# Design Choices

1. Frameworks & Libraries:

* Vue.js 3 (CDN): Reactive and lightweight frontend framework.
* Chart.js: Visualizes revenue and order data with minimal config.
* AG Grid: Powerful data grid with filtering, sorting, and inline editing.

2. Page Structure:

* Three pages: Revenue Analysis, Inventory Management, Product Registration.
* Navigation through Vue state binding.

3. Responsiveness & UI:

* CSS Flexbox and media queries for layout.
* Clean UI with soft colors and shadows for usability.

# Challenges & Solutions

* Integration of multiple libraries:

Solved by using CDN links and Vue's lifecycle hooks.

* Real-time UI updates:

Vue's reactivity and watchers ensure auto-updating charts and tables.

* Dynamic data grouping for charts:

Custom logic for grouping sales data by daily, weekly, monthly, annually.

* Image handling:

File Reader API displays image previews before submission.

* Low stock detection:

Computed property identifies and alerts on stock <= 5 items.

# Screenshots & Page Descriptions

1. Revenue Analysis Page:

* Metrics: Total Orders, Total Revenue.
* Filters: Category, Time Range.
* Charts: Orders (line), Revenue (bar).

A screenshot of a computer

AI-generated content may be incorrect.

2. Inventory Management Page:

* Product grid with sorting/filtering/editing.
* Category filter and search.
* Low Inventory alert.

A screenshot of a computer

AI-generated content may be incorrect.

3. Product Registration Page:

* Form: Name, Description, Category, Price, Stock, Image.
* Preview image and confirm submission.
* Success message and real-time UI update.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.