

Hướng dẫn từng bước tạo Quản lý kho nhà hàng

Ứng dụng

Tổng quan Quy trình

Công cụ thiết lập Tạo dự án Tài liệu Phát triển giai đoạn 1 Phát triển giai đoạn 2 Phát triển giai đoạn 3 Triển khai
Phóng

Con trỏ + GitHub .cursorrules Backend Frontend Kiểm tra kết xuất nâng cao
GitHub + Kho lưu trữ + Tài liệu + Cơ sở dữ liệu + Tính năng di động Triển khai + Ra mắt
Kết xuất

BƯỚC 1:CÀI ĐẶT & MÔI TRƯỜNG

1.1 Cài đặtCông cụ cần thiết

Công cụ cần thiết:

đập mạnh

1. Node.js (phiên bản LTS)
<https://nodejs.org/> Tải xuống và [cài đặt](#)

2. Máy tính
<https://git-scm.com/> Tải xuống và [cài đặt](#)

3. IDE con trỏ
<https://cursor.sh/> Tải xuống và [cài đặt](#)

4. Cơ sở dữ liệu (chọn 1 trong 2)
Tùy chọn A: PostgreSQL cục bộ
<https://www.postgresql.org/download/>

Tùy chọn B: Sử dụng cơ sở dữ liệu đám mây (dễ hơn)
<https://neon.tech/> hoặc <https://railway.app/>

Xác minh cài đặt:

đập mạnh

[nút](#) --phiên bản # v18.0.0+
[npm](#) --phiên bản # v9.0.0+
[git](#) --version # v2.0.0+
[psql](#) --version # v14.0.0+ (nếu dùng local)

1.2 Thiết lập kho lưu trữ GitHub

Tạo kho lưu trữ GitHub:

- . Vào [GitHub.com](https://github.com) Đăng nhập tài khoản
- . Nhấp vào "Kho lưu trữ mới"
- . Tên kho lưu trữ:
- . Mô tả: Quản lý hàng tồn kho nhà hàng với OCR - Hỗ trợ tiếng Việt
- . Đặt thành Công khai (hoặc Riêng tư nếu muốn). Thêm README.md
- . Thêm .gitignore chọn "Node"
- . Nhấp vào "Tạo kho lưu trữ"

Sao chép kho lưu trữ:

đập mạnh

```
# Clone về máy git clone
```

```
https://github.com/your-username/restaurant-inventory-system.git cd Restaurant-inventory-system
```

```
# Mở bằng Con trỏ
```

```
con trỏ .
```

1.3 Thiết lập tài khoản Render

- . Vào [Render.com](https://render.com) Đăng ký tài khoản GitHub
- . Kết nối kho lưu trữ GitHub vừa tạo
- . Xác minh email và hoàn tất hồ sơ
- . Lưu ý: Sẽ sử dụng sau khi có code

BƯỚC 2: CREATEPROJECTSTRUCTURE

2.1 Tạo cơ sở dữ liệu thư mục

```
đáp mạnh

# Trong thư mục gốc của dự án, tạo các thư mục:

mkdir -p docs/{yêu cầu, thiết kế, dữ liệu, api, triển khai, thử nghiệm} mkdir

-p frontend/{public, src} mkdir -p

backend/{src, cơ sở dữ liệu, thử nghiệm}

mkdir -p tải lên/{biên lai, bằng chứng, xuất khẩu}

nhật ký mkdir

# Tạo các tập tin cần thiết

chạm .cursorrules

chạm .env.example

chạm vào docker-compose.yml

chạm vào SETUP.md
```

2.2Tập tin Setup.cursorrules

Trong Cursor, tạo file `.cursorrules` :

```
đáp mạnh

# Sao chép toàn bộ nội dung từ tạo tác .cursorrules đã được tạo trước đó

# Dán vào file .cursorrules trong thư mục gốc của dự án
```

