Health App Documentation 33828013

Outline

PulsePro keeps things straightforward. It’s a web app that lets you track your workouts without any fuss. Once you sign up and log in, you can start logging your sessions—what you did, how long you spent, how tough it was, even little notes for yourself. Everything stays organized, so you always know where to find your fitness history. Add a new workout, edit an old one, or delete a session if you need to. With PulsePro, you’re calling the shots on your fitness journey.

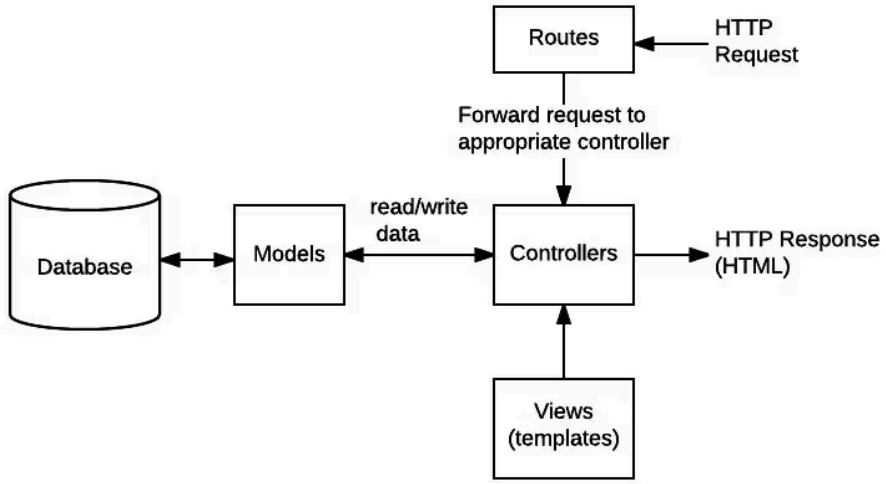
Looking for a specific workout? Just search by keyword, type, or date. If you want to see the big picture, PulsePro gives you a pie chart that breaks down your workouts by type. It’s a simple way to spot trends or just see what you’ve been up to.

Security’s tight here. Your password gets hashed, and only logged-in users can access or change workout data. The interface keeps things clean and straightforward, just what you need. PulsePro uses server-side rendering, database integration, solid authentication, and clear data visualisation. The focus stays on reliability, security, and making things as simple as possible for you.

Architecture

PulsePro runs on a Model–View–Controller (MVC) setup. The backend relies on Node.js with Express to manage routing, middleware, and all server-side logic. For views, EJS handles templating. User authentication uses express-session, with bcrypt hashing passwords for security.

On the data side, PulsePro connects to a MySQL database, storing user accounts and workout records. The app talks to the database through the mysql2 library. Express serves up static assets like CSS files.



==

Data Model

The app manages its data using two primary tables: users and workouts. In the users table, each individual's unique username and hashed password are stored. The workouts table records every workout, linking each one to a user through a user_id.

Each workout record includes the date, type, duration, intensity, and any comments the user wishes to include. This arrangement allows users to easily monitor their workouts while ensuring their data remains distinct and protected.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| health             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.05 sec)
```

```
mysql> use health;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+------------------+
| Tables_in_health |
+------------------+
| users            |
| workouts         |
+------------------+
2 rows in set (0.00 sec)
```

```
mysql> DESCRIBE users;
+---------------+--------------+------+-----+-------------------+----------------+
| Field         | Type         | Null | Key | Default           | Extra          |
+---------------+--------------+------+-----+-------------------+----------------+
| id            | int(11)      | NO   | PRI | NULL              | auto_increment |
| username      | varchar(50)  | NO   | UNI | NULL              |                |
| password_hash | varchar(255) | NO   |     | NULL              |                |
| created_at    | timestamp    | NO   |     | CURRENT_TIMESTAMP |                |
+---------------+--------------+------+-----+-------------------+----------------+
4 rows in set (0.08 sec)
```

