

# Bookspot

# A marketplace app for book lovers



# Scope

## What is it?

A dual-sided marketplace for viewing, buying and selling listed items (in this case, specifically books)

## Who is it for?

Anyone wanting to buy or sell books, or both. More specifically, it provides details and categories for the listing that are distinct to book listings.

## What problem does this solve?

TikTok in particular has helped cultivate a renewed interest in reading, as well as helping determine the popularity of certain titles. This can help provide an avenue for more affordable purchases, contributing to circular economy, provide owners with a platform to easily sell and makes it easier to track stock availability.

# Functions and Features

## User Authentication

Ability to create account, sign into account and sign out. Then able to sign in again in new session.

## Create Listing

Once signed in, a user can create a new listing, including uploading an image and specifying details unique to book listings.

## Purchase Listing

Once signed in, a user can purchase an available listing that is not their own.

## Edit Listing

Once signed in, a user can edit only their own listings to change the details or even delete it.

## View Bought and Sold Orders

Once signed in, a user can view their own bought items and any items of their own that have been sold.

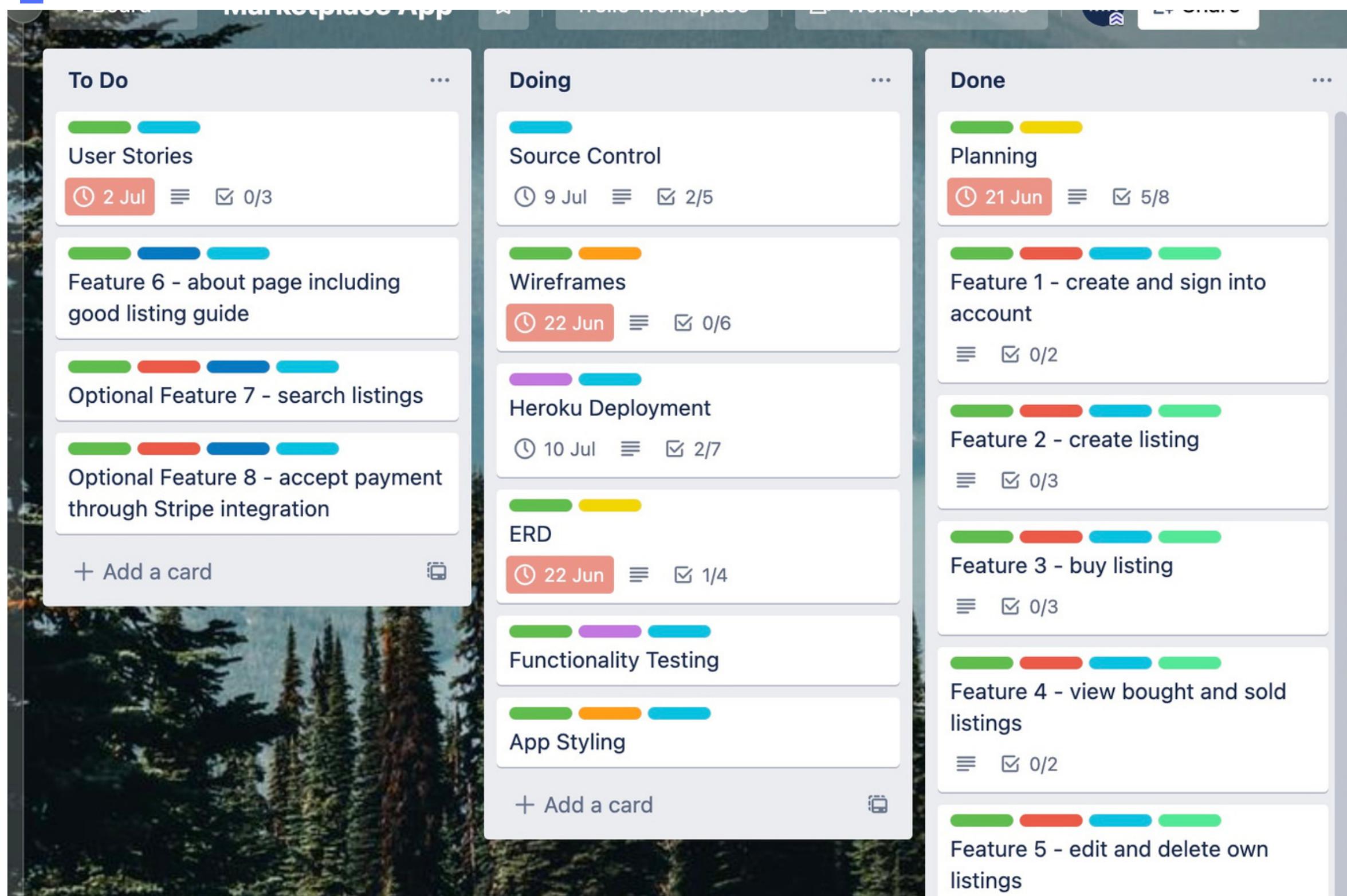
## Access Help Page

An added help page to the app, which will include a link to a guide on how to post a well put together listing, sitemap and the app documentation

