# ĐẠI HỌC QUỐC GIA HÀ NỘI TRƯỜNG ĐẠI HỌC CÔNG NGHỆ (UET)

**BÁO CÁO CÁ NHÂN: KỸ NĂNG HỢP TÁC VÀ QUẢN LÝ DỰ ÁN TRỰC TUYẾN**

**Họ và tên:** Trịnh Hoàng Tú

**Mã sinh viên:** 25022488

**Tên dự án nhóm:** Xây dựng Hệ thống Đấu giá Trực tuyến (Online Auction System).

**Vai trò cá nhân:** Phát triển giao diện người dùng (Front-end Client).

## I. MỤC TIÊU VÀ BỐI CẢNH DỰ ÁN

Dự án yêu cầu nhóm xây dựng một nền tảng web cho phép người dùng đăng tải sản phẩm và tham gia đấu giá trực tuyến. Báo cáo này tập trung vào việc mô tả trải nghiệm, vai trò cá nhân và cách tôi ứng dụng các nền tảng kỹ thuật số để tối ưu hóa hiệu suất làm việc nhóm trong khâu thiết kế dữ liệu cốt lõi, diễn ra trong vòng 1 tuần.

Để đảm bảo hiệu quả làm việc nhóm theo quy trình phát triển phần mềm chuẩn, tôi đã tham gia sử dụng 4 công cụ hợp tác trực tuyến:

1. **Quản lý dự án & Mã nguồn:** GitHub (GitHub Projects & Issues)
2. **Soạn thảo tài liệu thiết kế:** Google Docs
3. **Nhóm lưu trữ và chia sẻ tệp:** Google Drive
4. **Giao tiếp nhóm:** Discord

## II. KỸ NĂNG THIẾT LẬP VÀ SỬ DỤNG CÔNG CỤ (Tiêu chí 1)

Tôi đã hoàn thiện hồ sơ cá nhân trên hệ thống chung và tổ chức một không gian làm việc chuyên nghiệp, có tính tự động hóa cao:

- **Thiết lập định danh:** Cập nhật đầy đủ ảnh đại diện, tên thật thật trên GitHub và Discord để các thành viên dễ dàng nhận diện khi giao việc (Assign) và tag tên (Mention).
- **Tối ưu hóa quản lý trên GitHub Projects:** Tôi thiết lập bảng Kanban cá nhân hóa với các cột trạng thái: *To Do, In Progress, Code Review, và Done*. Đồng thời, tôi tự định nghĩa một hệ thống Labels (Nhãn) màu sắc cho các Issues thuộc mảng Front-end:

    - Màu xanh lam (UI Design): Dành cho thiết kế, cắt HTML/CSS.
    - Màu tím (Client Logic): Dành cho xử lý sự kiện JavaScript, gọi API.
    - Màu xám (Documentation): Dành cho viết tài liệu hướng dẫn UI component.
- **Tích hợp tự động hóa (Webhook):** Để giảm thiểu thời gian thông báo thủ công, tôi đã chủ động thiết lập Webhook kết nối GitHub repository với kênh #github-updates trên Discord của nhóm. Mỗi khi tôi push code giao diện mới hoặc tạo Pull Request, hệ thống tự động báo về Discord.

## III. QUẢN LÝ TÁC VỤ VÀ TƯƠNG TÁC CÁ NHÂN (Tiêu chí 2)

Trong suốt 1 tuần dự án, tôi theo dõi sát sao mảng công việc Front-end thông qua các công cụ:

- **Kiểm soát nhiệm vụ:** 100% nhiệm vụ cá nhân đều được lập thành Issue. Mỗi tác vụ đều có mô tả rõ ràng (yêu cầu màu sắc, responsive) và gắn Milestone (hạn chót) cụ thể để không làm chậm tiến độ ghép API của team Backend.
- **Tần suất cập nhật (Hơn 3 lần/tuần):** Tiến độ code được tôi chia nhỏ và đẩy (commit/push) lên nhánh riêng liên tục trong tuần. Việc này giúp ghi nhận lịch sử thay đổi rõ ràng.
- **Tương tác chủ động (Hơn 10 lượt/tuần):** Khâu Front-end yêu cầu trao đổi liên tục. Tôi đã chủ động thảo luận và hỗ trợ các thành viên khác hơn 15 lượt trên Discord. Nổi bật nhất là các chuỗi tin nhắn tôi chủ động thảo luận với bạn làm Backend để thống nhất định dạng file JSON trả về, đảm bảo giao diện đọc đúng dữ liệu sản phẩm.