```
mysql> DESCRIBE workouts;
+------------------+-------------+------+-----+---------+----------------+
| Field            | Type        | Null | Key | Default | Extra          |
+------------------+-------------+------+-----+---------+----------------+
| id               | int(11)     | NO   | PRI | NULL    | auto_increment |
| user_id          | int(11)     | NO   | MUL | NULL    |                |
| workout_date     | date        | NO   |     | NULL    |                |
| workout_type     | varchar(50) | NO   |     | NULL    |                |
| duration_minutes | int(11)     | NO   |     | NULL    |                |
| intensity        | varchar(20) | NO   |     | NULL    |                |
| notes            | text        | YES  |     | NULL    |                |
+------------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM workouts;
+----+---------+--------------+--------------+------------------+-----------+---------+
| id | user_id | workout_date | workout_type | duration_minutes | intensity | notes   |
+----+---------+--------------+--------------+------------------+-----------+---------+
|  3 |       1 | 2025-12-19   | Gym          |               20 | low       | NULL    |
|  4 |       1 | 2025-12-21   | Run          |               10 | high      | NULL    |
|  5 |       1 | 2025-12-10   | Swimming     |               21 | high      | 10 laps |
|  6 |       1 | 2025-12-13   | Cycling      |               45 | low       | NULL    |
|  7 |       1 | 2025-12-06   | Yoga         |               33 | high      | NULL    |
+----+---------+--------------+--------------+------------------+-----------+---------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM users;
+----+----------+--------------------------------------------------------------+---------------------+
