

**BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC ĐIỆN LỰC  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO KẾT THÚC HỌC PHẦN**

**LẬP TRÌNH BLOCKCHAIN**

**MẠNG CHIA SẺ KHO BÀI**

<b>Giảng viên hướng dẫn</b>	<b>:Nguyễn Thị Hồng Khánh</b>
<b>Sinh viên thực hiện</b>	<b>:Nguyễn Tiến Thanh</b>
<b>Ngành</b>	<b>:Công nghệ thông tin</b>
<b>Chuyên ngành</b>	<b>:Công nghệ phần mềm</b>
<b>Lớp</b>	<b>:D17CNPM1</b>
<b>Khóa</b>	<b>:2022 - 2027</b>

**Hà Nội, tháng 11 năm 2025**

## LỜI CẢM ƠN

Trong bài báo cáo chuyên đề này, chúng em xin gửi lời cảm ơn sâu sắc và lòng biết ơn chân thành đến những người đã đóng góp và hỗ trợ chúng em trong quá trình nghiên cứu và viết bài báo cáo này.

Đầu tiên, chúng em muốn bày tỏ lòng biết ơn đến giảng viên hướng dẫn của chúng em, cô Nguyễn Thị Hồng Khánh. Sự tận tâm và kiến thức sâu rộng của cô đã giúp định hình và chỉ dẫn cho chúng em trong quá trình nghiên cứu. Những ý kiến xây dựng và hướng dẫn của cô đã cung cấp cho chúng em một cơ sở vững chắc để phát triển ý tưởng và triển khai nghiên cứu của mình.

Chúng em cũng muốn gửi lời cảm ơn đến các thành viên trong ban giảng dạy và các chuyên gia đã dành thời gian để đọc và đánh giá bài báo cáo chuyên đề của chúng em. Sự đánh giá chuyên môn và phản hồi xây dựng từ thầy cô đã giúp chúng em nắm bắt những khía cạnh quan trọng và cải thiện chất lượng của bài viết.

Cuối cùng, chúng em muốn gửi lời cảm ơn đến tất cả những người đã cung cấp tư liệu, dữ liệu và thông tin quan trọng cho bài báo cáo chuyên đề này. Các nguồn tài liệu và thông tin này đã đóng góp quan trọng để nâng cao chất lượng và sự chính xác của nghiên cứu.

Một lần nữa, chúng em xin chân thành cảm ơn tất cả những người đã đóng góp và hỗ trợ chúng em trong quá trình nghiên cứu và viết bài báo cáo chuyên đề này. Sự giúp đỡ của các bạn đã làm cho dự án này trở thành hiện thực và mang lại những kết quả mà chúng em tự hào.

Chúng em xin chân thành cảm ơn!

## PHIẾU CHẤM ĐIỂM

Danh sách sinh viên tham gia:

Họ và tên	Nội dung thực hiện	Điểm	Chữ ký	Ghi chú
Nguyễn Tiến Thanh 22810310382	Lập trình, phát triển ứng dụng			

Danh sách giảng viên chấm thi:

Họ và tên giảng viên	Chữ ký	Ghi chú

# MỤC LỤC

Trang

LỜI CẢM ƠN .....	
PHIẾU CHẤM ĐIỂM.....	
MỤC LỤC.....	
DANH MỤC HÌNH ẢNH, BẢNG BIỂU.....	
PHẦN MỞ ĐẦU .....	1
PHẦN NỘI DUNG .....	2
CHƯƠNG 1: TỔNG QUAN VỀ BLOCKCHAIN .....	2
1.1 Khái niệm về Blockchain.....	2
1.2 Đặc điểm cốt lõi của Blockchain .....	2
1.3 Cơ chế hoạt động của Blockchain.....	4
1.4 Phân loại Blockchain.....	4
CHƯƠNG 2: NỀN TẢNG CÔNG NGHỆ VÀ KIẾN TRÚC HỆ THỐNG .....	6
2.1 Hợp đồng Thông minh (Smart Contract) và Môi trường thực thi .....	6
2.1.1 Ngôn ngữ lập trình Solidity .....	6
2.1.2 Máy ảo Ethereum (EVM) và Hardhat Network.....	6
2.2 Kiến trúc Ứng dụng Phi tập trung (DApp Architecture) .....	7
2.2.1 Mô hình Ba tầng của DApp .....	7
2.2.2 Cơ chế Đồng bộ Dữ liệu (Synchronization Mechanism).....	7
2.3 Các Công nghệ và Công cụ Phát triển Chủ chốt.....	8
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	10
3.1 Phân tích Yêu cầu Nghiệp vụ (Business Requirements Analysis).....	10
3.2 Thiết kế Smart Contract: WarehouseLeasing.sol .....	11
3.2.1 Cấu trúc dữ liệu chính (Structs) .....	11
3.2.2 Các chức năng quan trọng.....	11
3.3 Thiết kế Mô hình Dữ liệu Off-chain (MySQL Database).....	12

3.3.1 Bảng warehouses.....	13
3.3.2 Bảng leases.....	13
3.3.3 Bảng transactions .....	14
3.3.4 Bảng users .....	14
3.4 Luồng hoạt động chính và Cơ chế Đồng bộ .....	14
CHƯƠNG 4: THIẾT KẾ ỨNG DỤNG .....	16
4.1 Trang Home.....	16
4.2 Trang User.....	16
4.3 Trang Warehouse.....	17
4.4 Trang WarehouseDetail .....	17
4.5 Trang My-warehouses.....	18
4.6 Trang My-leases .....	18
4.7 Trang Register-warehouse.....	19
PHẦN KẾT LUẬN .....	20
DANH MỤC TÀI LIỆU THAM KHẢO .....	21