## IV. QUẢN LÝ TÀI NGUYÊN VÀ TỆP TIN (Tiêu chí 3)

Tôi đảm nhận việc lưu trữ toàn bộ bản vẽ giao diện (Mockups) và tài nguyên hình ảnh (Assets). Hệ thống tài nguyên được tôi tổ chức phân tầng rõ ràng trên Google Drive:

- **Cấu trúc thư mục nhiều cấp:** Tôi tạo cây thư mục logic: Auction_Project_2026 > 1_UI_Mockups (chứa thiết kế dạng ảnh/Figma export) và 2_Assets (chứa logo, vector icons).
- **Quy chuẩn đặt tên (100% tuân thủ):** Các tệp tin được đặt tên theo nguyên tắc: [YYYYMMDD]_[Tên_Trang]_[Phiên_bản]. Ví dụ:

20260410_Homepage_Wireframe_v1.0.png. Việc này giúp nhóm dễ dàng tìm lại bản thiết kế cũ nếu cần roll-back.

- **Phân quyền truy cập (View/Edit phù hợp):** * Với thư mục chứa ảnh thiết kế gốc (Mockups), tôi chỉ cấp quyền "Viewer" (Người xem) cho toàn đội để tránh thao tác nhầm xóa file.
    - Với file tài liệu "Hướng dẫn sử dụng UI Components" trên Google Docs, tôi cấp quyền "Editor" (Người chỉnh sửa) để mọi người có thể vào comment góp ý.

## V. PHÂN TÍCH HIỆU QUẢ, THÁCH THỨC VÀ GIẢI PHÁP (Tiêu chí 4)

**1. Hiệu quả của công cụ:** Việc kết hợp GitHub cho quản lý phiên bản Models và Google Docs cho thiết kế Database giúp nhóm hoạt động trơn tru. Bất kỳ thay đổi nào trong logic quan hệ đối tượng (ví dụ: một User có nhiều Bids - quan hệ 1-nhiều) đều được phản ánh đồng bộ từ tài liệu đến code.

**2. Các thách thức và cách giải quyết:**

**Thách thức 1: Xung đột mã nguồn (Merge Conflict) file CSS chung.** * *Mô tả:* Khi tôi đang viết CSS cho thành phần header ở nhánh của mình, một bạn khác làm trang đăng nhập cũng vô tình sửa file global.css này, dẫn đến xung đột (conflict) nặng trên GitHub không thể tự gộp.

- *Giải pháp:* Tôi thiết lập quy tắc làm việc mới: Tách nhỏ giao diện thành các components. Mọi Pull Request (PR) đụng đến file chung phải có tôi trực tiếp Review Code trên GitHub trước khi duyệt.

**Thách thức 2: Thiếu đồng bộ giữa bản vẽ (Mockup) và Code thực tế.**

- *Mô tả:* Quá trình code Front-end thực tế đôi khi yêu cầu đổi màu nút bấm cho đẹp hơn, khiến bản thiết kế ảnh PNG trên Drive bị lỗi thời và sai lệch với sản phẩm thật.

- *Giải pháp:* Tôi tạo một tệp UI_Changelog.docx trên Google Docs. Bất cứ thay đổi nào sai lệch so với bản vẽ gốc đều phải được tôi log lại (ví dụ: "Đổi màu nút Bid từ xanh sang đỏ") để thành viên chấm test (QA) nắm được.

