YoPrint Laravel Coding Project Spec

This mini project is intended to verify your proficiency in Laravel. We kept the project short and simple to ensure you can do it within a few hours.

> 💡 **AI Use Policy for This Assignment**
>
> AI tools (like ChatGPT, GitHub Copilot, Claude or Cursor) are **allowed and encouraged** — but with a caveat:
>
> > **Use AI as an expert's tool, not a beginner's crutch.**
>
> We expect you to have the skills to complete this task without AI. Use it to speed things up, not to cover up gaps in core understanding. Your submission should reflect *your* competence, with AI assisting — not replacing — your judgment and ability.
>
> **PS: We took the assignment ourselves with Cursor - and it took us 45 minutes**

Overall Project Goal

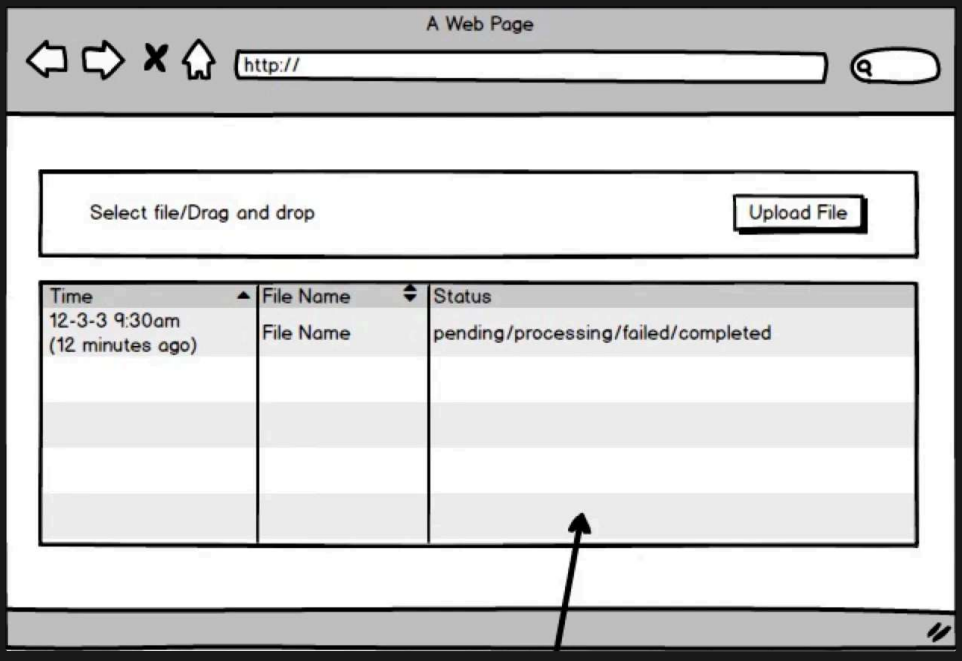
The app does the following. Users will be able to upload CSV files into our system. Once uploaded, we will process the file in the background. We will then notify the user when the process completes. We will also show the user a history of all the file uploads. Here are the detailed specs.

Design Decisions

- You are to come up with your own database schema
- You may use SQLite
- You will need to use background processing. Horizon recommended.
  - You may just use Redis for the queue

UI Design

Here is what the UI should look like.



Requirements

- UI must have a button for users to upload the file.
- UI must have a list of all the recent uploads, including upload time and upload status.
  - You don't have to do any pagination here.
- UI should refresh in real-time the upload status
  - You may choose to poll or use WebSockets. Your choice.
- Bonus point if you used transformers for the endpoint.

CSV Upload / Parsing Requirements

Here is the CSV file format.

Field
UNIQUE_KEY
PRODUCT_TITLE
PRODUCT_DESCRIPTION
STYLE#
SANMAR_MAINFRAME_COLOR
SIZE
COLOR_NAME
PIECE_PRICE

Requirements:

- You must clean up any non-UTF-8 characters
- The file upload must be idempotent.
  - Meaning the user should be able to upload the same file many times without creating duplicate entries.
- The file should also be able to UPSERT the entry.
  - For example, the user can make minor edits to the row (like PIECE_PRICE) and upload the file back into the system.
  - Your code should be able to handle updating the affected rows instead of creating a new row - use UNIQUE_KEY.
  - Test files provided
- Use background processing/workers to process the file upload.
  - I.e., every time a user files upload, you will schedule a job to process that file in the background.

Files Provided

- **Initial import file**

  yoprint_test_import.csv 37815.2KB
- **Upsert file (for subsequent upload)**

  yoprint_test_updated.csv 15.8KB