CoCaTr

T2A2 - Two Sided Rails Marketplace
Harley Martin-Farrell

What is CoCaTr?

Coach - Carry - Trade

A central hub for people to arrange in-game trades and services with the added confidence that user reviews and moderation provide.

The CoCaTr Experience

- Easily see how many successful exchanges a user has done next to their name

- Users with completed exchanges can post reviews

- Built in report system to help admins catch repeat scammers and inappropriate listings

- A listing can be about services in multiple games

- Coaching section is open to anyone offering their coaching services

- Get in on the beginning of the next big thing

The Social Media Experience

- Often no way to see if the user is a reputed seller, coach or trader

- No idea if they've completed any successful trades or exchanges

- Reporting often requires talking to a mod directly, who may or may not take action.

- Communities are split by game, requiring user to join multiple groups/servers

- Advertising is often frowned upon, or even not allowed at all

- You already missed buying shares in discord
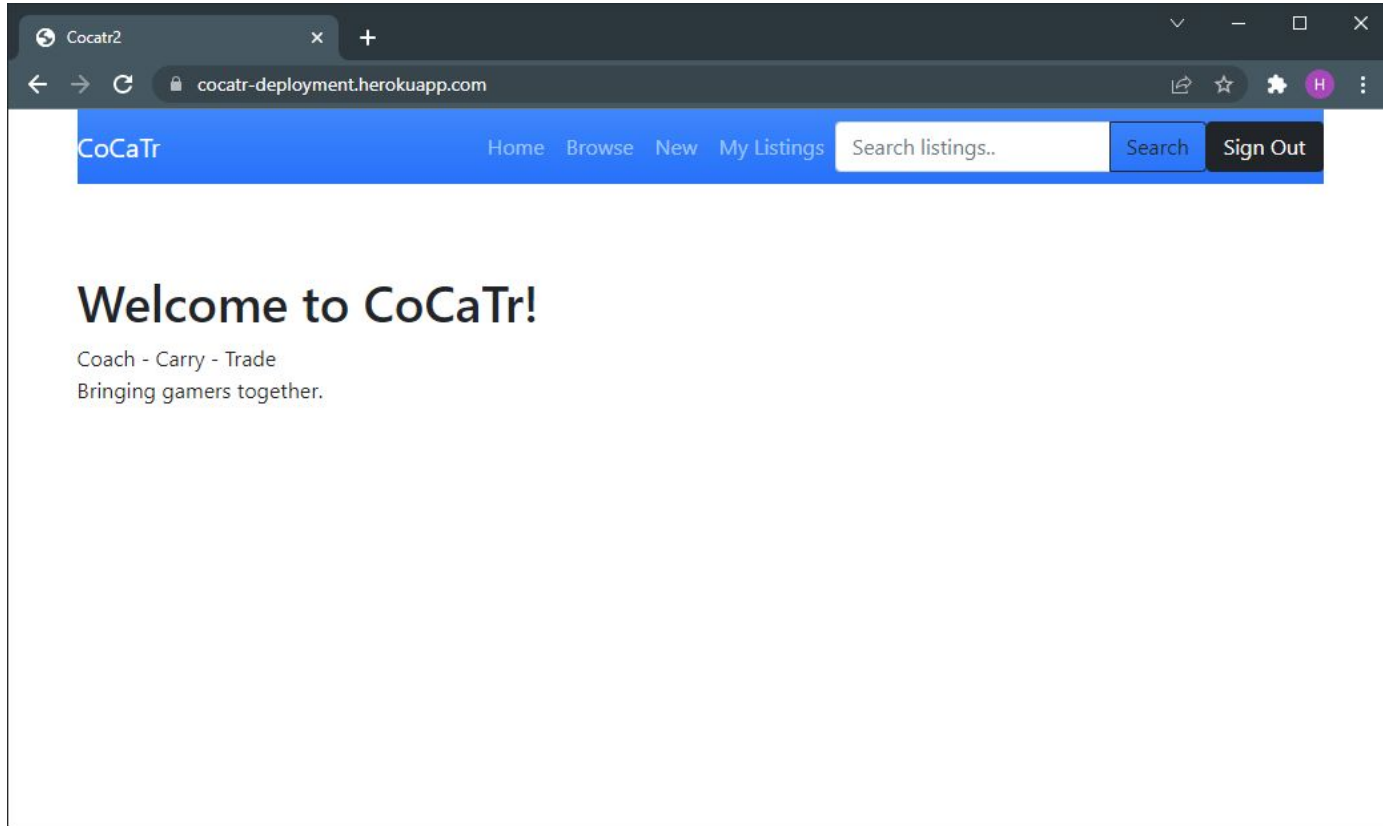
Home # Page - Work in Progress
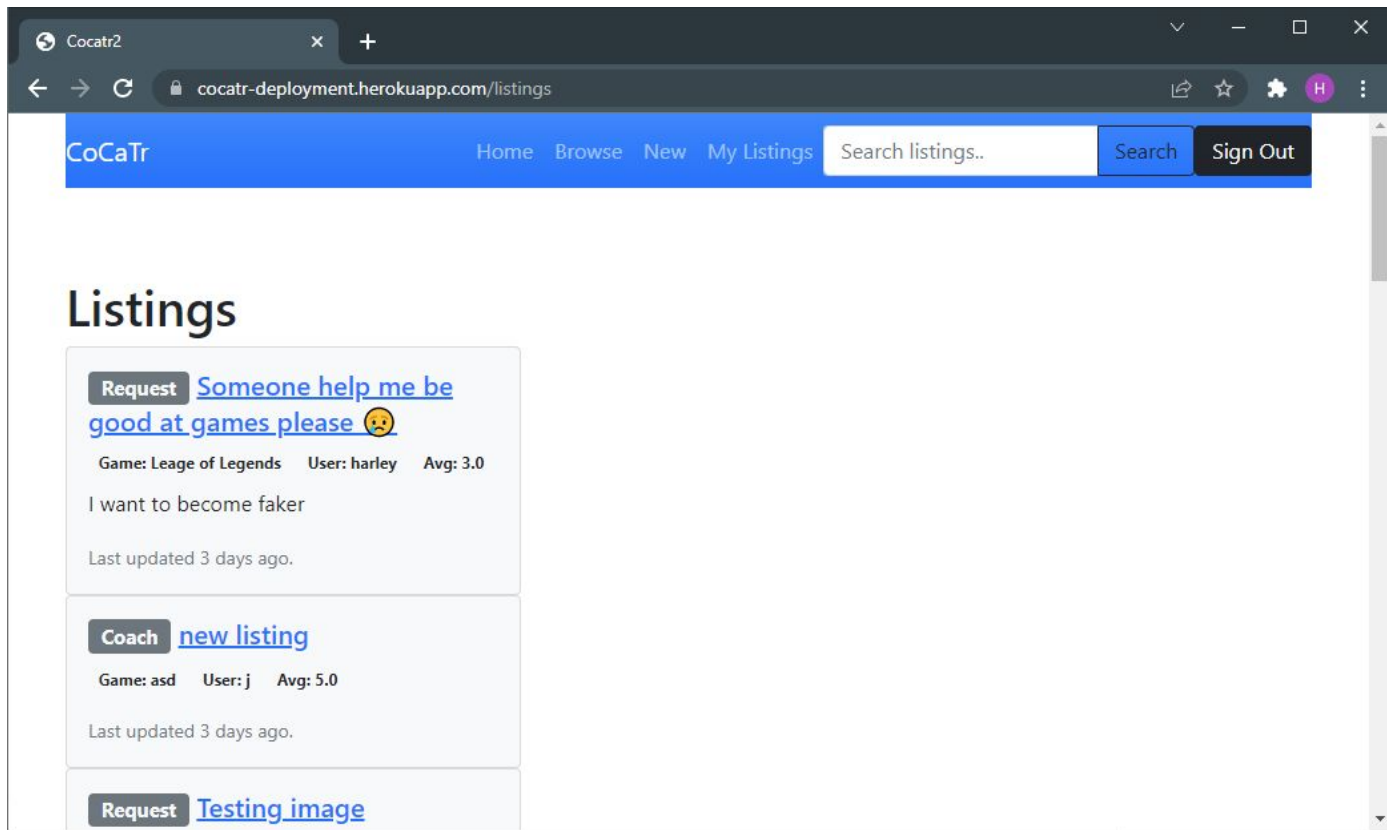
Listings # Index - Browse

Owners of Listings can see additional info on Listing#Index

Yellow [!] alerts of pending applications, unsure if the number of pending applications would be better???

Closed Listings are not visible on Listings # Browse, but are available to the owner under "My Listings" and can be reopened.