## DANH MỤC HÌNH ẢNH, BẢNG BIỂU

	Trang
Bảng 1.1: Đặc điểm cốt lõi của Blockchain.....	2
Bảng 1.2: Phân loại Blockchain.....	4
Bảng 2.1: Mô hình 3 tầng của DApp .....	7
Bảng 2.2: Các công nghệ và công cụ phát triển.....	8
Bảng 3.1: Phân tích Yêu cầu Nghiệp vụ .....	10
Bảng 3.2: Các chức năng quan trọng .....	11
Hình 3.1: Bảng warehouse .....	13
Hình 3.2: Bảng leases.....	13
Hình 3.3: Bảng transactions .....	14
Hình 3.4: Bảng user.....	14
Hình 4.1: Trang Home.....	16
Hình 4.2: Trang User.....	16
Hình 4.3: Trang Warehouse.....	17
Hình 4.4: Trang WarehouseDetail .....	17
Hình 4.5: Trang My-warehouses.....	18
Hình 4.6: Trang My-leases.....	18
Hình 4.7: Trang Register-warehouse.....	19

## PHẦN MỞ ĐẦU

Trong bối cảnh nền kinh tế chuỗi cung ứng và logistics phát triển mạnh mẽ, nhu cầu về một nền tảng chia sẻ kho bãi hiệu quả, minh bạch và an toàn trở nên cấp thiết. Các mô hình quản lý kho bãi truyền thống thường tồn tại những hạn chế về chi phí trung gian cao, thiếu niềm tin giữa các bên và rủi ro về tranh chấp hợp đồng. Để khắc phục những vấn đề này, đề tài " Mạng chia sẻ kho bãi" được lựa chọn, với mục tiêu ứng dụng công nghệ Blockchain và Hợp đồng thông minh (Smart Contract) để xây dựng một nền tảng phi tập trung (DApp) an toàn và tự động hóa.

Dự án đã triển khai một hệ thống hoàn chỉnh, bao gồm Smart Contract Solidity (quản lý đăng ký và thuê kho bãi, thanh toán ETH), hệ thống Backend Node.js/Express và Frontend thân thiện React/Material-UI hỗ trợ tiếng Việt. Đặc biệt, hệ thống được thiết kế với cơ chế Persistent Blockchain và Auto Sync để đảm bảo dữ liệu không bao giờ bị mất khi khởi động lại và luôn đồng bộ giữa blockchain với cơ sở dữ liệu ngoài, từ đó tạo ra một giải pháp chia sẻ kho bãi minh bạch, hoạt động bền vững và đáng tin cậy.

## PHẦN NỘI DUNG

### CHƯƠNG 1: TỔNG QUAN VỀ BLOCKCHAIN

#### 1.1 Khái niệm về Blockchain

**Blockchain** (Chuỗi khối) là một công nghệ cơ sở dữ liệu phân tán, được thiết kế để tạo ra một sổ cái kỹ thuật số an toàn, minh bạch và bất biến. Công nghệ này tổ chức dữ liệu thành các **khối (blocks)** và sử dụng mật mã để liên kết chúng theo trình tự thời gian, tạo thành một **chuỗi (chain)**. Về bản chất, Blockchain là một cơ chế đồng thuận phân tán, cho phép các bên không tin cậy nhau có thể giao dịch và lưu trữ dữ liệu một cách đáng tin cậy mà không cần đến bên trung gian tập trung.

#### 1.2 Đặc điểm cốt lõi của Blockchain

*Bảng 1.1: Đặc điểm cốt lõi của Blockchain*

Đặc điểm	Mô tả
Phi tập trung (Decentralization)	Thay vì lưu trữ dữ liệu tại một máy chủ hoặc tổ chức duy nhất (kiến trúc tập trung), Blockchain phân tán các bản sao của sổ cái kỹ thuật số đến tất cả các máy tính (node) tham gia mạng. Điều này giúp loại bỏ điểm lỗi đơn lẻ (Single Point of Failure), tăng cường khả năng chống kiểm duyệt và giảm thiểu rủi ro bị thao túng dữ liệu từ một bên duy nhất.
Bất biến (Immutability)	Một khi dữ liệu giao dịch đã được ghi vào một khối và thêm vào chuỗi bằng cơ chế đồng thuận, nó sẽ trở nên vĩnh



	<p>viễn và không thể thay đổi hay xóa bỏ. Tính bất biến này được đảm bảo thông qua việc sử dụng Hàm băm mật mã (Cryptographic Hash Functions). Mỗi khối mới đều chứa mã băm của khối ngay trước đó; nếu bất kỳ dữ liệu nào trong khối trước bị thay đổi, mã băm của nó cũng sẽ thay đổi, làm hỏng toàn bộ chuỗi khối liên kết.</p>
Minh bạch (Transparency)	<p>Mọi thành viên tham gia mạng lưới đều có quyền truy cập và kiểm tra lịch sử giao dịch trên sổ cái. Mặc dù danh tính người dùng thực thường được ẩn dưới dạng địa chỉ ví mật mã, nhưng hoạt động giao dịch của họ lại được công khai, tạo ra một môi trường minh bạch về mặt hoạt động.</p>
Tính đồng thuận (Consensus)	<p>Đây là tập hợp các quy tắc và cơ chế mà các node trong mạng Blockchain sử dụng để đạt được thỏa thuận chung về tính hợp lệ của giao dịch và thứ tự thêm khối mới vào chuỗi. Các thuật toán đồng thuận phổ biến bao gồm Proof-of-Work (PoW) và Proof-of-Stake (PoS). Cơ chế đồng thuận đảm</p>

	bảo rằng mọi bản sao sổ cái phân tán đều giống nhau.
--	--

### 1.3 Cơ chế hoạt động của Blockchain

Quy trình thêm một giao dịch vào Blockchain diễn ra theo các bước cơ bản sau:

**Bước 1:** Khởi tạo: Một giao dịch (ví dụ: A gửi tiền cho B, hoặc trong dự án này là Người thuê tạo hợp đồng thuê kho bãi) được khởi tạo và ký số.

**Bước 2:** Phát tán: Giao dịch được truyền tải đến mạng lưới các node.

**Bước 3:** Xác thực: Các node xác thực tính hợp lệ của giao dịch (chữ ký số, số dư, quy tắc).

**Bước 4:** Tập hợp: Các giao dịch hợp lệ được tập hợp thành một khối mới.

**Bước 5:** Thêm khối: Khối mới phải trải qua quá trình đồng thuận để được chấp nhận và thêm vào cuối chuỗi khối hiện có.

