Coding Challenge Workshop

Prep Work:

Getting Started

- **Questions and Clarifications**
- **Tech Stack**

Sketch out Your Solution

- **Design Diagram/Wireframes**
- **Class/Component/File/DB Structure and Flow**
- **Brainstorming Bonus Features and Tests**

Implementation Timeline

- **Research Areas**
- **Possible Hurdles**
- **Order of Operations**

Pseudo-Code

Real Work:

Deliverables

Bonus Features

Comments

Tests!

Notes On Your Choices/Process/Bugs

Refactoring

Readme

Choose one of the following two coding challenges and complete the prep-work for that coding challenge!

Shop Support API

High level

There are times when shops want to sell fonts that require a license key to use. One pain point that exists: they have to manually message customers through Creative Market after every purchase to provide when with a license key.
We want to empower shops to automatically send a license key to a customer without exposing customer data (like an email) to the shop. We already send the shop a webhook for every sale that includes some customer information and an order ID, so we'd like to introduce an API endpoint to enable this new workflow feature!

Scope

Write a small API that:

1. Receives a request (e.g. POST /license-key)

```
{
   userID: 123,
   userID_customer: 456,
   licenseKey: "SOME_STRING",
   orderID: 123123
}
```

1. After validating the incoming request, we'll look up the customer's userID, ensure the orderID exists between that customer and that shop, and then email them a license key.
2. For every successful API request, we'd like to store the license key that the shop sent along.
3. We'd also like to increment a count every time this is done (shown in the schema below). Think of it like an aggregation column.

Data

Assume some prior knowledge on this project. Namely, there is a table with some users and a table with order info. Feel free to seed in some dummy data. Schemas might look like:

```
USERS
   userID (PK)
   emailAddress
   numLicenseKeysSent

LICENSE_KEYS
   userID (customer)
   licenseKey

ORDERS
   orderID (PK)
   userID_customer
   userID_shop
```

Minimum Requirements

- Save things to a repo so we can review your work.
- Write in Ruby
- Libraries are welcome, but no frameworks. Sorry, Rails!
- Write your own queries. No ORMs, please.
- This is for our API, so tests are a requirement.
- Be thoughtful about validation and API responses.

Digital Thank You Page

Overview

Many customers send gift crates to their grandchildren, nieces, nephews and friend's kids. Many times those children are in a different location than the gifter. Many gifters don't receive any feedback on how the gifts were used, what projects the children received, and if they were enjoyed.

Digital Thank You feature will provide a mechanism to encourages giftees to send feedback directly to the gifter. We know the gifter and giftee email, we should be able to provide this loop, and enhance the gifting experience.

Goal

1. Give positive feedback to the gifter - which should encourage repeat gifting and subscription renewals
2. Encourage giftees to use their gifts
3. Allow the giftees to put into words (images) how fun the crates are, enhancing their joy

Functional Requirements

- Send 'Give Thank You' email to a giftee 7 days after their gift's shipping date. Email should contain the gifter's and the giftee's names

- Email should contain a specific link that takes the giftee to a 'Give Thank You' page.

- The 'Give Thank You' page allows the giftee to customize the thank you note to make it more personalized
    - Allow giftee to **upload a photo** and/or **choose from pre-design templates**
    - Allow the giftee to write a personalized note
        i. Note should be plain text - and shouldn't allow for any scripting
        ii. Text area should be large enough to write a meaningful note / story
        iii. Note should be prefilled with Childname, gifter email and default thank you message.

        Dear <gifter name>,
        Thank you for my Crate. I had a lot of fun making my project.
        Thanks <child name>

● When the giftee clicks on “Send Thank You” button, an email should be sent to the gifter.

Subject: <child name> sent thank you note to you! Body:
<child name> sent thank you note to you!

<Check it out>

● On clicking the “Check it out” button, gifter should be taken to their “Thank You” note page. It should have the thank you card that was designed by the giftee.

Existing Database Structure:

Customer:
- ID
- First Name
- Last Name
- Create Date
- Email

Order:
- ID
- Customer ID
- Item
- Order Total
- Giftee Email
- Child Name
- Create Date
- Shipping Date
- Payment Info
- Billing Address
- Shipping Address

Deliverables:

1. Create the “Digital Thank You” feature! Use whatever technologies you wish.
2. Remember to store thank you notes in the DB, and any associated data - whatever you think is necessary for the functionality outlined above
3. How could we improve this feature?