| id | username | password_hash                                                | created_at          |
+----+----------+--------------------------------------------------------------+---------------------+
|  1 | gold     | $2b$10$1qkDAmg71nAKDpXx6R4/YuOOkUIiXeCbWasBAoIQC6ahE1NPJPucW | 2025-12-12 20:27:41 |
|  2 | henry    | $2b$10$JDhOOp0aqDFWEwQl5d286.sh4dKh3TZDO4MhJuSCB2P/AXOFqBauW | 2025-12-12 20:28:21 |
|  3 | aaaa     | $2b$10$05lAuV7V.Th1Fdvr1bqcieXRLXkEWW.eki0tIcBHDMGEwgnNCiGcO | 2025-12-13 14:55:42 |
| 11 | jojo     | $2b$10$4d.g2QfJfYxmACeKO7T9MuRrDY/gHFGLunLPVKdvX1Sg2N1Mp936e | 2025-12-13 20:13:10 |
+----+----------+--------------------------------------------------------------+---------------------+
4 rows in set (0.01 sec)
```

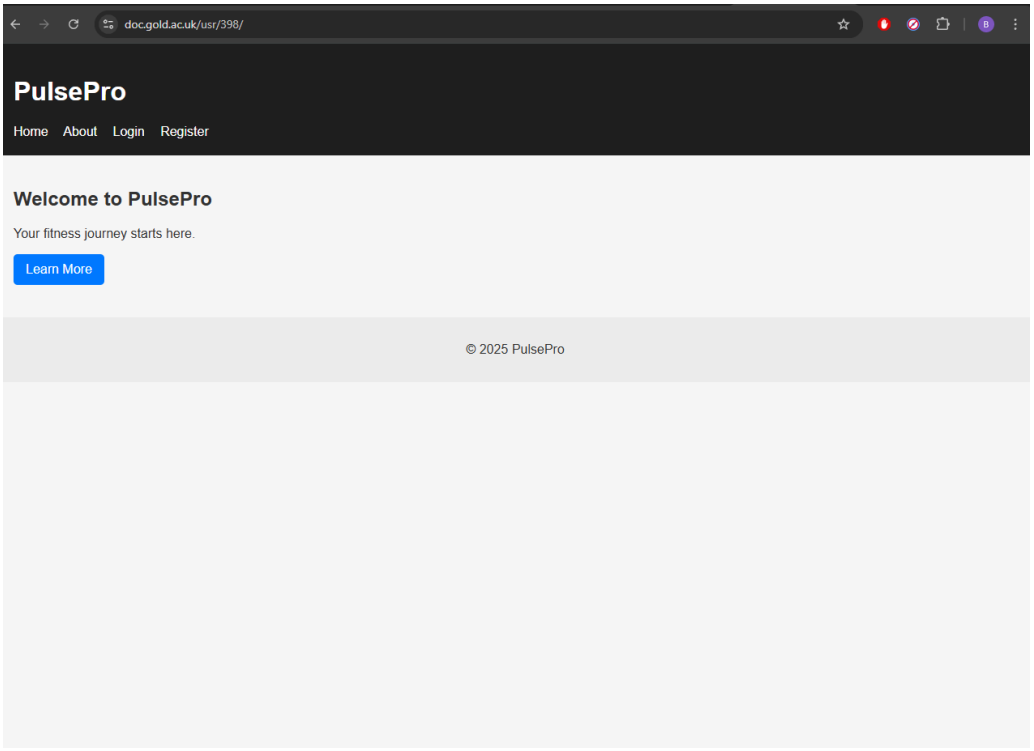
User Functionality

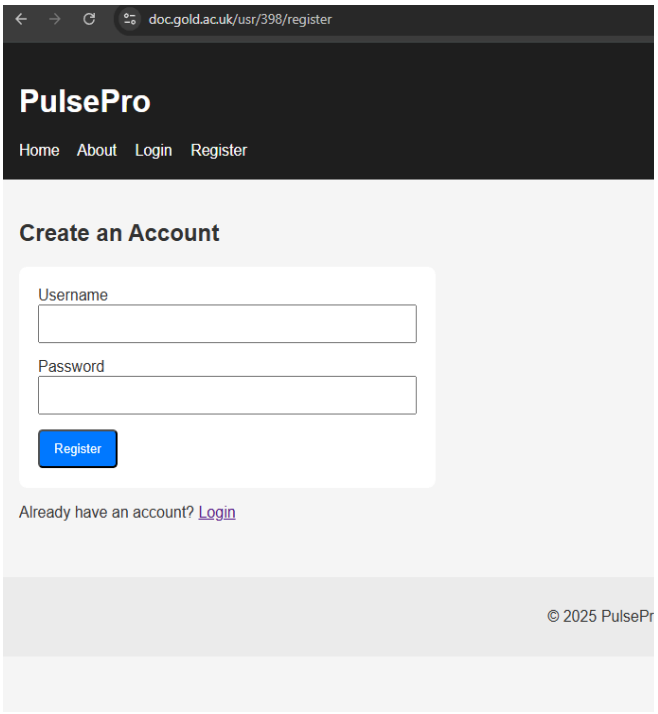
PulsePro simplifies exercise monitoring. Ensures your data remains protected offering users the key features needed to oversee their fitness plans.
Beginning is straightforward. Select an username set a password and sign up for your account. PulsePro employs bcrypt to encrypt your password prior, to storage guaranteeing the security of your data. After registering signing in is easy. The application utilizes session-based authentication allowing you to stay logged in even if you exit and come afterward.

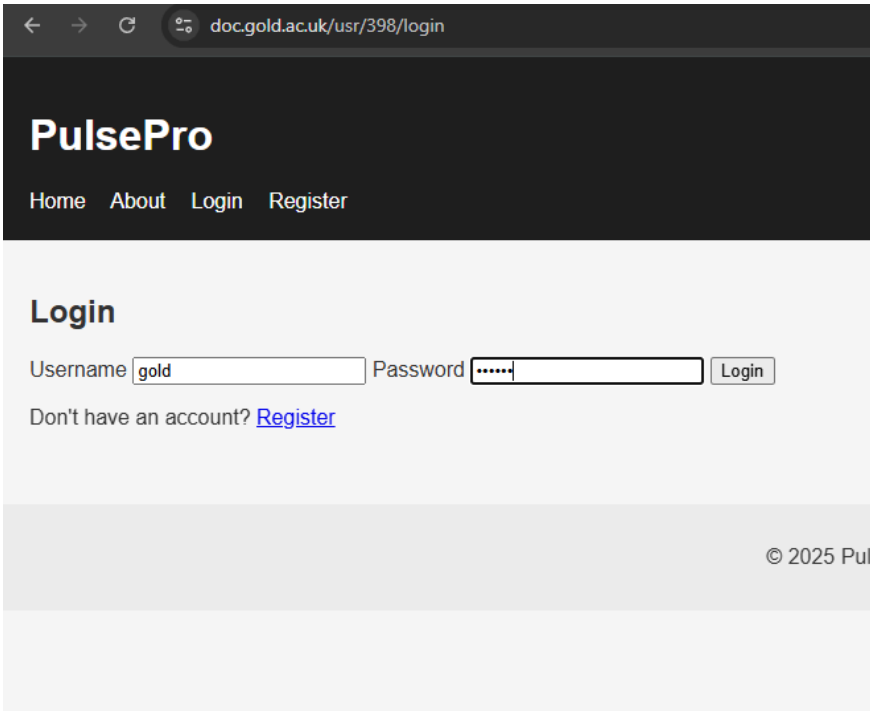
Once you log in you are directed to the workouts page. On this page you can see a record of all workouts you have entered—date, type, duration, intensity and any comments you added. To log a workout simply fill out the brief form. The app verifies all required information, before storing your entry.

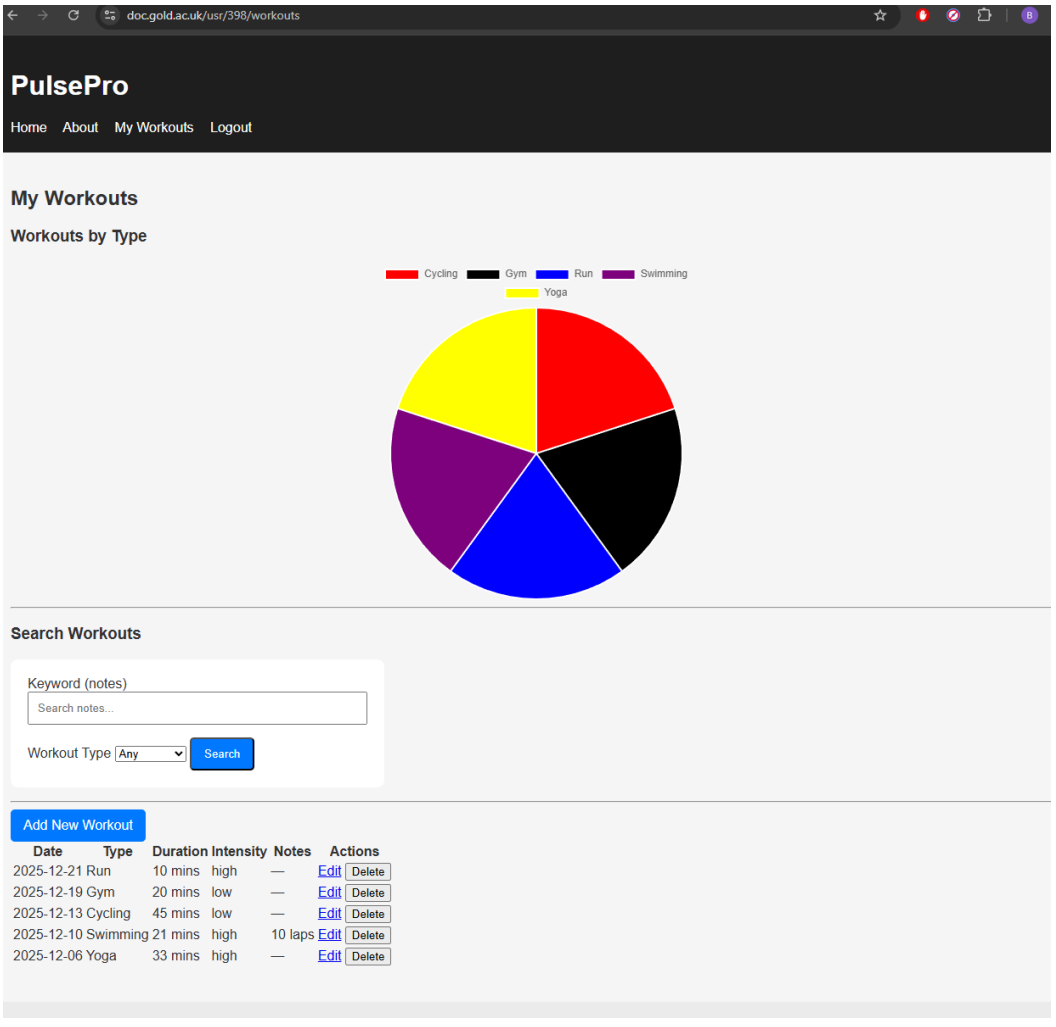
You can. Remove any workout ensuring your logs always reflect your actual activity. Only you are authorised to see or change your workouts—no one else can access them.