**Bước 6:** Cập nhật: Khi khối được thêm vào, tất cả các node trong mạng sẽ cập nhật sổ cái của mình, hoàn tất việc ghi nhận giao dịch.

### 1.4 Phân loại Blockchain

Blockchain được phân loại dựa trên quyền truy cập và kiểm soát

*Bảng 1.2: Phân loại Blockchain*

Loại Blockchain	Mức độ truy cập	Tính ứng dụng
Public Blockchain (Blockchain Công khai)	Mở, không cần cấp phép (Permissionless). Bất kỳ ai cũng có thể tham gia.	Tiền mã hóa (Bitcoin, Ethereum), ứng dụng phi tập trung công cộng.
Private Blockchain (Blockchain Riêng tư)	Đóng, cần cấp phép (Permissioned). Quyền	Quản lý chuỗi cung ứng nội bộ, quản lý dữ liệu doanh nghiệp.

	kiểm soát thuộc về một tổ chức duy nhất.	
Consortium Blockchain (Blockchain Liên minh)	Cần cấp phép. Quyền kiểm soát chia sẻ giữa một nhóm các tổ chức đã chọn.	Trao đổi dữ liệu liên ngân hàng, liên tổ chức.

## CHƯƠNG 2: NỀN TẢNG CÔNG NGHỆ VÀ KIẾN TRÚC HỆ THỐNG

### 2.1 Hợp đồng Thông minh (Smart Contract) và Môi trường thực thi

#### 2.1.1 Ngôn ngữ lập trình Solidity

Solidity là ngôn ngữ lập trình hướng đối tượng, được thiết kế đặc biệt để phát triển Smart Contract trên các nền tảng tương thích với Máy ảo Ethereum (EVM), như Ethereum và Hardhat Network.

- **Tính năng chính:** Hỗ trợ các kiểu dữ liệu cơ bản, kế thừa, thư viện và cơ chế bảo mật tích hợp sẵn cho các giao dịch tiền mã hóa.

- **Vai trò trong dự án:** Smart Contract WarehouseLeasing.sol được viết bằng Solidity để định nghĩa logic nghiệp vụ chính của mạng lưới, bao gồm:

- + Đăng ký và quản lý metadata của kho bãi.
- + Xử lý việc tạo hợp đồng thuê (lease), giữ tiền thanh toán (ETH) và phân phối tiền cho chủ kho bãi khi hợp đồng hoàn tất.

#### 2.1.2 Máy ảo Ethereum (EVM) và Hardhat Network

**EVM (Ethereum Virtual Machine):** Là môi trường thực thi phân tán, cô lập, nơi các Smart Contract được chạy. Mọi node trong mạng lưới đều chạy EVM để đảm bảo tính đồng nhất trong việc thực thi hợp đồng.

**Hardhat Network:** Là một môi trường phát triển cục bộ được sử dụng trong dự án. Hardhat mô phỏng đầy đủ các chức năng của mạng lưới Ethereum, cho phép các nhà phát triển triển khai (deploy), kiểm thử (test) và gỡ lỗi (debug) Smart Contract một cách nhanh chóng và miễn phí.

**Ưu điểm của Hardhat:** Hỗ trợ tính năng Persistent Blockchain được sử dụng trong dự án, cho phép lưu trữ trạng thái blockchain và địa chỉ contract qua các lần khởi động lại, giúp quá trình phát triển liên tục và ổn định.

## 2.2 Kiến trúc Ứng dụng Phi tập trung (DApp Architecture)

### 2.2.1 Mô hình Ba tầng của DApp

Hệ thống được chia thành ba tầng chính

*Bảng 2.1: Mô hình 3 tầng của DApp*

Tầng Giao diện (Presentation Layer - Frontend)	+ Sử dụng ReactJS và thư viện giao diện Material-UI.  + Chịu trách nhiệm hiển thị thông tin, xử lý tương tác người dùng, và kết nối với ví tiền mã hóa (như MetaMask) thông qua Web3 Middleware.
Tầng Logic Nghiệp vụ (Business Logic Layer)	+ Logic On-chain (Phi tập trung): Được xử lý bởi Smart Contract. Xử lý các nghiệp vụ quan trọng yêu cầu tính bất biến và độ tin cậy cao (đăng ký tài sản, giao dịch thuê, thanh toán).  + Logic Off-chain (Tập trung): Được xử lý bởi Backend API (Node.js/Express). Xử lý các nghiệp vụ yêu cầu tốc độ và khả năng truy vấn phức tạp (quản lý người dùng, xử lý hình ảnh, logic nghiệp vụ phụ trợ).
Tầng Dữ liệu (Data Layer)	+ Dữ liệu Blockchain: Lưu trữ trạng thái hợp đồng thuê, ID kho bãi trên chuỗi.  + Dữ liệu Database (MySQL): Lưu trữ dữ liệu ngoài chuỗi (Off-chain data) như URL hình ảnh, mô tả chi tiết, thông tin người dùng, và dữ liệu có thể thay đổi nhanh chóng.

### 2.2.2 Cơ chế Đồng bộ Dữ liệu (Synchronization Mechanism)

Để đảm bảo tính nhất quán giữa dữ liệu trên chuỗi và dữ liệu ngoài chuỗi, dự án đã triển khai cơ chế đồng bộ tự động:

+ Tự động Sync Warehouses: Backend liên tục theo dõi (subscribe) các sự kiện (Events) phát ra từ Smart Contract. Khi một kho bãi mới được đăng ký hoặc cập nhật trên blockchain, Backend sẽ ghi nhận thông tin đó vào Database MySQL.

+ Tăng tốc độ truy vấn: Việc sử dụng MySQL để lưu trữ bản sao dữ liệu kho bãi cho phép Frontend thực hiện các truy vấn phức tạp (tìm kiếm, lọc) với tốc độ nhanh hơn nhiều so với việc truy vấn trực tiếp lên Blockchain, cải thiện trải nghiệm người dùng.