2.3 Tạo file tài liệu

Trong thư mục docs/ , tạo các tập tin:

```
đáp mạnh

# Sao chép các tạo phẩm đã tạo:

docs/requirements/BRD.md # Yêu cầu kinh doanh

tài liệu/yêu cầu/TRD.md # Yêu cầu kỹ thuật

docs/requirements/business_rules.md # Quy tắc kinh doanh docs/design/

ui_specs.md # Thông số kỹ thuật UI/UX

docs/dữ liệu/database_schema.sql # Sơ đồ cơ sở dữ liệu

docs/dữ liệu/sample_data.json # Dữ liệu mẫu & Mẫu
```

2.4 Cam kết thiết lập ban đầu

```
đáp mạnh

git add .

git commit -m "Thiết lập dự án ban đầu với tài liệu và cấu trúc" git push origin

main
```

BƯỚC 3: DATABASESETUP

3.1Thiết lập cơ sở dữ liệu (Chọn Tùy chọn A hoặc B)

Tùy chọn A: LocalPostgreSQL

```
đáp mạnh

# Tạo cơ sở dữ
liệu createdb restaurant_inventory

# Kiểm tra kết nối

psql restaurant_inventory -c "CHỌN phiên bản();"
```

Tùy chọn B: Cơ sở dữ liệu đám mây (Khuyến nghị)

Sử dụng Neon.tech (Cấp miễn phí):

- . Vào neon.tech Đăng ký
- . Tạo dự án mới: "Nhà hàng tồn kho"
- . Sao chép chuỗi kết nối: postgresql://username:password@host/database
- . Lưu chuỗi kết nối cho bước sau

3.2 Tạo tệp .env

```
đáp mạnh

# Sao chép .env.example thành .env
cp .env.ví dụ .env

# Chỉnh sửa .env với kết nối cơ sở dữ liệu

NODE_ENV=phát triển

DATABASE_URL=postgresql://tên người dùng:mật khẩu@máy chủ/cơ sở dữ liệu

JWT_SECRET=khóa-bí-mật-của-bạn-ở-đây

GOOGLE_VISION_API_KEY=your-google-api-key # Sẽ thiết lập sau
```

BƯỚC 4: PHÁT TRIỂN BACKEND (Tuần 1-2)

4.1 Khởi tạo Backend

```
đáp mạnh

cd phía sau

npm init -y

# Cài đặt các phụ thuộc với Cursor
```

Dấu nhắc con trỏ:

```
"Thiết lập backend Node.js theo yêu cầu TRD. Cài đặt các phụ thuộc: express, typescript, prisma, joi validation, bcrypt, jsonwebtoken, cors, helmet, morgan. Thiết lập cấu trúc dự án cơ bản theo /docs yêu cầu."
```

4.2 Triển khai DatabaseSchema

Dấu nhắc con trỏ:

```
"Tạo lược đồ Prisma từ /docs/data/database_schema.sql. Bao gồm tất cả các bảng: người dùng, danh mục, mục, hàng tồn kho, giao dịch, đối chiếu. Thêm các mối quan hệ và chỉ mục thích hợp."
```

4.3 Hệ thống xác thực

Dấu nhắc con trỏ:

```
"Triển khai hệ thống xác thực với JWT. Tạo bộ điều khiển, phần mềm trung gian, tuyến đường cho đăng nhập/đăng ký/đăng xuất. Hỗ trợ 4 vai trò: chủ sở hữu, người quản lý, giám sát, nhân viên theo quy tắc kinh doanh."
```

4.4 Các thao tác CRUD cơ bản

Dấu nhắc con trỏ:

```
"Tạo các hoạt động CRUD cho Quản lý mục. Bao gồm bộ điều khiển, dịch vụ, tuyến đường với xác thực phù hợp. Sử dụng dữ liệu mẫu từ /docs/data/sample_data.json để kiểm tra."
```