If you have built up a record of workouts finding what you want is straightforward. The search tool allows you to narrow down by keyword (from your notes) or workout category.
It's fast. Helps maintain your records tidy.

PulsePro goes beyond just tracking; it helps you recognize your habits as well. A visual overview of your workouts is available thanks to Chart.js. The pie chart displays the distribution of your workout types, offering a clear snapshot of your activity trends.




doc.gold.ac.uk/usr/398/register
PulsePro
Home About Login Register
Create an Account
Username
Password
Register
Already have an account? Login
© 2025 PulsePr


doc.gold.ac.uk/usr/398/login
PulsePro
Home About Login Register
Login
Username gold
Password
Login
Don't have an account? Register
© 2025 Pul


doc.gold.ac.uk/usr/398/workouts
PulsePro
Home About My Workouts Logout
My Workouts
Workouts by Type
Cycling Gym Run Swimming Yoga
Search Workouts
Keyword (notes)
Search notes...
Workout Type Any
Search
Add New Workout
Date Type Duration Intensity Notes Actions
2025-12-21 Run 10 mins high — Edit Delete
2025-12-19 Gym 20 mins low — Edit Delete
2025-12-13 Cycling 45 mins low — Edit Delete
2025-12-10 Swimming 21 mins high 10 laps Edit Delete
2025-12-06 Yoga 33 mins high — Edit Delete

Advanced techniques and Additional features added

Alongside the functions PulsePro incorporates various sophisticated and enhanced capabilities to showcase a wider spectrum of programming methods.
A sophisticated method employed is user verification. Plain text passwords are never kept. Rather the application utilizes the bcrypt library to encode passwords to storing them in the database and to safely match hashes during user sign-, in. This safeguards user information. Exemplifies strong security measures.