### 2.3 Các Công nghệ và Công cụ Phát triển Chủ chốt

*Bảng 2.2: Các công nghệ và công cụ phát triển*

Loại công nghệ	Công cụ / Nền tảng	Vai trò trong dự án
Blockchain Core	Solidity, EVM	Ngôn ngữ viết Smart Contract và môi trường thực thi.
Phát triển Blockchain	Hardhat	Môi trường phát triển, kiểm thử, triển khai Smart Contract.
Web3 Middleware	Ethers.js	Thư viện JavaScript giúp Frontend và Backend tương tác với Smart Contract.
Frontend	ReactJS, Material-UI	Xây dựng giao diện người dùng hiện đại, thân thiện, tương tác trực quan.

Backend & API	Node.js, ExpressJS	Xây dựng API trung gian, xử lý logic Off-chain, và kết nối Database.
Database	MySQL	Lưu trữ dữ liệu ngoài chuỗi (hình ảnh, mô tả, caching dữ liệu).
Quản lý mã nguồn	Git, GitHub	Quản lý phiên bản và hợp tác phát triển.

## CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1 Phân tích Yêu cầu Nghiệp vụ (Business Requirements Analysis)

Hệ thống **Mạng Chia Sẻ Kho Bãi** được thiết kế để giải quyết các vấn đề về minh bạch, tin cậy và tự động hóa trong lĩnh vực cho thuê kho bãi. Các yêu cầu nghiệp vụ chính bao gồm:

*Bảng 3.1: Phân tích Yêu cầu Nghiệp vụ*

STT	Yêu cầu nghiệp vụ	Mô tả chi tiết	Tầm quan trọng
R1	Đăng ký Kho bãi (Registration)	Chủ sở hữu phải có khả năng đăng ký kho bãi mới lên Blockchain, bao gồm thông tin: địa chỉ ví chủ sở hữu, diện tích, giá thuê/ngày.	Cốt lõi (Smart Contract)
R2	Thuê Kho bãi (Leasing)	Người thuê có thể tạo hợp đồng thuê, thanh toán tổng chi phí thuê bằng ETH trực tiếp vào Smart Contract.	Cốt lõi (Smart Contract)
R3	Truy vấn Hiệu suất cao	Cần sử dụng cơ sở dữ liệu ngoài (MySQL) để cache và cho phép tìm kiếm, lọc dữ liệu kho bãi nhanh chóng.	Cần thiết (DB & API)
R4	Đồng bộ Dữ liệu	Dữ liệu giao dịch quan trọng trên Blockchain phải được tự động đồng bộ (Auto Sync) xuống MySQL để đảm bảo tính thống nhất trên giao diện.	Nâng cao (Auto Sync Service)



### 3.2 Thiết kế Smart Contract: WarehouseLeasing.sol

Smart Contract là nơi thực thi logic nghiệp vụ cốt lõi và quản lý dòng tiền ký quỹ.

#### 3.2.1 Cấu trúc dữ liệu chính (Structs)

Hợp đồng định nghĩa hai cấu trúc dữ liệu chính được lưu trữ on-chain:

1. Warehouse (Kho bãi):

- ownerAddress (address): Địa chỉ ví của chủ sở hữu.
- totalArea (uint256): Tổng diện tích (m2).
- availableArea (uint256): Diện tích còn có sẵn để cho thuê.
- pricePerSqmPerDay (uint256): Giá thuê trên mỗi m2 mỗi ngày (tính bằng Wei/ETH).
- isActive (bool): Trạng thái hoạt động.

2. Lease (Hợp đồng thuê):

- warehouseId (uint256): ID kho bãi trên Blockchain.
- tenantAddress (address): Địa chỉ ví người thuê.
- startDate (uint256): Ngày bắt đầu thuê (timestamp).
- endDate (uint256): Ngày kết thúc thuê (timestamp).
- totalAmount (uint256): Tổng số tiền ETH đã thanh toán và đang được ký quỹ.
- status (uint8): Trạng thái hợp đồng (e.g., 1: Active, 2: Completed).

#### 3.2.2 Các chức năng quan trọng

*Bảng 3.2: Các chức năng quan trọng*

Hàm	Mô tả
registerWarehouse	Ghi nhận kho bãi mới và các thông số cơ bản lên chuỗi. Phát ra sự kiện WarehouseRegistered.

updateWarehouseArea	Cho phép chủ kho cập nhật diện tích có sẵn.
createLease	Tạo hợp đồng thuê mới. Người thuê gửi giao dịch kèm theo ETH (sử dụng từ khóa payable). Hợp đồng giữ số tiền này và phát ra sự kiện LeaseCreated.
completeLease	Chuyển trạng thái hợp đồng sang Hoàn tất. Chỉ có thể được gọi bởi chủ kho hoặc hệ thống tự động. Phát ra sự kiện LeaseCompleted để Backend đồng bộ và thông báo.
getLeaseInfo	Lấy thông tin chi tiết của một hợp đồng thuê.

### 3.3 Thiết kế Mô hình Dữ liệu Off-chain (MySQL Database)

Cơ sở dữ liệu MySQL được sử dụng để lưu trữ các dữ liệu phụ trợ và bản sao (cache) của dữ liệu on-chain để phục vụ truy vấn tốc độ cao.

### 3.3.1 Bảng warehouses

#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú
<input type="checkbox"/> 1	<b>id</b> 🔑	int(11)			Không	<i>Không</i>	
<input type="checkbox"/> 2	<b>blockchain_id</b> 🔑	int(11)			Không	<i>Không</i>	
<input type="checkbox"/> 3	<b>owner_address</b> 🔑	varchar(42)	utf8mb4_unicode_ci		Không	<i>Không</i>	
<input type="checkbox"/> 4	<b>name</b>	varchar(200)	utf8mb4_unicode_ci		Không	<i>Không</i>	
<input type="checkbox"/> 5	<b>location</b>	varchar(255)	utf8mb4_unicode_ci		Không	<i>Không</i>	
<input type="checkbox"/> 6	<b>total_area</b>	decimal(10,2)			Không	<i>Không</i>	Diện tích tổng (m <sup>2</sup> )
<input type="checkbox"/> 7	<b>available_area</b>	decimal(10,2)			Không	<i>Không</i>	Diện tích còn trống (m <sup>2</sup> )
<input type="checkbox"/> 8	<b>price_per_sqm_per_day</b>	decimal(20,0)			Không	<i>Không</i>	Giá thuê mỗi m <sup>2</sup> /ngày (Wei)
<input type="checkbox"/> 9	<b>image_url</b>	varchar(500)	utf8mb4_unicode_ci		Có	<i>NULL</i>	
<input type="checkbox"/> 10	<b>description</b>	text	utf8mb4_unicode_ci		Có	<i>NULL</i>	
<input type="checkbox"/> 11	<b>is_active</b> 🔑	tinyint(1)			Có	1	
<input type="checkbox"/> 12	<b>created_at</b>	timestamp			Không	current_timestamp()	
<input type="checkbox"/> 13	<b>updated_at</b>	timestamp			Không	current_timestamp()	