4.5Kiểm tra phần cuối

```
đáp mạnh

# Bắt đầu phần cuối

npm chạy dev

# Test với Postman hoặc cuộn tròn

http ://localhost:3000/api/health
```

Cam kết tiến độ:

```
đáp mạnh

git add .

git commit -m "Backend: Xác thực + CRUD cơ bản + Thiết lập cơ sở dữ liệu" git push

origin main
```

BƯỚC 5: PHÁT TRIỂN FRONTEND (Tuần 2-3)

5.1 Khởi tạo Frontend

```
đáp mạnh

cd giao diện

npm tạo vite@latest . -- --template react-ts

# Cài đặt các phụ thuộc bổ sung với Cursor
```

Dấu nhắc con trỏ:

"Thiết lập giao diện React với Vite + TypeScript + Tailwind CSS. Cài đặt các thành phần phụ thuộc: react-router-dom, axios, react-query, zustand, react-hook-form, zod. Thiết lập theo thông số kỹ thuật UI từ /docs/design/."

5.2 Triển khai hệ thống thiết kế

Dấu nhắc con trỏ:

"Tạo các thành phần hệ thống thiết kế theo /docs/design/ui_specs.md. Bao gồm các thành phần Button, Input, Card, Modal, Layout với Tailwind CSS. Sử dụng bảng màu và kiểu chữ tiếng Việt."

5.3 Trang xác thực

Dấu nhắc con trỏ:

"Tạo trang Đăng nhập/Đăng ký với thiết kế đáp ứng. Bao gồm định tuyến dựa trên vai trò và ngôn ngữ tiếng Việt. Kết nối với API xác thực phụ trợ."

5.4 Triển khai bảng điều khiển

Dấu nhắc con trỏ:

"Tạo trang Dashboard theo wireframes trong /docs/design/. Bao gồm số liệu thống kê nhanh, cảnh báo, hoạt động gần đây, trạng thái phòng ban. Sử dụng dữ liệu mẫu từ /docs/data/ để thử nghiệm."

5.5Giao diện quản lý hàng tồn kho

Dấu nhắc con trỏ:

"Tạo giao diện quản lý hàng tồn kho: các thành phần ItemList, ItemCard, ItemDetail. Bao gồm các hoạt động tìm kiếm, lọc, CRUD. Thiết kế đáp ứng ưu tiên thiết bị di động."

Kiểm tra giao diện:

```
đáp mạnh

npm chạy dev

# Mở http://localhost:5173
```

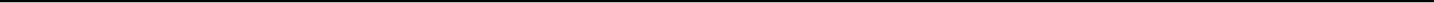
Cam kết tiến độ:

```
đáp mạnh

git add .

git commit -m "Frontend: Hệ thống thiết kế + Xác thực + Bảng điều khiển + Giao diện người dùng hàng

tồn kho" git push origin main
```



BƯỚC 6: TÍCH HỢP OCR (Tuần 3-4)

6.1 Thiết lập Google Vision API

- . Vào Google Cloud Console
- . Tạo dự án mới: "OCR hàng tồn kho nhà hàng"
- . Bật API Vision
- . Tạo thông tin xác thực: Khóa API
- . Thêm khóa API vào tệp .env

6.2 Triển khai OCR ở phía sau

Dấu nhắc con trỏ:

"Triển khai dịch vụ xử lý OCR với Google Vision API. Bao gồm xử lý trước hình ảnh, trích xuất văn bản, chấm điểm độ tin cậy. Xử lý văn bản tiếng Việt và logic kinh doanh từ /docs/requirements/business_rules.md."

6.3Hệ thống kết hợp thông minh

Dấu nhắc con trỏ:

"Tạo hệ thống đối sánh thông minh cho kết quả OCR. Sử dụng đối sánh mờ với bí danh tiếng Việt từ dữ liệu mẫu. Bao gồm tính năng chấm điểm độ tin cậy và giao diện đánh giá thủ công."

