all issues (including breaking changes), run:
web-1  |   npm audit fix --force
web-1  |
web-1  | Run `npm audit` for details.
web-1  |
web-1  | > test
web-1  | > mocha
web-1  |
```

```
web-1  |
web-1  |
web-1  | Database connection successful
web-1  |   ✓ Returns server status
web-1  |   ✓ Posts data successfully
web-1  |  Server!
web-1  |    ✓ Returns the default welcome message
web-1  |
web-1  |  Testing Add User API
web-1  |    ✓ positive : /register (78ms)
web-1  |
web-1  |  Testing Add User API
web-1  |    ✓ Negative: /register with invalid username type
web-1  |
web-1  |  Testing Render
web-1  |    ✓ test "/login" route should render with an html response
web-1  |
web-1  |  Profile Route Tests
web-1  |    GET /fitness
web-1  |      ✓ should return 401 if user is not authenticated
web-1  | in fitness
web-1  |      ✓ should return user profile when authenticated (75ms)
web-1  |
web-1  |
web-1  |  8 passing (281ms)
web-1  |
web-1  |
web-1  | > prestart
web-1  | > npm install
web-1  |
web-1  |
web-1  | up to date, audited 368 packages in 419ms
web-1  |
web-1  | 86 packages are looking for funding
web-1  |   run `npm fund` for details
web-1  |
web-1  | 12 vulnerabilities (3 low, 6 high, 3 critical)
web-1  |
```

```
web-1  | To address issues that do not require attention, run:
web-1  |   npm audit fix
web-1  |
web-1  | To address all issues (including breaking changes), run:
web-1  |   npm audit fix --force
web-1  |
web-1  | Run `npm audit` for details.
web-1  |
web-1  | > start
web-1  | > nodemon index.js
web-1  |
web-1  | [nodemon] 2.0.22
web-1  | [nodemon] to restart at any time, enter `rs`
web-1  | [nodemon] watching path(s): *.*
web-1  | [nodemon] watching extensions: js,mjs,json
web-1  | [nodemon] starting `node index.js`
web-1  | Database connection successful
```

**Deployment:** https://gym-bros.onrender.com/login