Hình 3.1: Bảng warehouse

### 3.3.2 Bảng leases

#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú
1	<b>id</b> 🔑	int(11)			Không	<i>Không</i>	
2	<b>blockchain_id</b> 🔑	int(11)			Không	<i>Không</i>	
3	<b>warehouse_id</b> 🔑	int(11)			Không	<i>Không</i>	
4	<b>tenant_address</b> 🔑	varchar(42)	utf8mb4_unicode_ci		Không	<i>Không</i>	
5	<b>area</b>	decimal(10,2)			Không	<i>Không</i>	Diện tích thuê (m <sup>2</sup> )
6	<b>start_date</b>	datetime			Không	<i>Không</i>	
7	<b>end_date</b>	datetime			Không	<i>Không</i>	
8	<b>total_price</b>	decimal(20,0)			Không	<i>Không</i>	Tổng giá (Wei)
9	<b>transaction_hash</b>	varchar(66)	utf8mb4_unicode_ci		Có	<i>NULL</i>	
10	<b>is_active</b> 🔑	tinyint(1)			Có	1	
11	<b>is_completed</b> 🔑	tinyint(1)			Có	0	
12	<b>created_at</b>	timestamp			Không	current_timestamp()	
13	<b>updated_at</b>	timestamp			Không	current_timestamp()	

Hình 3.2: Bảng leases

### 3.3.3 Bảng transactions

#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú
1	<b>id</b> 🔑	int(11)			Không	Không	
2	<b>transaction_hash</b> 🔑	varchar(66)	utf8mb4_unicode_ci		Không	Không	
3	<b>from_address</b> 🔑	varchar(42)	utf8mb4_unicode_ci		Không	Không	
4	<b>to_address</b>	varchar(42)	utf8mb4_unicode_ci		Có	NULL	
5	<b>type</b> 🔑	enum('register_warehouse', 'create_lease', 'complete')	utf8mb4_unicode_ci		Không	Không	
6	<b>amount</b>	decimal(20,0)			Có	NULL	
7	<b>status</b>	enum('pending', 'success', 'failed')	utf8mb4_unicode_ci		Có	pending	
8	<b>block_number</b>	int(11)			Có	NULL	
9	<b>created_at</b>	timestamp			Không	current_timestamp()	

Hình 3.3: Bảng transactions

### 3.3.4 Bảng users

#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú
1	<b>id</b> 🔑	int(11)			Không	Không	
2	<b>wallet_address</b> 🔑	varchar(42)	utf8mb4_unicode_ci		Không	Không	
3	<b>name</b>	varchar(100)	utf8mb4_unicode_ci		Có	NULL	
4	<b>email</b>	varchar(100)	utf8mb4_unicode_ci		Có	NULL	
5	<b>phone</b>	varchar(20)	utf8mb4_unicode_ci		Có	NULL	
6	<b>avatar_url</b>	varchar(255)	utf8mb4_unicode_ci		Có	NULL	
7	<b>created_at</b>	timestamp			Không	current_timestamp()	
8	<b>updated_at</b>	timestamp			Không	current_timestamp()	

Hình 3.4: Bảng user

## 3.4 Luồng hoạt động chính và Cơ chế Đồng bộ

Quá trình vận hành hệ thống được thực hiện qua luồng dữ liệu On-chain ↔ Off-chain và các dịch vụ tự động:

- **Lắng nghe sự kiện (Event Listening):** Dịch vụ Backend khởi tạo một trình lắng nghe (listener) sử dụng Ethers.js, kết nối với Smart Contract đang triển khai trên Hardhat Network.

- **Đăng ký/Thuê:**

- + Người dùng thực hiện giao dịch `registerWarehouse()` hoặc `createLease()`.

- + Smart Contract phát ra sự kiện (Events) tương ứng (`WarehouseRegistered`, `LeaseCreated`).

- **Ghi nhận Đồng bộ (Auto Sync):**

- + Trình lắng nghe của Backend bắt được sự kiện.

- + Backend sử dụng dữ liệu từ sự kiện để chèn (INSERT) hoặc cập nhật (UPDATE) thông tin vào các bảng MySQL (`warehouses`, `leases`).

- **Tự động hóa hoàn tất hợp đồng:**

- + Dịch vụ nền (`auto-complete-expired-leases.js`) chạy định kỳ, quét các hợp đồng trong MySQL gần ngày hết hạn.

