

# Sorting Algorithms Documentation

---

## Fit and Brawl Gym Application

---

**Document Version:** 1.0

**Last Updated:** November 7, 2025

---

### Introduction

---

Welcome! This document explains how data is sorted and organized in the **Fit and Brawl Gym** web application.

#### What is Fit and Brawl Gym?

Fit and Brawl Gym is a web-based management system for a fitness and martial arts gym. The system allows:

- **Members** to view classes, book training sessions, and leave feedback
- **Trainers** to manage their schedules
- **Admins** to oversee operations and member subscriptions

#### What You'll Learn

This document focuses on **how information is sorted** in the application - for example:

- How feedback is ordered (newest first vs. most helpful)
  - How booking sessions are arranged by date or time
  - How equipment lists are filtered by status
  - How admin views organize subscription requests
- 

## Table of Contents

---

1. [Quick Overview](#)
  2. [Feedback Page Sorting](#)
  3. [Reservations Page Sorting](#)
  4. [Equipment Page Filtering](#)
  5. [Admin Subscriptions Sorting](#)
  6. [Summary Table](#)
  7. [Technical Details](#)
-

## Quick Overview

---

The application uses **two main approaches** to sort data:

### 1. Server-Side Sorting (Database Level)

- Sorting happens on the server before data reaches your browser
- Used for: Initial page loads, large datasets
- Technology: SQL `ORDER BY` commands
- **Example:** When you first visit the feedback page, reviews are already sorted

### 2. Client-Side Sorting (Browser Level)

- Sorting happens in your web browser using JavaScript
- Used for: Interactive features (clicking sort buttons)
- Technology: JavaScript `.sort()` method
- **Example:** Clicking "Sort by Date" button to reorder your bookings

## Why Use Both?

- **Server-side** is efficient for large amounts of data
  - **Client-side** provides instant, interactive sorting without reloading the page
- 

## Feedback Page Sorting

---

### What This Page Does

The **Feedback page** displays customer reviews and testimonials from gym members. Users can read what others think about the gym's services and facilities.

**Page Location:** When you click "Feedback" in the navigation menu

### What Can Be Sorted?

Users can choose how to view feedback:

A screenshot of a feedback page interface. At the top left is a magnifying glass icon followed by the text "Search Reviews". To its right is a "Sort by:" dropdown menu with "Most Recent" selected. Below these are two input fields: one for "Type here..." and another for "Most Recent". Underneath these fields is a section titled "Options:" containing two bullet points: "Most Recent (newest reviews **first**)" and "Most Relevant (most helpful **first**)".

### How It Works

Option 1: Most Recent (Default)

- Shows the newest feedback first

- Older reviews appear at the bottom
- **Example:** A review from today appears before a review from last week

### Option 2: Most Relevant

- Shows feedback with the most "helpful" votes first
- Members can mark reviews as helpful or not helpful
- Reviews with more "Helpful" votes appear at the top

## Behind the Scenes

**Where the sorting happens:** Server (Database)

**The sorting code:**

```
// If user selects "Most Relevant"
if ($sort_by === 'relevant') {
    // Sort by helpful votes (highest first), then by date
    ORDER BY helpful_count DESC, date DESC
}
// If user selects "Most Recent" (or default)
else {
    // Sort by date only (newest first)
    ORDER BY date DESC
}
```

**What this means:**

- `DESC` = Descending (high to low, or newest to oldest)
- `helpful_count` = Number of "helpful" votes
- `date` = When the feedback was posted

## Real Example

**Before Sorting:**

```
Feedback #1: "Great gym!" - Posted Nov 5 - 2 helpful votes
Feedback #2: "Love it!" - Posted Nov 7 - 5 helpful votes
Feedback #3: "Good experience" - Posted Nov 6 - 1 helpful vote
```

**After "Most Recent" Sort:**

```
Feedback #2: "Love it!" - Posted Nov 7 - 5 helpful votes ← Newest
Feedback #3: "Good experience" - Posted Nov 6 - 1 helpful vote
Feedback #1: "Great gym!" - Posted Nov 5 - 2 helpful votes
```