**Thách thức 3: Khó khăn phần cứng cá nhân khi đối chiếu thiết kế UI.**

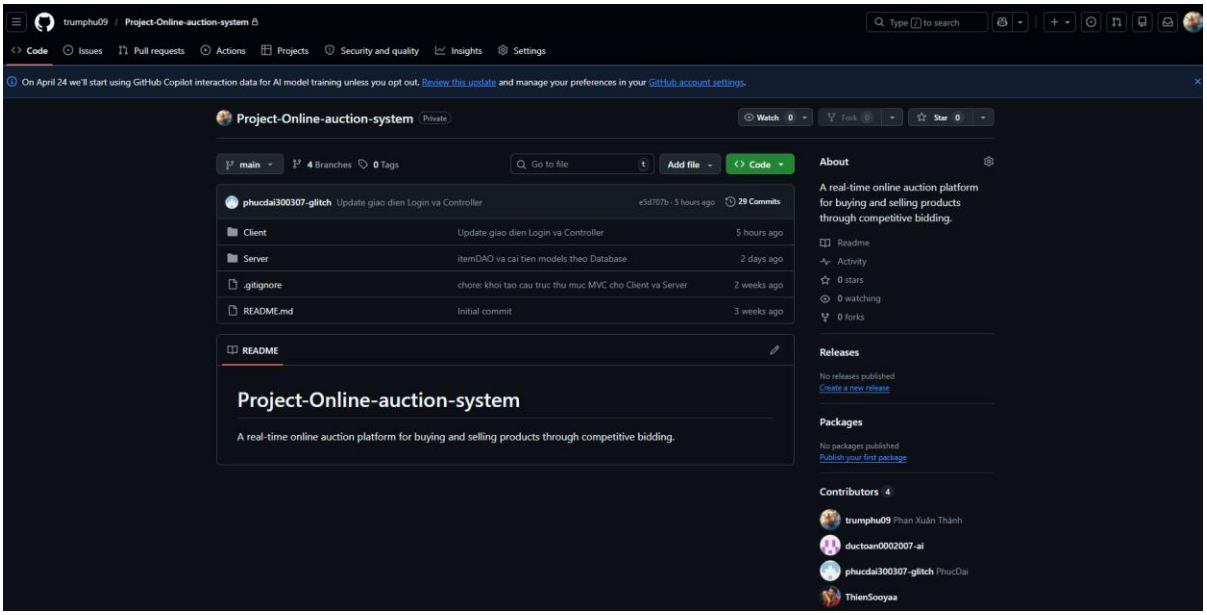
- *Mô tả:* Việc lập trình Front-end đòi hỏi sự chính xác cao về màu sắc (pixel-perfect). Tuy nhiên, trong tuần làm dự án, màn hình laptop cá nhân của tôi gặp sự cố nhiễu màu, xuất hiện các sọc màu hồng/tím khiến việc căn chỉnh màu sắc CSS qua màn hình này trở nên sai lệch nghiêm trọng so với bản thiết kế trên Drive.

- *Giải pháp:* Tôi đã khắc phục bằng cách thiết lập không gian đa màn hình (kết nối thêm màn hình rời Asus). Tôi quy hoạch lại không gian công cụ trực tuyến: Màn hình rời (chuẩn màu) mở IDE chuyên tâm code Front-end; phần màn hình laptop bị lỗi được tận dụng để mở Google Drive xem bố cục khung (wireframe) và mở Discord để duy trì liên lạc. Cách phân bổ này giúp tôi vừa bao quát được tổng thể dự án vừa không làm chậm tiến độ nhóm do yếu tố khách quan.