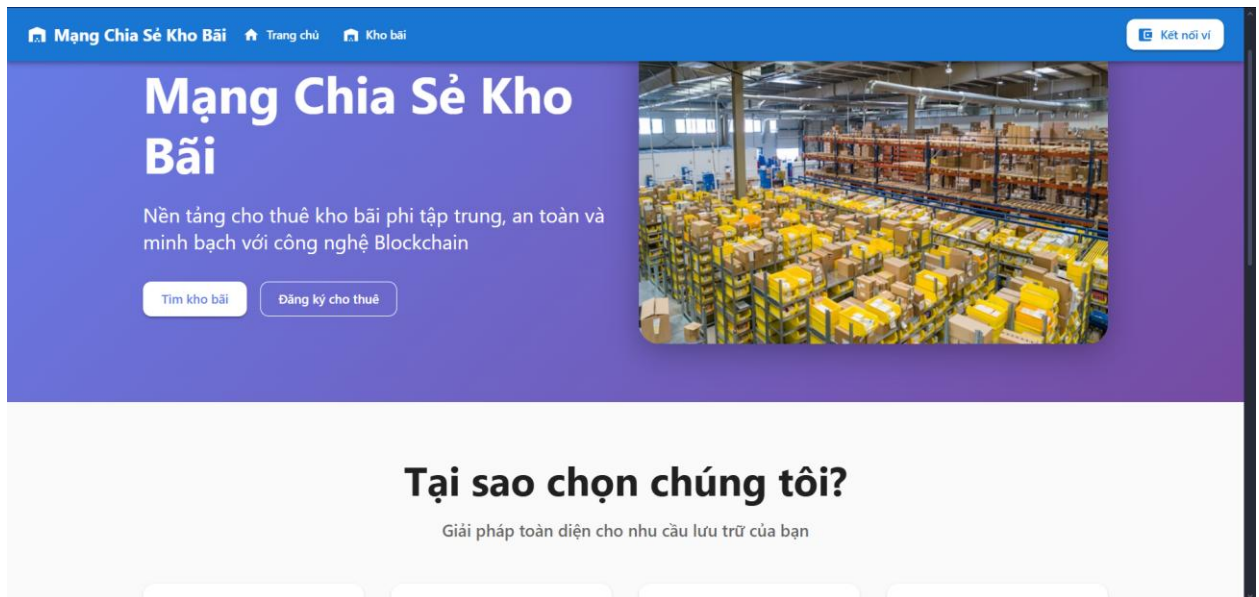
- + Dịch vụ này thực hiện giao dịch gọi hàm `completeLease()` trên Smart Contract.

- + Sự kiện `LeaseCompleted` được phát ra, và Backend lại đồng bộ trạng thái mới này vào MySQL.

Luồng thiết kế này đảm bảo rằng tất cả các nghiệp vụ cốt lõi và dòng tiền ký quỹ được quản lý an toàn bởi Smart Contract, trong khi dữ liệu hiển thị được tối ưu tốc độ truy vấn bằng cách lưu trữ tạm thời trong MySQL.