6.4 Giao diện camera phía trước

Dấu nhắc con trỏ:

"Tạo giao diện chụp ảnh cho điện thoại di động. Bao gồm kính ngắm, điều khiển chụp, xem trước hình ảnh, trạng thái xử lý OCR. Thực hiện theo các mẫu thiết kế di động từ /docs/design/."

6.5 Luồng xử lý biên lai

Dấu nhắc con trỏ:

"Triển khai quy trình xử lý biên lai hoàn chỉnh: Camera OCR Xem lại Lưu. Bao gồm chỉ báo tiến trình, xử lý lỗi và hỗ trợ tiếng Việt."

Kiểm tra OCR với hình ảnh mẫu:

đáp mạnh

```
# Tải lên mẫu biên lai từ /docs/data/sample_receipts/
# Kiểm tra độ chính xác với biên lai in máy và viết tay
```

Cam kết tiến độ:

đáp mạnh

```
git add .
git commit -m "OCR: Tích hợp Google Vision + So sánh thông minh + Giao diện camera" git push
origin main
```

BƯỚC 7: HỆ THỐNG HÒA GIẢI (Tuần 4-5)

7.1 Logic đối chiếu

Dấu nhắc con trỏ:

"Triển khai hệ thống đối chiếu theo công thức: RÚT HÀNG = BÁN + TRẢ HÀNG + LÃNG PHÍ + NHÂN VIÊN TIÊU THỤ + MẪU. Bao gồm tính toán chênh lệch, mức cảnh báo, quy trình phê duyệt theo quy tắc kinh doanh."

7.2 Quản lý sự khác biệt

Dấu nhắc con trở:

"Tạo hệ thống quản lý sai lệch. Bao gồm biểu mẫu giải trình, tải lên bằng chứng hình ảnh, quy trình phê duyệt của quản lý, phát hiện mẫu cho các vấn đề lặp lại."

7.3 Cảnh báo thời gian thực

Dấu nhắc con trở:

"Triển khai hệ thống cảnh báo thời gian thực với WebSocket. Bao gồm các mức cảnh báo (có thể chấp nhận, cảnh báo, điều tra, nghiêm trọng), định tuyến thông báo, quy tắc leo thang."

7.4 Bảng điều khiển của phòng ban

Dấu nhắc con trở:

"Tạo bảng điều khiển đối chiếu theo từng phòng ban. Bao gồm các đối chiếu đang chờ xử lý, báo cáo sai lệch, số liệu hiệu suất. Hỗ trợ giao diện di động."

Đối chiếu thử nghiệm:

```
đáp mạnh

# Test với dữ liệu đối chiếu mẫu

# Xác minh mức cảnh báo và quy trình phê duyệt

# Kiểm tra giao diện di động
```

Cam kết tiến độ:

```
đáp mạnh

git add .

git commit -m "Đối chiếu: Logic + Cảnh báo + Bảng điều khiển phòng ban + Quy trình làm việc" git

push origin main
```

BƯỚC 8: BÁO CÁO & PHÂN TÍCH(Tuần5-6)

8.1 Tạo báo cáo

Dấu nhắc con trở:

"Triển khai hệ thống báo cáo: Tóm tắt hàng ngày, Phân tích tổn thất, Hiệu suất phòng ban, Xu hướng hàng tồn kho. Bao gồm định dạng tiếng Việt, khả năng xuất (PDF, Excel), báo cáo theo lịch trình."

8.2 Bảng điều khiển phân tích

Dấu nhắc con trỏ:

"Tạo bảng phân tích với biểu đồ và KPI. Bao gồm xu hướng hàng tồn kho, phân tích tổn thất, so sánh phòng ban.
Sử dụng Chart.js hoặc Recharts với thiết kế đáp ứng."

8.3Chức năng xuất

Dấu nhắc con trỏ:

"Triển khai chức năng xuất cho báo cáo. Hỗ trợ các định dạng PDF, Excel, CSV. Bao gồm định dạng tiếng Việt, tiêu đề thích hợp, thông tin kiểm tra."