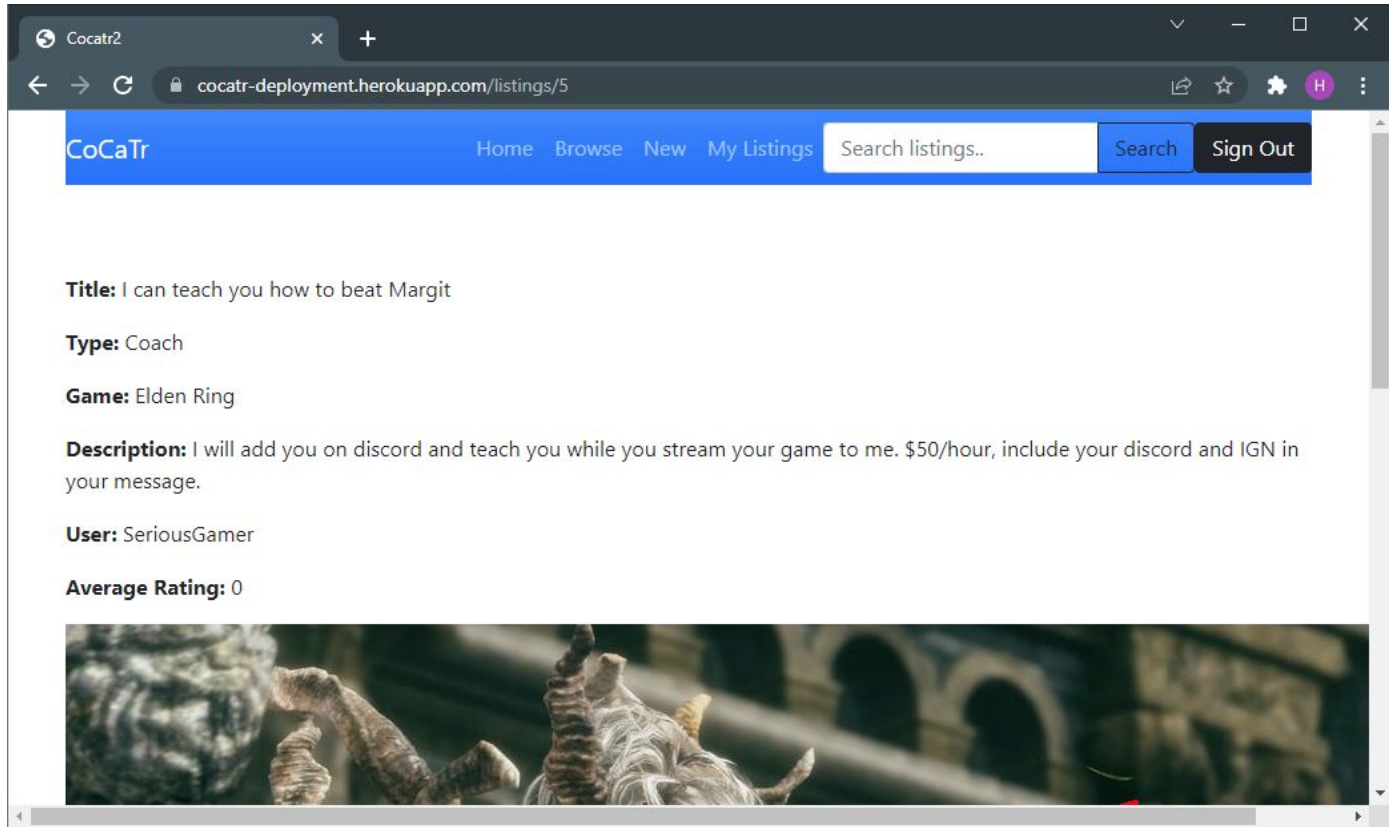# Listing Card - In Depth

Listing # Show - Work in Progress
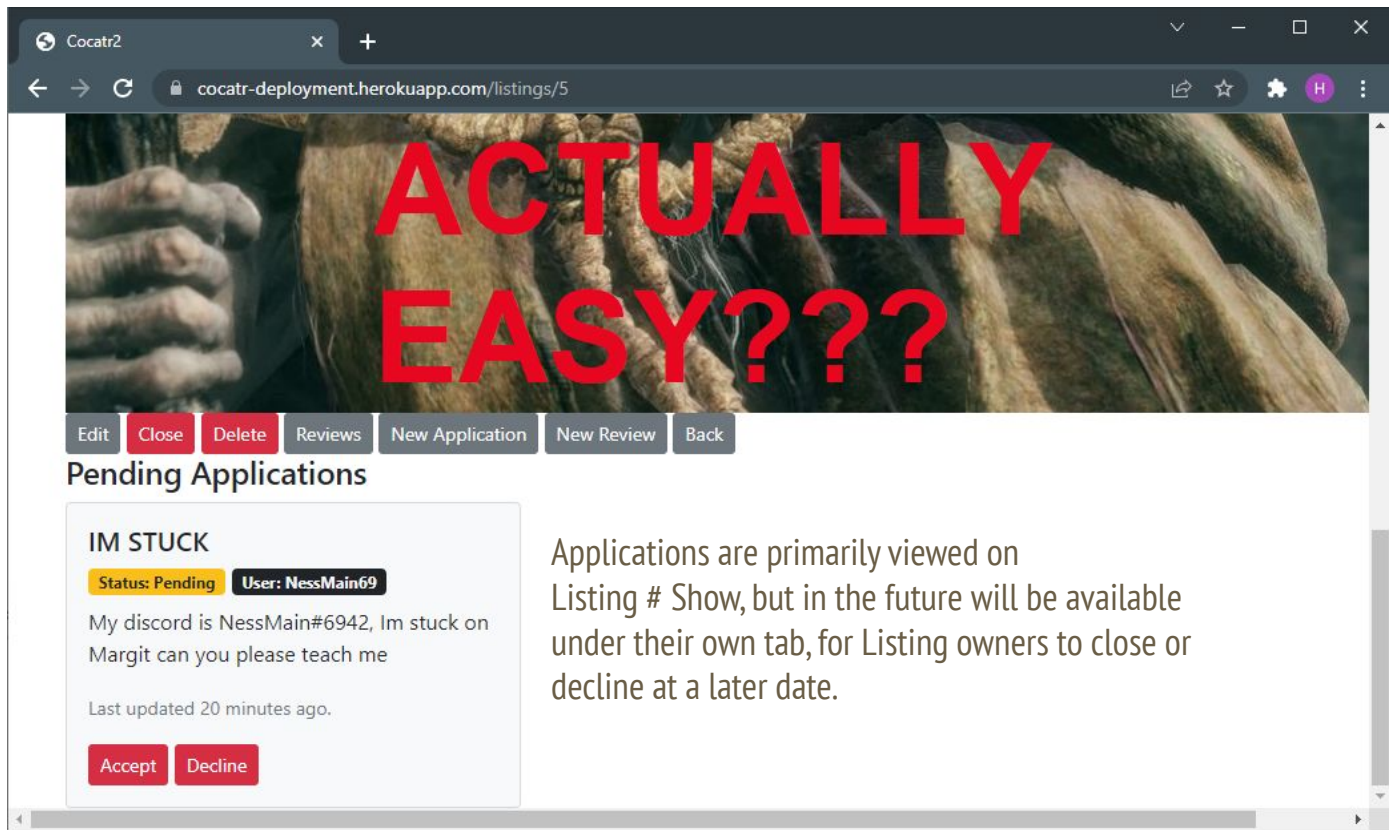
Listing # Show - Pending Applications

```
app > models > 🦀 listing.rb
1    class Listing < ApplicationRecord
2    include Filterable
3    include PgSearch::Model
4      belongs_to :user
5      has_many :reviews
6      has_many :applications
7      has_one_attached :listing_picture
8
9      # Enums define what the integer values mean, and provide an alternate way to use them
10     enum listing_type: { "Coach" => 0, "Offer" => 1, "Request" => 2, "Trade" => 3 }
11     enum status: { "Open" => 0, "Closed" => 1, "Archived" => 2, "Removed" => 3}
12
13     # pg_search scope to be a bit more flexible for the search bar
14     pg_search_scope :filter_by_title, against: :title
15
16     # Regular scopes for assorted views
17     scope :filter_by_status, -> (status) {where status: status}
18     scope :filter_by_listing_type, -> (listing_type) {where listing_type: listing_type}
19     scope :filter_by_username, -> (username) { joins(:user).where('username like ?', "#{username}")}
20
```

https://www.justinweiss.com/articles/search-and-filter-rails-models-without-bloating-your-controller/

Really Nice Code Stuff - Scopes

```ruby
1   class ListingsController < ApplicationController
2     before_action :set_listing, only: %i[ show edit update destroy update_listing ]
3     before_action :authenticate_user! , only: [:new, :create, :destroy, :my_listings]
4
5     # This modified index method accepts search params
6     def index
7       @listings = Listing.filter_by(params.slice(:title, :username)).filter_by_status("Open")
8     end
9
10    def my_listings
11      @listings = Listing.filter_by_username(current_user.username)
12    end
13
14    # GET /listings/1 or /listings/1.json
15    def show
16      @applications = @listing.applications.filter_by_status("Pending").filter_by_application_type("Applicat:
17    end
18
```

Thanks to the work done in the scopes, limiting what gets shown on each page is really simple.

Really Nice Code Stuff - Scopes

Other Notes

- Simple_form was an absolute trap, write your forms yourself. 2/10

- Pg_search is a god send, every article about writing similar searches makes it look absolutely awful. 10/10

- Putting the cards for displaying Listings and Applications in their own partials was a great idea, I'm a genius. 10/10

- I'm not looking forward to implementing Stripe/payment into a website.