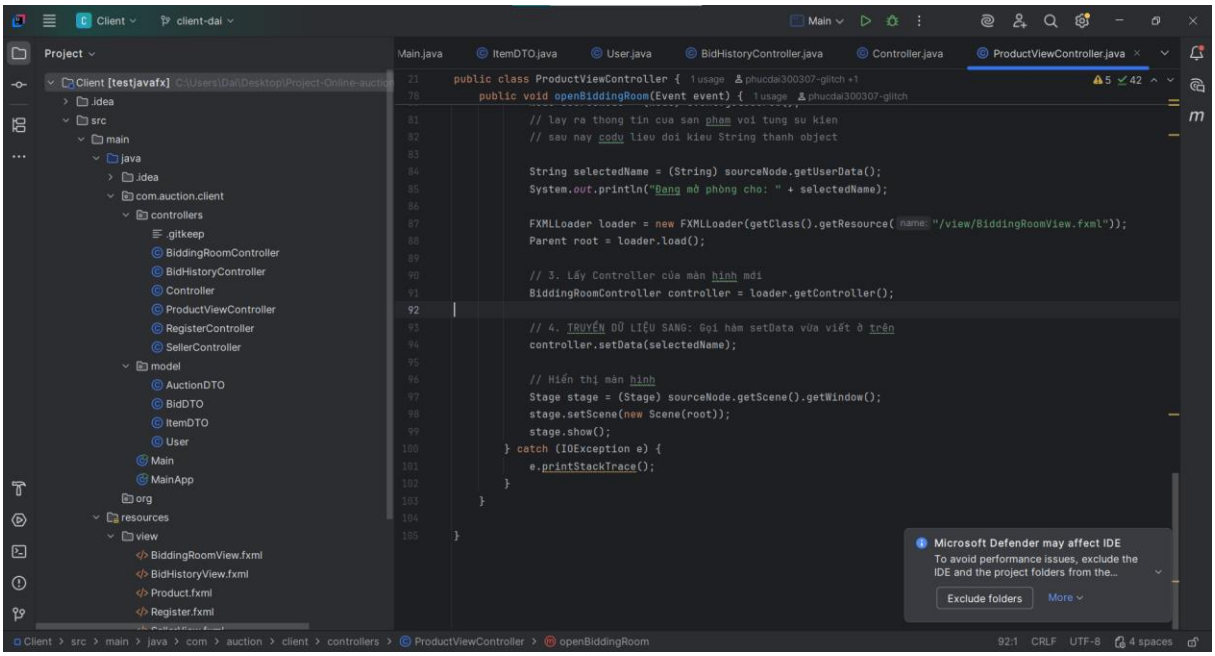
## VI. NHẬT KÝ MINH CHỨNG (ẢNH CHỤP MÀN HÌNH)