**After "Most Relevant" Sort:**

```
Feedback #2: "Love it!" - 5 helpful votes ← Most helpful
Feedback #1: "Great gym!" - 2 helpful votes
Feedback #3: "Good experience" - 1 helpful vote
```

**Code Location:** [/public/php/feedback.php](#) (lines 38-105)

# Reservations Page Sorting

## What This Page Does

The **Reservations page** shows your booked training sessions. Members can:

- View upcoming classes they've registered for
- See past completed sessions
- Book new training sessions with coaches

**Page Location:** When you click "Reservations" in the navigation menu

## What Can Be Sorted?

In the "My Booked Sessions" section, you can sort your bookings by:

MY BOOKED SESSIONS				
Date [↑]	Time [↑]	Class	Coach	Status
Click ↑	Click ↑			
to sort	to sort			

## How It Works

### Sort Button 1: Date

**What it does:** Arranges bookings by their calendar date

**Click once:** Earliest dates first (↑ ascending)

```
Nov 10 ← First  
Nov 15  
Nov 20 ← Last
```

**Click again:** Latest dates first (↓ descending)

```
Nov 20 ← First  
Nov 15  
Nov 10 ← Last
```

### Sort Button 2: Time

**What it does:** Arranges bookings by time of day (ignoring the date)

**Click once:** Morning sessions first (↑ ascending)

```
09:00 AM ← First  
11:30 AM  
02:00 PM  
05:30 PM ← Last
```

**Click again:** Evening sessions first (↓ descending)

```
05:30 PM ← First
02:00 PM
11:30 AM
09:00 AM ← Last
```

## Behind the Scenes

**Where the sorting happens:** Browser (JavaScript)

**The sorting code:**

```
// Sort by DATE
sortedBookings.sort((a, b) => {
    const dateA = new Date(a.datetime); // Convert to date object
    const dateB = new Date(b.datetime);
    return dateA - dateB; // Compare dates
});

// Sort by TIME ONLY
sortedBookings.sort((a, b) => {
    // Extract hours and minutes
    const timeOnlyA = timeA.getHours() * 60 + timeA.getMinutes();
    const timeOnlyB = timeB.getHours() * 60 + timeB.getMinutes();
    return timeOnlyA - timeOnlyB; // Compare times
});
```

**What this means:**

- The code compares two bookings at a time (a and b)
- For dates: compares full date and time
- For times: converts time to minutes since midnight and compares

## Real Example

**Your Bookings (unsorted):**

```
1. Boxing - Nov 15, 2:00 PM
2. MMA - Nov 10, 9:00 AM
3. Muay Thai - Nov 20, 11:00 AM
4. Boxing - Nov 10, 5:00 PM
```

**After "Sort by Date" (ascending):**

```
1. MMA - Nov 10, 9:00 AM ← Earliest date
2. Boxing - Nov 10, 5:00 PM ← Same date, but later
3. Boxing - Nov 15, 2:00 PM
4. Muay Thai - Nov 20, 11:00 AM ← Latest date
```

**After "Sort by Time" (ascending):**

```
1. MMA - Nov 10, 9:00 AM ← Earliest time (9 AM)
2. Muay Thai - Nov 20, 11:00 AM ← Second (11 AM)
3. Boxing - Nov 15, 2:00 PM ← Third (2 PM)
4. Boxing - Nov 10, 5:00 PM ← Latest time (5 PM)
```

*Notice: When sorting by time, the dates don't matter - only the clock time!*

**Code Location:** /public/js/reservations.js (lines 387-407)

## Equipment Page Filtering

### What This Page Does

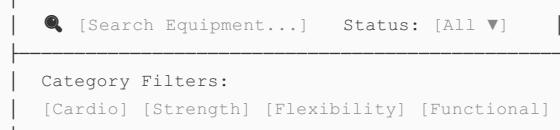
The **Equipment page** shows all the gym equipment available for members to use, including:

- Cardio machines (treadmills, bikes)
- Strength equipment (dumbbells, barbells)
- Flexibility tools (yoga mats, resistance bands)

**Page Location:** When you click "Equipment" in the navigation menu

### What Can Be Filtered?

**Note:** This page uses **filtering** (showing/hiding items) rather than **sorting** (reordering items)



A screenshot of a web-based equipment filtering interface. At the top, there is a search bar with a magnifying glass icon and the placeholder text "[Search Equipment...]" followed by a status dropdown labeled "Status: [All ▾]". Below the search bar is a section titled "Category Filters:" containing four chips: "[Cardio]", "[Strength]", "[Flexibility]", and "[Functional]".

### Filter Options

#### 1. Search Box

Type keywords to find specific equipment:

- Example: Type "treadmill" to see only treadmills
- Searches in: equipment name, description, and category

#### 2. Status Filter

Show equipment by availability:

- **All** - Show everything
- **Available** - Ready to use right now
- **In Use** - Currently being used by someone
- **Under Maintenance** - Being repaired
- **Out of Order** - Not working

#### 3. Category Chips

Click a category to show only that type:

- **Cardio** - Running, cycling machines
- **Strength** - Weights, resistance equipment

- **Flexibility** - Stretching and yoga tools
- **Functional** - Multi-purpose training equipment

## How It Works

**Where the filtering happens:** Browser (JavaScript)

**The filtering code:**

```
// Check each equipment item one by one
const filtered = EQUIPMENT_DATA.filter(item => {

    // Filter 1: Check if category matches (if selected)
    if (activeCategory) {
        if (!item.category.includes(activeCategory)) {
            return false; // Hide this item
        }
    }

    // Filter 2: Check if status matches (if selected)
    if (status !== 'all') {
        if (item.status !== status) {
            return false; // Hide this item
        }
    }

    // Filter 3: Check if search text matches (if typed)
    if (searchText) {
        const searchIn = item.name + item.description + item.category;
        if (!searchIn.includes(searchText)) {
            return false; // Hide this item
        }
    }

    return true; // Show this item (passed all filters)
});


```

**What this means:**

- Each filter acts as a "checkpoint"
- Equipment must pass ALL active filters to be shown
- If any filter fails, the equipment is hidden

## Real Example

**All Equipment (before filtering):**

1. Treadmill - Cardio - Available
2. Dumbbells - Strength - In Use
3. Yoga Mat - Flexibility - Available
4. Stationary Bike - Cardio - Under Maintenance
5. Kettlebell - Strength - Available

**Filter: Category = "Cardio"**

1. Treadmill - Cardio - Available	← Shown
2. Dumbbells - Strength - In Use	← Hidden
3. Yoga Mat - Flexibility - Available	← Hidden
4. Stationary Bike - Cardio - Under Maintenance	← Shown
5. Kettlebell - Strength - Available	← Hidden

#### Filter: Category = "Cardio" + Status = "Available"

1. Treadmill - Cardio - Available	← Shown (matches both!)
2. Dumbbells - Strength - In Use	← Hidden
3. Yoga Mat - Flexibility - Available	← Hidden
4. Stationary Bike - Cardio - Under Maintenance	← Hidden (wrong status)
5. Kettlebell - Strength - Available	← Hidden

#### Filter: Search = "bell"

1. Treadmill - Cardio - Available	← Hidden
2. Dumbbells - Strength - In Use	← Shown (contains "bell")
3. Yoga Mat - Flexibility - Available	← Hidden
4. Stationary Bike - Cardio - Under Maintenance	← Hidden
5. Kettlebell - Strength - Available	← Shown (contains "bell")

**Code Location:** [/public/js/equipment.js](#) (lines 60-93)

## Admin Subscriptions Sorting

### What This Page Does

The **Admin Subscriptions page** is used by gym administrators to:

- View membership requests from new customers
- Approve or reject subscription applications
- Track active members

**Page Location:** Admin dashboard → Subscriptions

**Who can access:** Only gym administrators

### What Gets Sorted?

When administrators view subscription requests, they are automatically sorted by submission date.