Báo cáo thử nghiệm:

```
đáp mạnh

# Tạo báo cáo với dữ liệu mẫu
# Kiểm tra chức năng xuất khẩu
# Kiểm tra định dạng tiếng Việt
```

Cam kết tiến độ:

```
đáp mạnh

git add .
git commit -m "Báo cáo: Bảng điều khiển phân tích + Chức năng xuất + Định dạng tiếng Việt" git push
origin main
```

BƯỚC 9: CHUẨN BỊ TRIỂN KHAI (Tuần 6)

9.1 Cấu hình môi trường

Dấu nhắc con trỏ:

"Thiết lập cấu hình môi trường sản xuất. Bao gồm Dockerfile, docker-compose.yml, xác thực biến môi trường, cấu hình ghi nhật ký, tăng cường bảo mật."

9.2 Di chuyển cơ sở dữ liệuScripts

Dấu nhắc con trỏ:

"Tạo tập lệnh di chuyển cơ sở dữ liệu và hạt giống cho sản xuất. Bao gồm tải dữ liệu mẫu, tạo chỉ mục, tối ưu hóa hiệu suất."

9.3 Tối ưu hóa xây dựng

Dấu nhắc con trỏ:

"Tối ưu hóa quy trình xây dựng cho sản xuất. Bao gồm phân tách mã, tối ưu hóa hình ảnh, phân tích gói, chiến lược lưu trữ đệm. Mục tiêu số liệu hiệu suất từ các yêu cầu."

9.4 Triển khai bảo mật

Dấu nhắc con trỏ:

"Triển khai bảo mật sản xuất: thực thi HTTPS, cấu hình CORS, giới hạn tốc độ, xác thực đầu vào, ngăn chặn tấn công SQL injection. Thực hiện theo các yêu cầu bảo mật từ TRD."

Kiểm tra cuối cùng:

```
đáp mạnh

# Chạy bộ kiểm tra đầy đủ
npm chạy thử nghiệm

# Xây dựng phiên bản sản xuất
npm chạy xây dựng

# Kiểm tra bản dựng sản xuất tại địa phương
npm chạy bản xem trước
```

BƯỚC 10: DEPLOY TO RENDER

10.1 Triển khai Backend

- . Vào Bảng điều khiển Render
- . Nhấp vào "Mới +" "Dịch vụ web"
- . Kết nối kho lưu trữ GitHub
- . Cấu hình dịch vụ:

```
Tên: restaurant-inventory-backend
Thư mục gốc: backend
Lệnh xây dựng: npm install && npm run build
Bắt đầu lệnh: npm start
```

. Thêm biến môi trường:

```
NODE_ENV=sản xuất
DATABASE_URL=url-cơ-sở-sản-xuất-của-bạn
JWT_SECRET=bí mật sản xuất của bạn
GOOGLE_VISION_API_KEY=khóa-api-của-bạn
```

. Triển khai

10.2Triển khai giao diện

. Render Dashboard "Mới +" "Trang web tĩnh"

. Kết nối cùng một kho lưu trữ GitHub

. Cấu hình:

```
Tên: restaurant-inventory-frontend
Thư mục gốc: frontend
Lệnh xây dựng: npm install && npm run build
Xuất bản thư mục: dist
```

. Thêm biến môi trường:

```
VITE_API_URL=https://your-backend-url.onrender.com
VITE_APP_NAME=Quản lý hàng tồn kho nhà hàng
```

. Triển khai

10.3 Thiết lập cơ sở dữ liệu khi sản xuất

Tùy chọn A: Hiển thị PostgreSQL:

. Render Dashboard "Mới +" "PostgreSQL"

. Kết nối với dịch vụ phụ trợ

. Chạy di chuyển:

```
đập mạnh

# Kết nối đến DB sản xuất và chạy:
npm run db:migrate
npm run db:seed
```