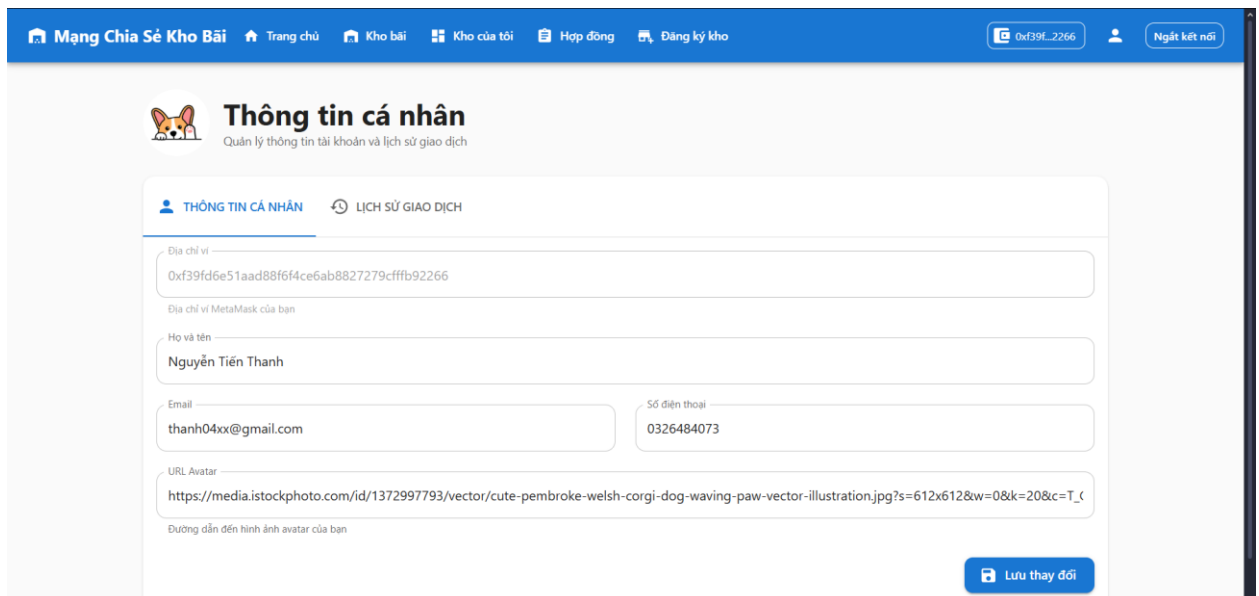
## CHƯƠNG 4: THIẾT KẾ ỨNG DỤNG

### 4.1 Trang Home



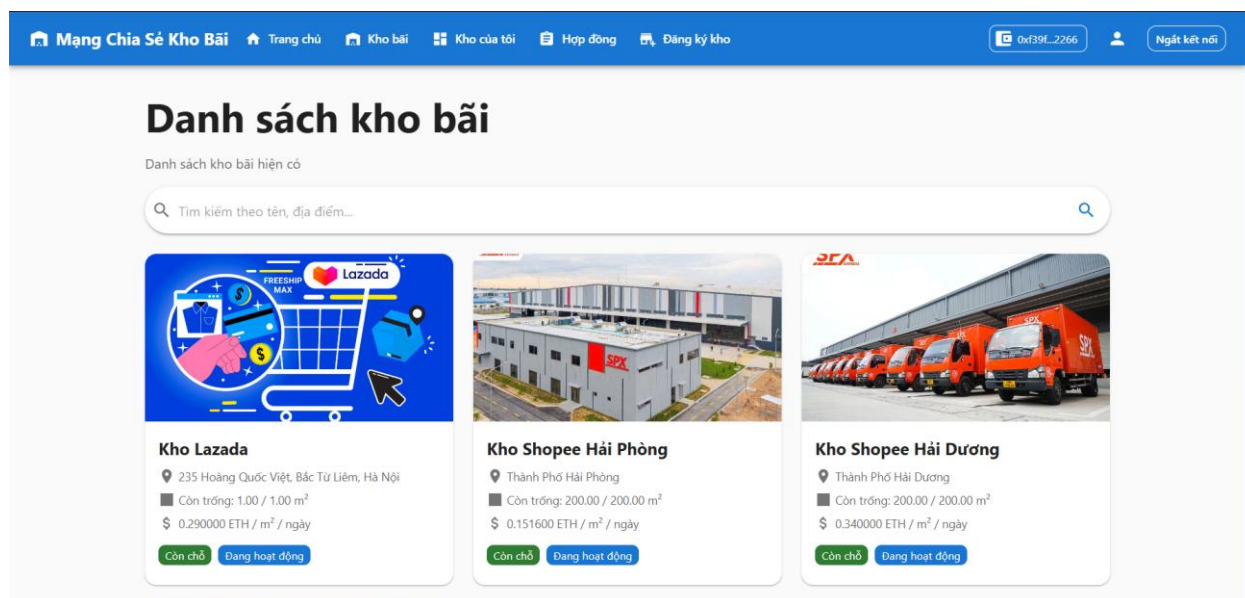
Hình 4.1: Trang Home

### 4.2 Trang User



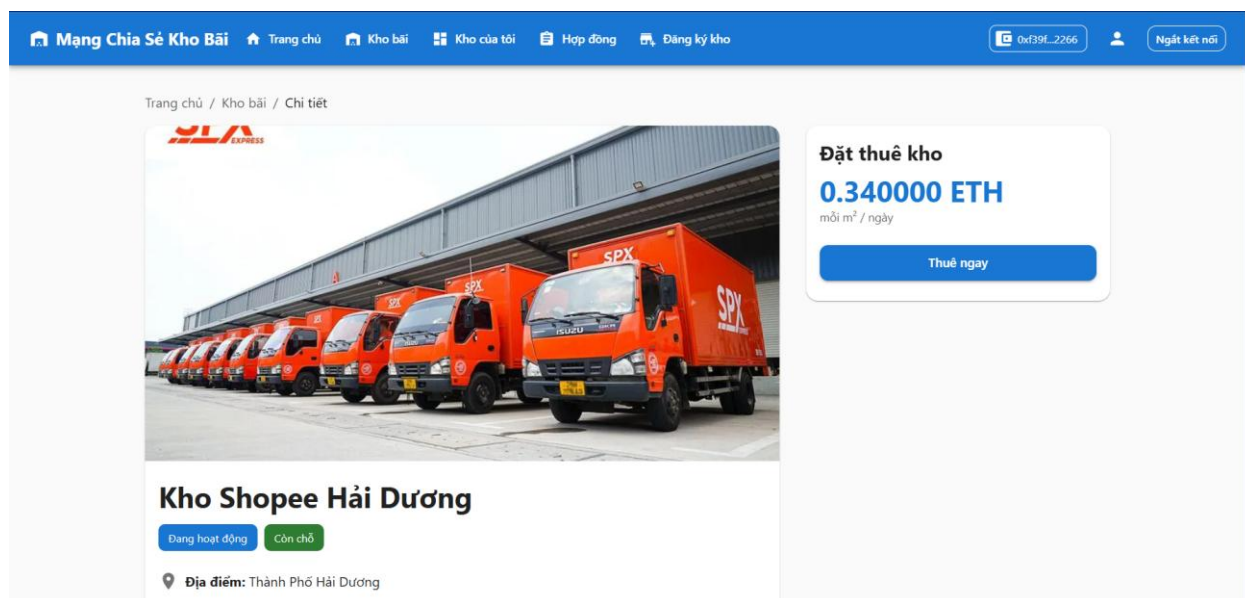
Hình 4.2: Trang User

## 4.3 Trang Warehouse



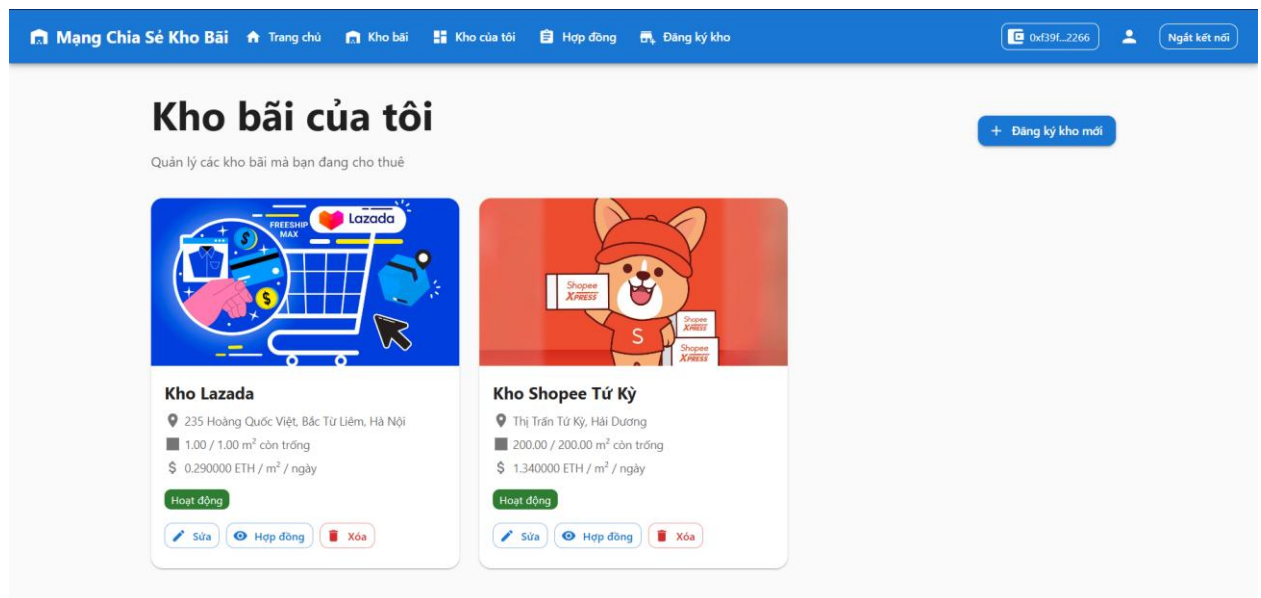
Hình 4.3: Trang Warehouse

## 4.4 Trang WarehouseDetail



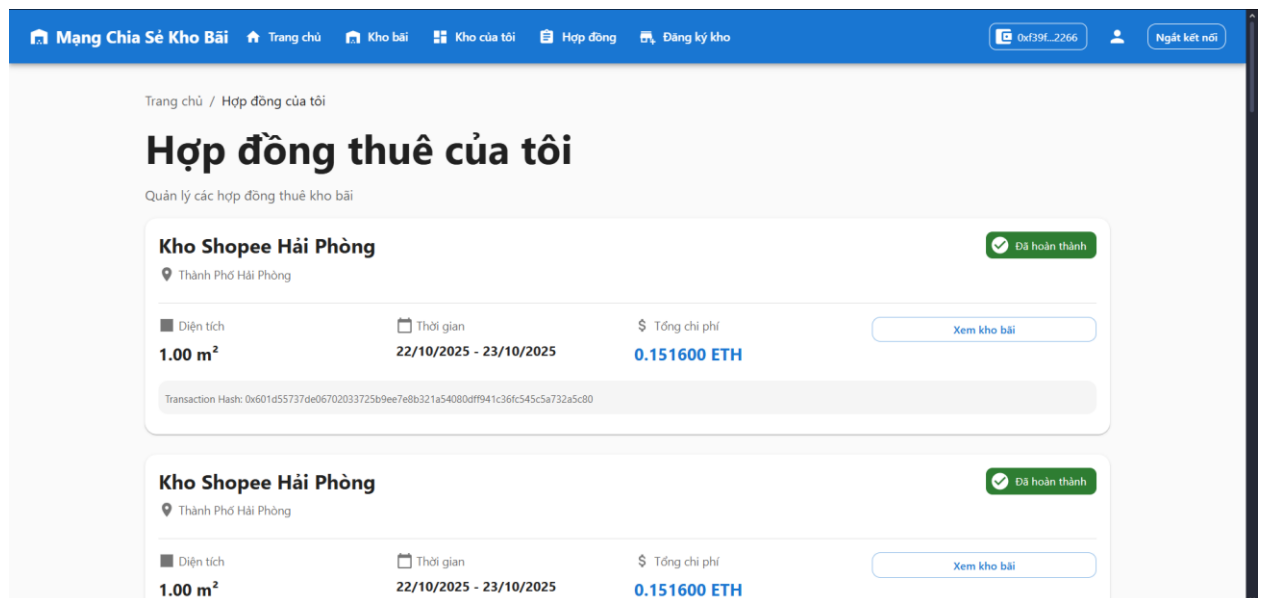
Hình 4.4: Trang WarehouseDetail

## 4.5 Trang My-warehouses



Hình 4.5: Trang My-warehouses

## 4.6 Trang My-leases



Hình 4.6: Trang My-leases



## 4.7 Trang Register-warehouse

The screenshot shows the 'Đăng ký kho bãi mới' (Register new warehouse) page. The header is blue with navigation links: 'Mạng Chia Sẻ Kho Bãi', 'Trang chủ', 'Kho bãi', 'Kho của tôi', 'Hợp đồng', and 'Đăng ký kho'. On the right of the header, there is a user profile icon, a wallet address '0xf39f...2266', and a 'Ngắt kết nối' (Disconnect) button. The main content area has a breadcrumb trail: 'Trang chủ / Kho của tôi / Đăng ký'. The form is titled 'Đăng ký kho bãi mới' with a subtitle 'Đăng ký kho bãi của bạn để cho thuê'. It contains several input fields: 'Tên kho bãi \*\*', 'Địa điểm \*\*', 'Tổng diện tích (m<sup>2</sup>) \*\*', and 'Giá thuê (ETH/m<sup>2</sup>/ngày) \*\*'. Below the last two fields, there is a small text 'VD: 0.00005'. There is also a 'Chọn ảnh từ máy' (Choose image from machine) button and a 'URL hình ảnh' (Image URL) field. A note below the URL field says 'Bạn có thể dán URL hoặc chọn file để tải lên (tối đa 300 ký tự)'. At the bottom, there is a 'Mô tả' (Description) text area.