PENDING SUBSCRIPTION REQUESTS				
Name	Plan	Submitted		
John Doe	Gladiator	Nov 5, 2025 10:30 AM	← Oldest	
Jane Smith	Champion	Nov 6, 2025 2:15 PM		
Mike Jones	Brawler	Nov 7, 2025 9:00 AM	← Newest	

### How It Works

**Sorting rule:** Always shows oldest requests first

**Why?** Administrators should handle older requests before newer ones (first-come, first-served)

**Where the sorting happens:** Browser (JavaScript)

**The sorting code:**

```
// Convert date strings to timestamps and sort
data.sort((a, b) => {
    const dateA = Date.parse(a.date_submitted); // Convert to number
    const dateB = Date.parse(b.date_submitted);
    return dateA - dateB; // Oldest first (ascending)
});
```

**What this means:**

- `Date.parse()` converts "Nov 5, 2025 10:30 AM" to a number (milliseconds since 1970)
- Smaller numbers = older dates
- Result: Oldest submissions appear at the top

## Real Example

**Submissions (as received):**

```
Request 1: Submitted Nov 7, 2025 at 9:00 AM
Request 2: Submitted Nov 5, 2025 at 10:30 AM
Request 3: Submitted Nov 6, 2025 at 2:15 PM
```

**After Automatic Sorting (oldest first):**

```
Request 2: Submitted Nov 5, 2025 at 10:30 AM ← First (oldest)
Request 3: Submitted Nov 6, 2025 at 2:15 PM
Request 1: Submitted Nov 7, 2025 at 9:00 AM ← Last (newest)
```

**Code Location:** `/public/php/admin/js/subscriptions.js` (line 19)

## 📊 Summary Table

Here's a quick reference of all sorting features:

Page	What Gets Sorted	How to Sort	Where It Happens
<b>Feedback</b>	Customer reviews	Dropdown menu: "Recent" or "Relevant"	Server (Database)
<b>Reservations</b>	Your bookings	Click "Date ↑" or "Time ↑" buttons	Browser (JavaScript)
<b>Equipment</b>	Gym equipment	Select filters (Category, Status, Search)	Browser (JavaScript)
<b>Admin Panel</b>	Membership requests	Automatic (oldest first)	Browser (JavaScript)

## Quick Comparison

Feature	Server-Side Sorting	Client-Side Sorting
<b>Speed</b>	Fast for large data	Instant for small data
<b>When</b>	Page loads	Click a button
<b>Reload page?</b>	Sometimes	Never
<b>Used in</b>	Feedback (initial load)	Reservations, Equipment

## Technical Details

*This section is for developers who want to understand the implementation.*

### Technologies Used

#### Frontend (Browser):

- **JavaScript** - Programming language for interactive features
- **Array.sort()** - Built-in JavaScript sorting method (uses Timsort algorithm)
- **Array.filter()** - Built-in JavaScript filtering method

#### Backend (Server):

- **PHP** - Server-side programming language
- **MySQL** - Database management system
- **SQL ORDER BY** - Database sorting command

### Algorithm Performance

All sorting methods used in this application are efficient:

Method	Type	Performance	Explanation
JavaScript <code>.sort()</code>	Timsort	$O(n \log n)$	Very efficient - handles 1000 items in ~10,000 operations
SQL <code>ORDER BY</code>	Database	$O(n \log n)$	Optimized by database engine with indexes
JavaScript <code>.filter()</code>	Linear scan	$O(n)$	Fast - checks each item once

#### What this means in plain English:

- **$O(n)$**  - Time increases linearly (100 items = 100 units of time)
- **$O(n \log n)$**  - Very efficient for most data sizes (100 items ≈ 664 units of time)
- All these are considered "fast" algorithms for web applications

### Code File Locations

For developers who want to view or modify the sorting code:

Feature	File Path	Key Lines
Feedback sorting (PHP)	/public/php/feedback.php	38-105
Feedback sorting (JS)	/public/js/feedback.js	24-42
Reservations sorting	/public/js/reservations.js	387-407
Equipment filtering	/public/js/equipment.js	60-93
Admin subscriptions	/public/php/admin/js/subscriptions.js	18-19