Tùy chọn B: Cơ sở dữ liệu bên ngoài (Neon.tech):

. Cập nhật DATABASE_URL trong môi trường Render

. Chạy di chuyển từ máy cục bộ

10.4 Cấu hình miền

- . Tên miền tùy chỉnh (tùy chọn):
 - Trong Cài đặt dịch vụ Render
 - Thêm tên miền tùy chỉnh
 - Cấu hình bản ghi DNS

- . Chứng chỉ HTTPS:
 - Kết xuất tự động cung cấp SSL
 - Xác minh HTTPS hoạt động

BƯỚC 11: TESTING & QA (Tuần 7)

11.1 Kiểm thử chức năng

Danh sách kiểm tra thử nghiệm:

đáp mạnh	
Xác thực người dùng (tất cả các vai trò)	Hoạt động CRUD kiểm kê
Xử lý biên lai OCR (máy + viết tay)	Quy trình đối chiếu
Hệ thống cảnh báo	Báo cáo & xuất
Phản hồi trên thiết bị di động	Hỗ trợ tiếng Việt

11.2Kiểm tra hiệu suất

Dấu nhắc con trỏ:

"Tạo tập lệnh kiểm tra hiệu suất. Kiểm tra thời gian phản hồi của API, tốc độ xử lý OCR, thời gian tải bảng điều khiển. Xác minh mục tiêu từ yêu cầu TRD được phản hồi."

11.3Kiểm tra bảo mật

Danh sách kiểm tra bảo mật:

đáp mạnh	
Bảo vệ chống SQL injection	
Ngăn chặn XSS	
Nỗ lực bỏ qua xác thực tệp lên	Bảo mật tải
Giới hạn tốc độ	Thực thi HTTPS

11.4 Kiểm tra sự chấp nhận của người dùng

Các tình huống UAT:

đáp mạnh	
	Chủ sở hữu có thể xem tất cả các
báo cáo	Người quản lý có thể chấp thuận các
sai lệch	Giám sát viên có thể quản lý hàng tồn kho của
phòng ban	Nhân viên có thể xử lý biên lai và gửi báo
cáo	Giao diện di động hoạt động trên máy tính bảng

BƯỚC 12: KHAI THÁC & GIÁM SÁT (Tuần 8)

12.1 Ra mắt mềm

- . Triển khai phiên bản cuối cùng
- . Tải dữ liệu mẫu
- . Tạo tài khoản thử nghiệm cho tất cả các vai trò
- . Tài liệu hướng dẫn sử dụng
- . Thiết lập giám sát

12.2 Tài liệu đào tạo người dùng

Dấu nhắc con trỏ:

"Tạo tài liệu đào tạo người dùng: Hướng dẫn bắt đầu nhanh, kịch bản hướng dẫn bằng video, tài liệu FAQ, hướng dẫn khắc phục sự cố.
Tất cả đều bằng tiếng Việt."

12.3 Thiết lập giám sát

- . Giám sát ứng dụng:
 - Thời gian phản hồi
 - Tỷ lệ lỗi
 - Hoạt động của người dùng
 - Độ chính xác của OCR
- . Giám sát kinh doanh:
 - Độ chính xác của hàng tồn kho
 - Tiết kiệm thời gian
 - Giảm thiểu tổn thất
 - Sự chấp nhận của người dùng

12.4 Danh sách kiểm tra khi ra mắt

dập mạnh	
Triển khai sản xuất ổn định	
Hoàn tất di chuyển cơ sở dữ liệu	
Chứng chỉ SSL đang hoạt động	Cấu hình biến môi trường
Tạo tài khoản người dùng	
Dữ liệu mẫu đã được tải	Tài liệu đào tạo đã sẵn sàng
Xác định quy trình hỗ trợ	Kiểm tra quy trình sao lưu
Cấu hình cảnh báo giám sát	