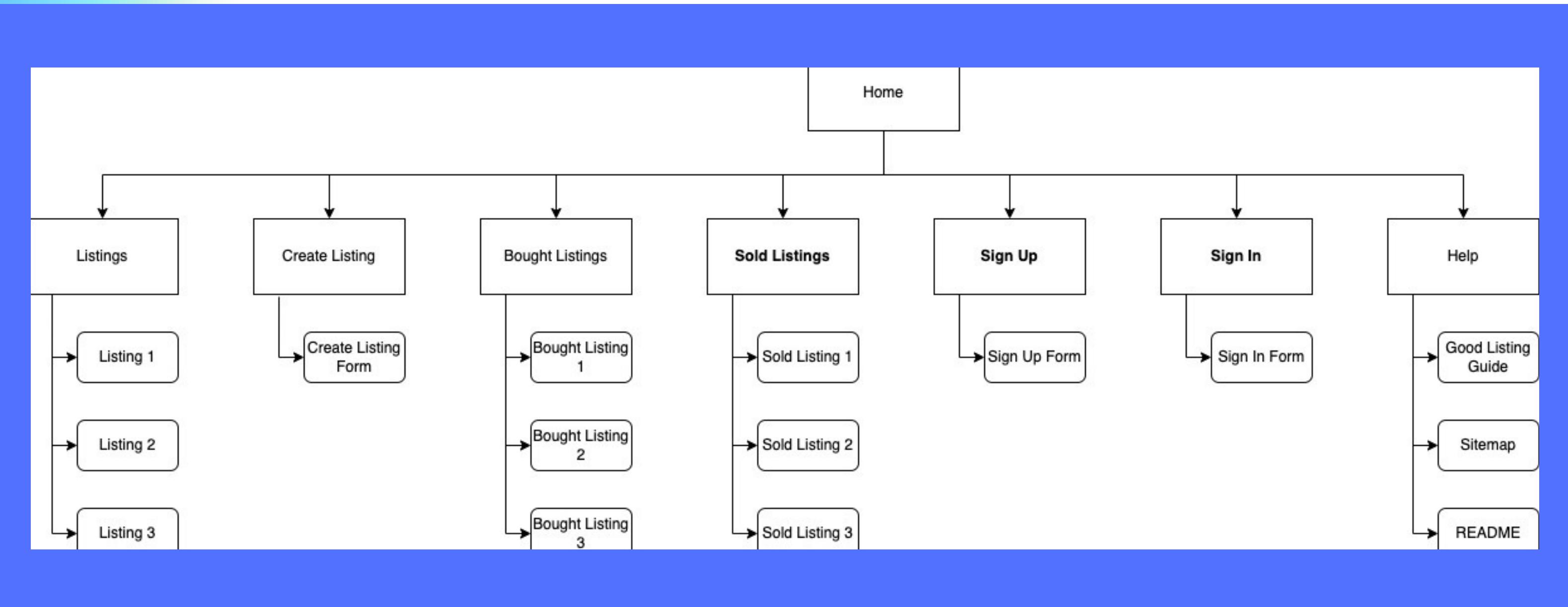
## Optional Search Function and Payment

If time allows, will implement a search function to look for key words in a listing. Payment processing in the form of Stripe will be utilised.