```
router.post('/register', async (req, res) => {
  const { username, password } = req.body;
  const hash = await bcrypt.hash(password, 10);
```

The application additionally employs session-based authentication via session. After a user logs in successfully their session data is saved on the server. Verified on secured routes (, like workouts and goals) through middleware. This guarantees that only users who are authenticated have permission to view or alter their information.

```
function ensureLoggedIn(req, res, next) {
    if (!req.session.user) {
      req.session.flash = { type: 'error', message: 'Please log in first.' };
      return res.redirect('/login');
    }
    next();
  }

  module.exports = { ensureLoggedIn };
```

An additional sophisticated capability is data visualization. The workouts page features a pie chart built with Chart.js, which visually breaks down the users workouts by category. This enhances user experience by displaying database information through an understandable graphical representation instead of simple text.

In addition, to CRUD operations PulsePro enhances capabilities by enabling users to modify and remove workouts rather than solely adding them. This enables data lifecycle

management and showcases the application of dynamic routing, form processing and SQL update/delete commands.

Finally, the application includes search functionality that allows users to filter workouts stored in the database based on criteria such as workout type, and notes. This shows effective querying of relational data and enhances the practical usefulness of the system.

```
router.post('/delete/:id', ensureLoggedIn, async (req, res) => {
  await pool.query(
    "DELETE FROM workouts WHERE id=? AND user_id=?",
    [req.params.id, req.session.user.id]
  );


  res.redirect('../..');
});

// saerch
router.post('/search', ensureLoggedIn, async (req, res) => {
  const { keyword, workout_type} = req.body;

  let sql = "SELECT * FROM workouts WHERE user_id = ?";
  let params = [req.session.user.id];

  if (keyword) {
    sql += " AND notes LIKE ?";
    params.push(`%${keyword}%`);
  }

  if (workout_type && workout_type !== "any") {
    sql += " AND workout_type = ?";
    params.push(workout_type);
  }



  const [results] = await pool.query(sql, params);

  res.render('workouts_search', {
    title: "Search Results",
    results
  });
});
```

AI Declaration

I have used AI to help create my style.css file as it doesn’t affect the core parts of this project and allowed me to work with a clearer interface saving time for the core sections of the project which AI wasn’t used for.