## Other Sorting in the System

The application also uses sorting in other areas (handled by SQL):

Location	What's Sorted	How
Contact page	Contact messages	By date submitted (newest first)
Logged-in homepage	Membership requests	By date submitted (newest first)
Logged-in homepage	Upcoming bookings	By date and time slot (earliest first)
Logged-in homepage	Popular classes	By booking count (most popular first)
Daily bookings API	Training sessions	By start time
Trainer bookings API	Reservations	By reservation date

## ⌚ Key Takeaways

### For Users:

1. **Feedback Page** - Choose between "Most Recent" or "Most Relevant" sorting
2. **Reservations Page** - Click Date or Time buttons to sort your bookings
3. **Equipment Page** - Use filters to find equipment by category, status, or search
4. **Sorting is Fast** - All sorting happens quickly, whether on server or in browser

### For Developers:

1. **Two Approaches** - Server-side (SQL) for initial loads, client-side (JS) for interactions
2. **Efficient Algorithms** - All use  $O(n \log n)$  or better performance
3. **User Experience** - Interactive sorting provides instant feedback
4. **Code Organization** - Sorting logic is clearly separated and documented

## ❓ Frequently Asked Questions

**Q: Why doesn't the equipment page have a "sort" button?**

A: It uses filtering instead, which hides items that don't match your criteria rather than reordering them.

**Q: Can I sort by multiple criteria at once?**

A: Currently, each page supports one sort option at a time. The feedback page does sort by helpful votes AND date when "Most Relevant" is selected.

**Q: What happens to my sort preference when I refresh the page?**

A: Currently, sort preferences reset to default. Future versions may remember your preference.

**Q: Why do some pages sort on the server and others in the browser?**

A: Server sorting is used when loading data initially (efficient for large datasets). Browser sorting is used for interactive features where you click buttons to re-sort (instant response).

**Q: How fast is the sorting?**

A: Very fast! Even with hundreds of records, sorting happens in milliseconds.

---

## For New Developers

If you're new to the codebase and want to understand sorting:

### Start Here:

1. **Read this document** - You're already doing it! ✓
2. **Look at the Feedback page code** - Simplest example ( [/public/php/feedback.php](#) )
3. **Try the Reservations page** - See client-side sorting in action
4. **Open the browser console** - See sorting happen in real-time (Press F12)

### Key Concepts:

- **Ascending** = Low to high (1, 2, 3) or old to new
- **Descending** = High to low (3, 2, 1) or new to old
- **Filter** = Show/hide items based on criteria
- **Sort** = Reorder items based on criteria
- **Stable Sort** = Items with equal values keep their original order

### Debugging Tips:

```
// Add this to see what's being sorted
console.log('Before sort:', data);
data.sort(sortFunction);
console.log('After sort:', data);
```

### Common Sorting Patterns:

#### Ascending numbers:

```
numbers.sort((a, b) => a - b);
```

#### Descending numbers:

```
numbers.sort((a, b) => b - a);
```

#### Ascending strings:

```
strings.sort((a, b) => a.localeCompare(b));
```

#### Ascending dates:

```
dates.sort((a, b) => new Date(a) - new Date(b));
```

## Future Improvements

Potential enhancements to the sorting system:

1. **Remember sort preferences** - Save user's preferred sort order in localStorage
2. **Multi-column sorting** - Sort by multiple criteria (e.g., date then time)
3. **Custom sort orders** - Allow users to drag and drop to reorder
4. **Sort animations** - Visual feedback when items reorder
5. **Advanced filters** - Date ranges, multiple category selection
6. **Sort performance metrics** - Display how many items were sorted
7. **Export sorted data** - Download filtered/sorted results as CSV

---

#### Document End

*Last updated: November 7, 2025*

*For questions or updates about sorting algorithms, contact the development team.*

*File location: [/docs/sorting-algorithms.md](#)*