DỰ ÁN THỜI GIANTÓM TẮT

Tuần 1-2: Nền tảng

- Thiết lập dự án
- Xác thực phần cuối
- Sơ đồ cơ sở dữ liệu
- Các thao tác CRUD cơ bản

Tuần 3-4: Các tính năng cốt lõi

- Giao diện người dùng phía trước
- Tích hợp OCR Xử lý
- biên lai So khớp thông minh
-

Tuần 5-6: Tính năng nâng cao

- Hệ thống đối chiếu Cảnh
- báo thời gian thực
- Báo cáo & phân tích Tối
- ưu hóa di động

Tuần 7-8: Triển khai & Ra mắt

- Triển khai sản xuất Kiểm tra
- & Đảm bảo chất lượng
- Đào tạo người dùng

- Đưa vào hoạt động

ĐO LƯỜNG THÀNH CÔNGĐỂ THEO DÕI

Các chỉ số KPI kỹ thuật:

- Độ chính xác OCR: >95% (máy), >85% (viết tay) Thời gian
- phản hồi: <2 giây Thời gian hoạt
- động: >99,5%
- Hiệu suất di động: Thời gian tải <3 giây

KPI kinh doanh:

- Độ chính xác của hàng tồn kho:
- >98% Tiết kiệm thời gian: Giảm
- 80% Giảm tổn thất: Cải thiện 50% Tỷ
- lệ người dùng áp dụng: >90% sử dụng tích cực

Mục tiêu ROI:

- Hòa vốn: 6-12 tháng
- Tiết kiệm hàng năm: 15-25% chi phí tồn kho
- Tăng năng suất: Giảm 80% thời gian Cải
- thiện lợi nhuận: Tăng biên lợi nhuận 3-5%

KHẮC PHỤC SỰ CỐ THƯỜNG GẶP

Các vấn đề phát triển:


```
đáp mạnh

# Lỗi kết nối cơ sở dữ liệu

Kiểm tra định dạng DATABASE_URL

Kiểm tra cơ sở dữ liệu đang chạy

Kiểm tra cài đặt tường lửa


# Độ chính xác OCR thấp

Xác minh cấu hình khóa API      Kiểm tra

với hình ảnh mẫu      Kiểm tra yêu

cầu về chất lượng hình ảnh


# Vấn đề giao diện di động

Kiểm tra trên thiết bị thực tế

Xác minh kích thước mục tiêu cảm
ứng      Kiểm tra điểm dừng phản hồi
```

Các vấn đề triển khai:

```
đáp mạnh

# Xây dựng thất bại

Kiểm tra khả năng tương thích của phiên bản

Node.js      Xác minh các biến môi trường

Xem lại nhật ký xây dựng


# Lỗi thời gian chạy

Kiểm tra cấu hình môi trường      Xác minh kết

nối cơ sở dữ liệu      Xem lại nhật ký ứng

dụng
```

Các vấn đề về hiệu suất:

```
đáp mạnh

# Phản hồi API chậm

Kiểm tra chỉ mục cơ sở dữ liệu

Xem lại tối ưu hóa truy vấn

Giám sát tài nguyên máy chủ


# Xử lý OCR chậm      Xác minh

giới hạn tốc độ API      Kiểm

tra tối ưu hóa kích thước hình ảnh      Xem

lại hàng đợi công việc nền
```

Cải tiến Giai đoạn 2 (Tháng 2-3):

- Phát triển ứng dụng di động gốc
- Tích hợp hệ thống POS
- Phân tích nâng cao & thông tin chi tiết về AI
- Hỗ trợ nhiều địa điểm

Giai đoạn 3 mở rộng quy mô (Tháng 4-6):

- Học máy để dự báo nhu cầu
- Tích hợp nhà cung cấp & EDI
- Tích hợp hệ thống tài chính
- Tự động hóa quy trình làm việc

Với hướng dẫn chi tiết này, bạn có thể tạo một hệ thống Quản lý kho nhà hàng hoàn chỉnh trong 8 tuần!