*Hình 4.7: Trang Register-warehouse*

## PHẦN KẾT LUẬN

Đề tài “Mạng chia sẻ kho bãi” đã được hoàn thành thành công, chứng minh được tính khả thi của việc ứng dụng công nghệ phi tập trung vào lĩnh vực logistics và quản lý tài sản. Dự án đã xây dựng một hệ thống DApp Hybrid hoàn chỉnh, với Smart Contract Solidity đóng vai trò là lõi nghiệp vụ, quản lý các giao dịch cốt lõi như đăng ký kho bãi, tạo hợp đồng thuê và đảm bảo tính minh bạch và an toàn tuyệt đối. Thành tựu nổi bật của dự án là việc triển khai thành công cơ chế Persistent Blockchain và dịch vụ Auto Synchronization, giúp hệ thống vừa tận dụng được tính bất biến của Blockchain, vừa đạt được tốc độ truy vấn cao của cơ sở dữ liệu truyền thống, giải quyết hiệu quả bài toán đồng bộ dữ liệu trong môi trường DApp.

Mặc dù đạt được các mục tiêu về mặt chức năng và kiến trúc, dự án hiện vẫn đang hoạt động trên môi trường phát triển cục bộ. Hướng phát triển trong tương lai sẽ tập trung vào việc đưa Smart Contract lên các mạng thử nghiệm công khai (Testnet), tối ưu hóa chi phí Gas và mở rộng chức năng quản lý tài sản bằng cách sử dụng tiêu chuẩn NFT (ERC-721/1155) để mã hóa quyền sở hữu kho bãi. Những cải tiến này sẽ nâng cao tính thực tiễn, khả năng mở rộng và hiệu suất của nền tảng, tiến tới một giải pháp chia sẻ kho bãi hoàn toàn phi tập trung và thương mại hóa được.

## **DANH MỤC TÀI LIỆU THAM KHẢO**

1. Slide bài giảng Lập Trình Blockchain
2. Giáo trình môn học Lập Trình Blockchain