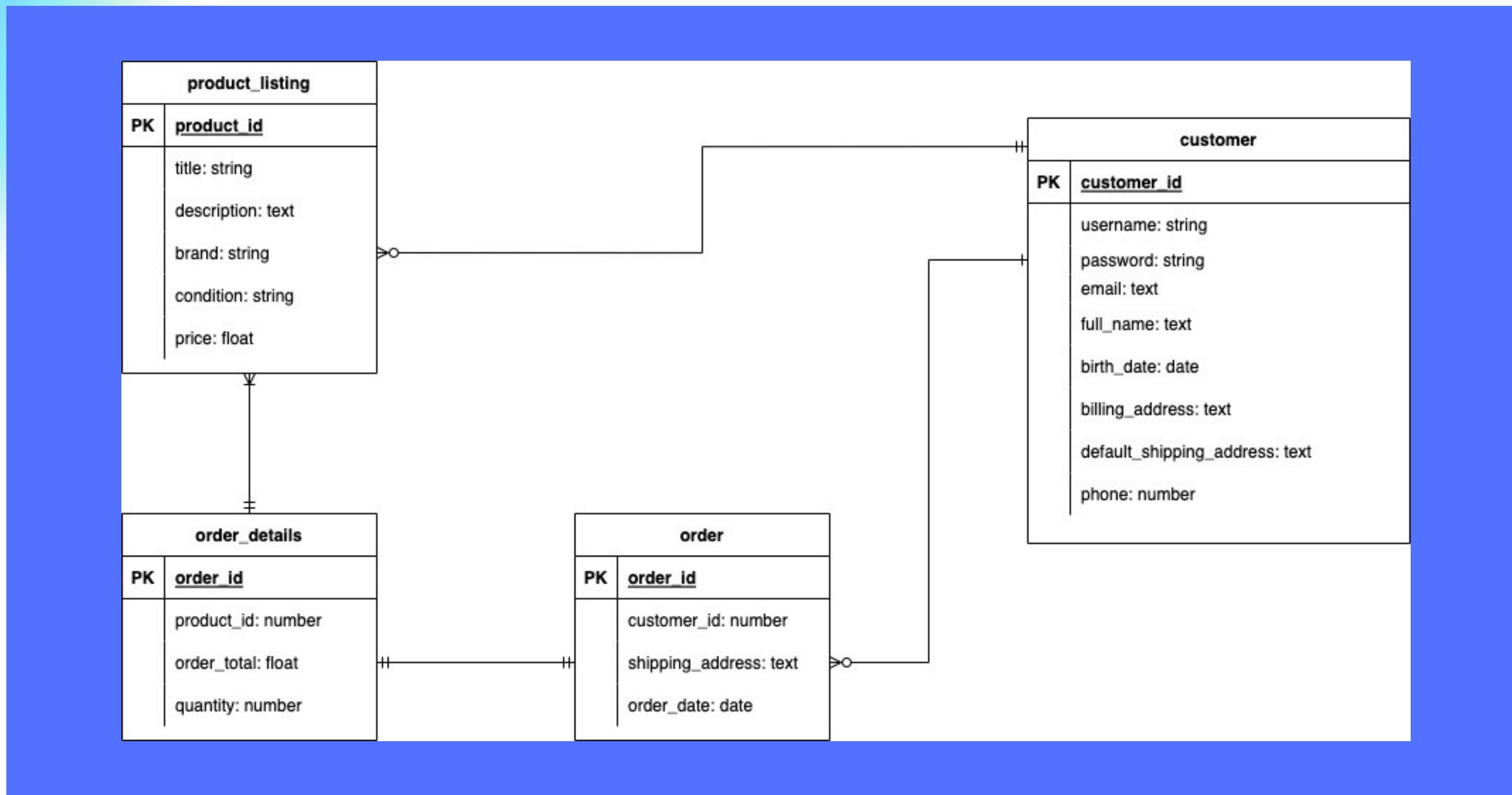
# Implementation Plan



# Sitemap



# ERD



# User Authentication

- Whilst a user can casually browse the listings on the site, they must create an account or log into an existing account to be authorised for additional functionality
- User authentication will be done using Devise
- User will sign up with an email, username and password they provide and identified internally by an unique account id

"As a casual browser, I want to be able to view the listings details without having to create an account."

"As a casual browser, I want to be able to see the sold status of a listing and create an account if I want to buy it."

## Sign up

Email

Username

Password

(6 characters minimum)

Password confirmation

SIGN UP

[Log in](#)

# Create Listing

- Only a signed in user can create a listing for their items.
- The listing will be created using a form and each field given an appropriate input data type.
- There will be the ability to upload an image of the item.

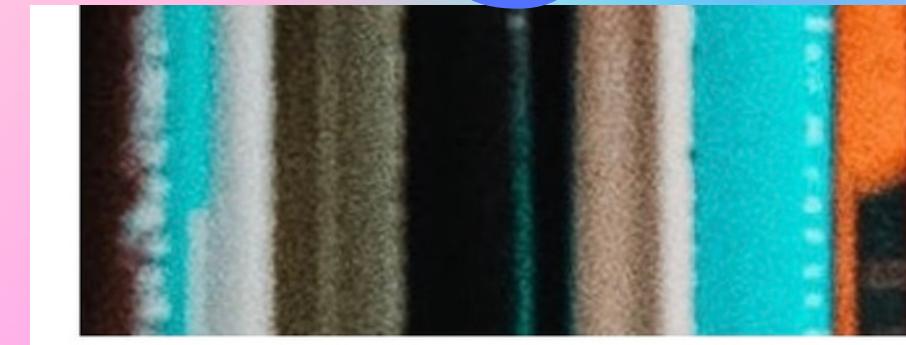
"As a seller, I want to be able to create a new listing and specify the details of the product."

