Nine Kiwis Task Details - Delowar Hossain

Backend Implementation

For this task, I have utilized the TALL stack (Tailwind CSS, Alpine.js, Laravel, and Livewire) along with an SQLite database to facilitate faster development. Since the main focus of the project is handling product information with basic CRUD functionality, I have opted not to implement authentication and authorization mechanisms for this initial version.

The backend encompasses the following core features:

1. **Create Sale Post** – Allows users to create new sale posts.
2. **List Sale Posts** – Displays a list of existing sale posts.
3. **Update Sale Post Status** – Updates the status of a sale post.
4. **API for Pending Sale Posts** – Exposes an API that retrieves all pending sale posts, sorted by their creation date in descending order.

Known Limitations

Due to time constraints, several essential features and best practices have not been implemented in this version, including:

1. Authentication & Authorization
2. Adherence to coding principles and design patterns
3. Localization support
4. Comprehensive validation mechanisms and error messages
5. Full CRUD functionality for product listings (Edit, Show, Delete)

Chrome Extension Implementation

For the Chrome extension, I have utilized the latest manifest version and integrated it with the backend API via the extension's background task. The extension fetches a list of pending sale posts and displays them in a user-friendly interface.

When a user selects a product from the list while on the Facebook Marketplace create item page, the extension automatically fills in the Facebook form with the relevant data from the selected sale post.

Please note that the submission process is not automated; users are prompted to review the data and may edit it before publishing to the marketplace. This method is intentionally designed assuming user's might want to change things on Facebook Marketplace.

Marketplace API Considerations

After reviewing the available Facebook Marketplace API, I determined that it is primarily intended for merchants, requiring a Business Manager account and app review process. As such, I opted not to use this API for the current task. Instead, the extension interacts directly with the Facebook Marketplace form, allowing users to select from pending sale posts and auto-fill the item details, which can then be manually published after review.

Developer Documentation (How to Run the Application)

Backend (Laravel)

Requirements:

- **PHP Version**: **^8.2**
- **PHP Extension**: **sqlite**

To set up and run the backend, follow these steps:

1. Navigate to the *nine-kiwis-backend* directory.
2. Run the following commands in sequence:

```
composer install
cp .env.example .env
php artisan key:generate
php artisan migrate
php artisan storage:link
npm install
npm run build
php artisan serve
```

Chrome Extension Setup

To load the Chrome extension:

1. Open the Chrome browser and navigate to `chrome://extensions`.
2. Enable **Developer Mode** in the top right corner.
3. Click on **Load Unpacked**.
4. Select the **Chrome-Extension** folder.