The screenshot shows a user interface for creating a new listing. At the top, there is a navigation bar with four buttons: 'LISTINGS', 'BOUGHT ITEMS', 'SOLD ITEMS', and 'SIGN OUT'. Below the navigation bar, the title 'New Listing' is displayed. The form consists of several input fields: 'Title' (a single-line text input), 'Description' (a multi-line text area with a character count indicator '1000'), 'Tattered And Worn' (radio button), 'Very Loved' (radio button), 'A Little Loved' (radio button), 'New And Shiny' (radio button), 'Price' (a single-line text input), 'Category' (a dropdown menu currently set to 'Biographies'), 'Picture' (a file upload input showing 'No file chosen'), and a 'CREATE LISTING' button at the bottom. A 'Back' link is located at the very bottom of the page.

# Purchase Listing

- Only a signed in user can purchase an item
- They will only be authorised to purchase if the item is identified as being listed by another user
- An item will only be available to purchase once, after it has been purchased there will be no button to purchase and a message will be displayed as such
- Optional additional function of taking actual payment using Stripe

"As a buyer, I want to be able to easily create an account to purchase items."



PURCHASE

**Title:** Book Listing

**Description:**

This is a good book available to sell

**Condition:** new\_and\_shiny

**Price:** 10

**Sold:** false

**Account:** 1

**Category:** 7

# Edit Listing

- Only a signed in user can edit a listing
- Users can only edit their own listings
- A listing can be edited in any field
- Listings can also be deleted by their author

"As a seller, I want to be able to sign into my account and edit my listings."

The screenshot shows a user interface for managing item listings. At the top, there is a navigation bar with links for 'LISTINGS', 'BOUGHT ITEMS', 'SOLD ITEMS', and 'SIGN OUT'. The main area is titled 'Editing Listing'. It contains several input fields and dropdown menus:

- Title:** Book Listing
- Description:** This is a good book available to sell
- Tattered And Worn:** A set of radio buttons with 'Very Loved' selected.
- New And Shiny:** A radio button that is selected.
- Price:** An input field containing the value 10.
- Category:** A dropdown menu currently set to 'General Fiction'.
- Picture:** A section with a 'Choose file' button and a message indicating 'No file chosen'.
- UPDATE LISTING:** A button at the bottom right of the form.

# View Bought and Sold Items

- Only a signed in user can view bought and sold items
- The user will only be able to see items that they have purchased themselves
- The user will only be able to see items that they listed for

"As a buyer, I want to be able to sign into my account and view my bought items."

The screenshot shows a user interface for managing items. At the top, there is a navigation bar with three tabs: 'LISTINGS' (with a house icon), 'BOUGHT ITEMS' (selected), and 'SOLD ITEMS'. To the right of the tabs is a 'SIGN OUT' button. Below the navigation bar, the title 'Purchased Items' is displayed. A table follows, listing a single item: 'Book' with the description 'Hello this is a book', 'Condition' 'very\_loved', 'Price' '\$860,868.00', 'Seller' 'Morgan', and 'Category' 'General Fiction'.

Title	Description	Condition	Price	Seller	Category
Book	Hello this is a book	very_loved	\$860,868.00	Morgan	General Fiction

The screenshot shows a user interface for managing items. At the top, there is a navigation bar with three tabs: 'LISTINGS' (with a house icon), 'BOUGHT ITEMS', and 'SOLD ITEMS' (selected). To the right of the tabs is a 'SIGN OUT' button. Below the navigation bar, the title 'Sold Items' is displayed. A table follows, listing a single item: 'Book' with the description 'Hello this is a book', 'Condition' 'very\_loved', 'Price' '\$860,868.00', 'Seller' 'Morgan2', and 'Category' 'General Fiction'.

Title	Description	Condition	Price	Seller	Category
Book	Hello this is a book	very_loved	\$860,868.00	Morgan2	General Fiction

# Additional Functions

## Access Help Page

A help page in the app that can be accessed by all, featuring a guide on how to create an effective listing, a copy of the sitemap and link to the app documentation.

## Optional Search Function

If time allows, will introduce a function to search listed items by a key term or terms.

## Optional Payment Processing

If time allows, will introduce the ability to properly process payment via Stripe.

# Gems and Third Party Services

## Devise Gem

Installed and used to handle the authentication functions of the app, primarily sign up, sign in and sign out.

## RSPEC

Installed and used run tests on the application code.

## Heroku

Hosting platform for the application.

## Ransack

Installed and used to help implement the search bar function

## Stripe

If time allows, will introduce the ability to properly process payment via Stripe.

# Challenges and Next Steps

## The Blank Page Complex

Actually taking it from planning to the code writing stage.

## Correctly implementing the decided Git workflow

Whilst implementing my Git workflow format, I kept forgetting to create a new branch for each feature or function, so some of those changes were implemented directly on the development branch.

## UI and Styling

This will be the primary next step, applying the styling and design to the app.

## Testing

Continue to run tests for functionality and deploy to production to solve any issues at deployment level.

# Thank You

---

Now for a tour of the application in progress