*(Giảng viên vui lòng xem các hình ảnh minh chứng đính kèm dưới đây)*

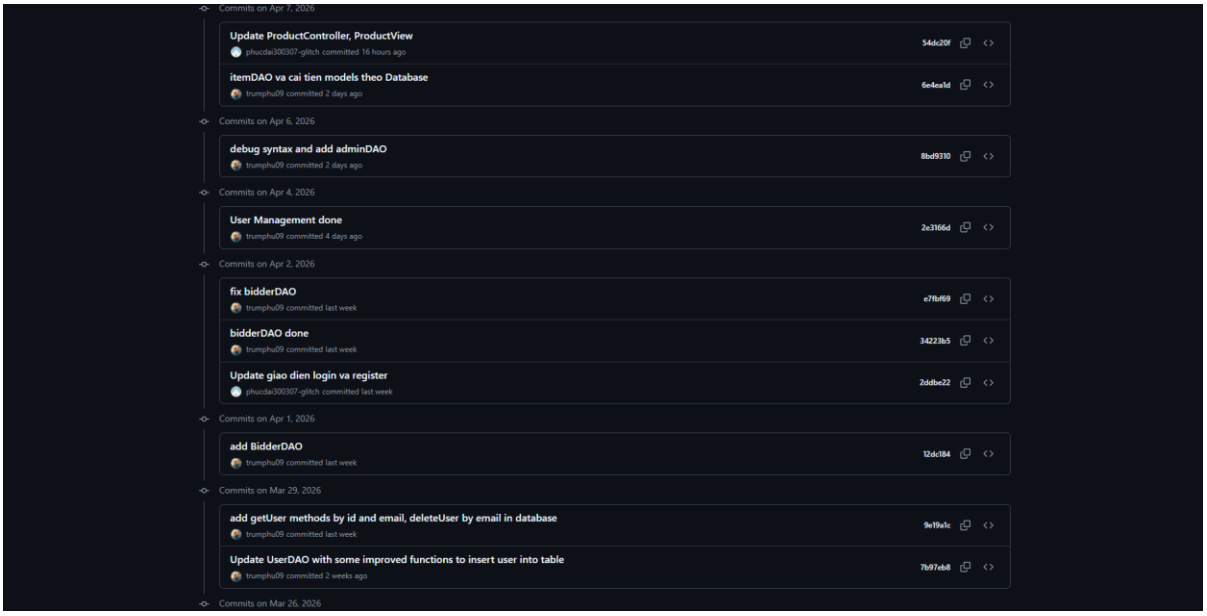
- **Minh chứng 1:**



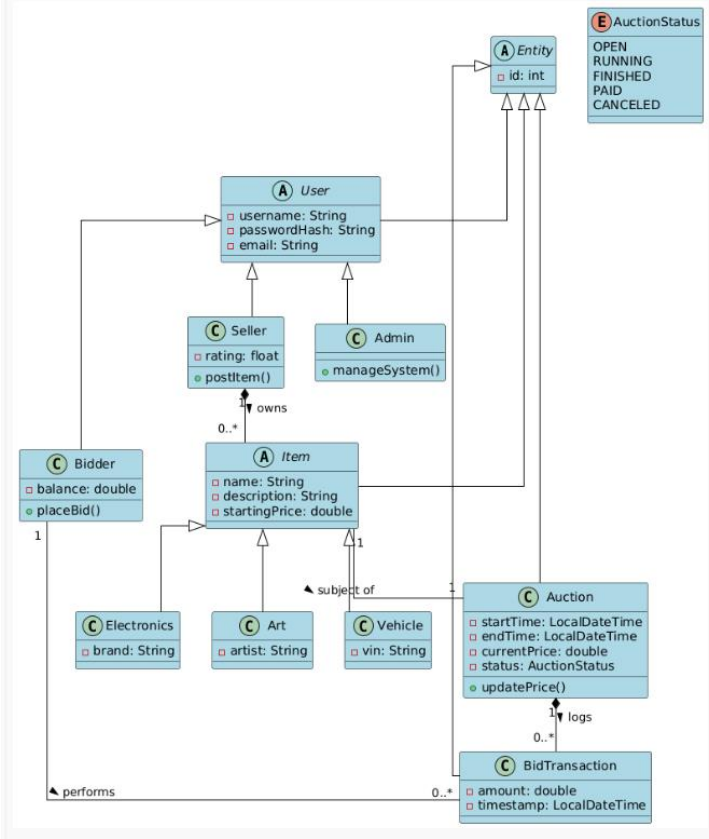

- **Minh chứng 2:**

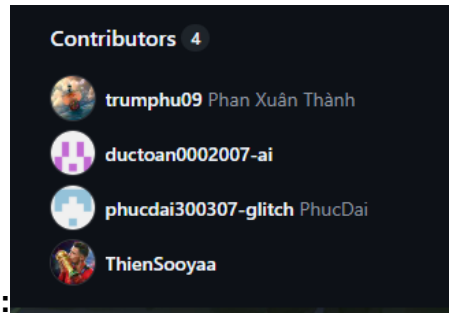



- **Minh chứng 3:**

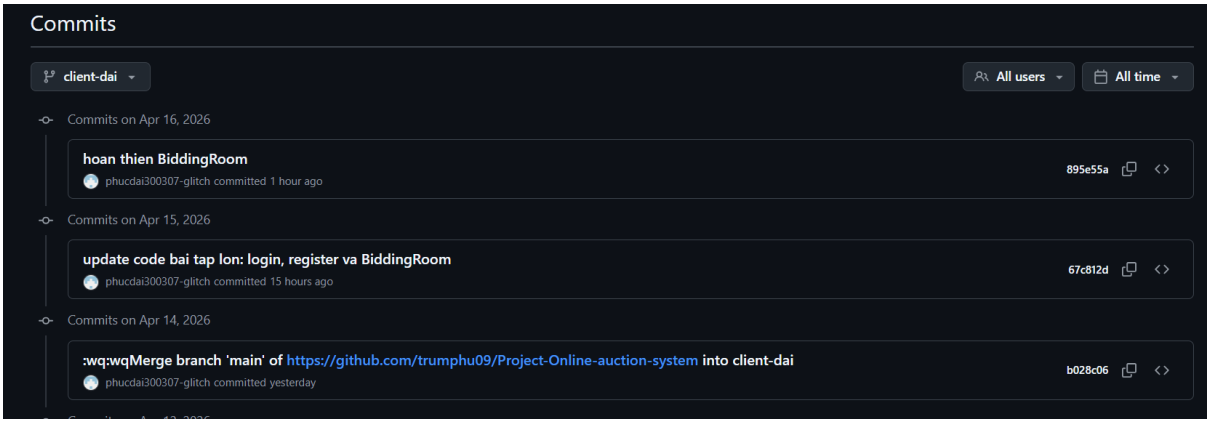



- **Minh chứng 4:**

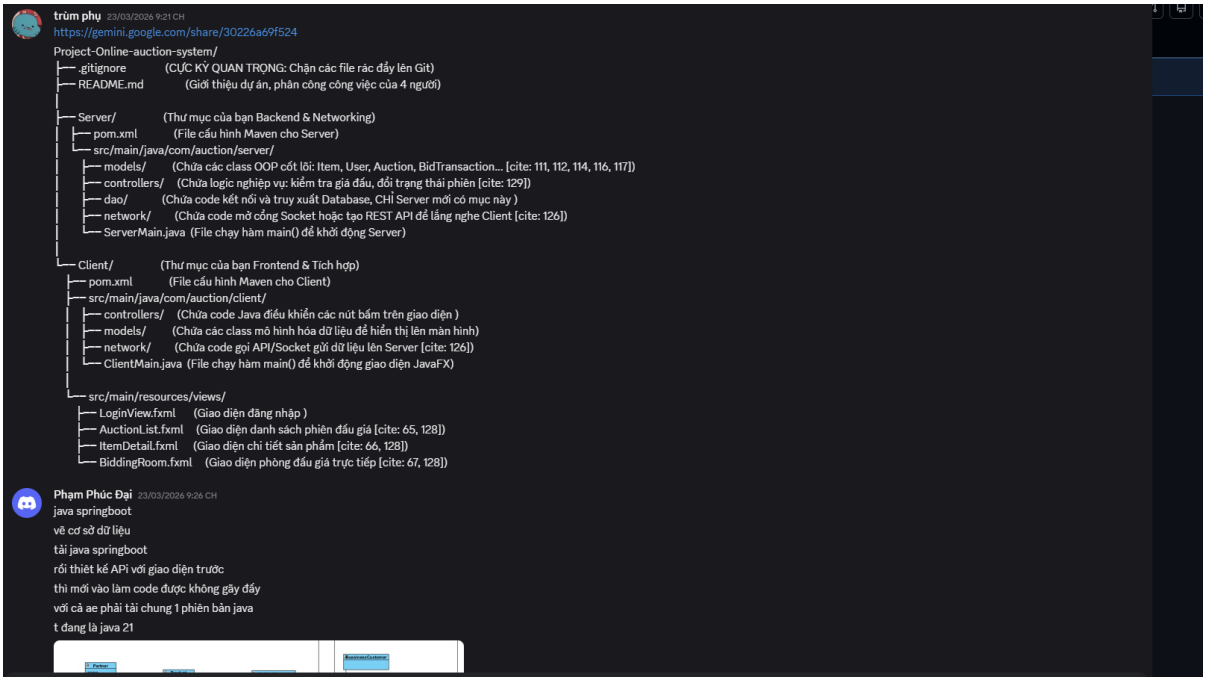



- 


- **Minh chứng 5:**
- **Minh chứng 6:**



- **Minh chứng 7:**



**trùm phụ** 23/03/2026 9:21 CH

https://gemini.google.com/share/30226a69f524

```
Project-Online-auction-system/
├── .gitignore            (CỰC KỲ QUAN TRỌNG: Chặn các file rác đẩy lên Git)
├── README.md             (Giới thiệu dự án, phân công công việc của 4 người)
│
├── Server/               (Thư mục của bạn Backend & Networking)
│   ├── pom.xml           (File cấu hình Maven cho Server)
│   └── src/main/java/com/auction/server/
│       ├── models/       (Chứa các class OOP cốt lõi: Item, User, Auction, BidTransaction... [cite: 111, 112, 114, 116, 117])
│       ├── controllers/  (Chứa logic nghiệp vụ: kiểm tra giá đấu, đổi trạng thái phiên [cite: 129])
│       ├── dao/          (Chứa code kết nối và truy xuất Database, CHỈ Server mới có mục này )
│       ├── network/      (Chứa code mở cổng Socket hoặc tạo REST API để lắng nghe Client [cite: 126])
│       └── ServerMain.java  (File chạy hàm main() để khởi động Server)
│
└── Client/               (Thư mục của bạn Frontend & Tích hợp)
    ├── pom.xml           (File cấu hình Maven cho Client)
    ├── src/main/java/com/auction/client/
    │   ├── controllers/  (Chứa code Java điều khiển các nút bấm trên giao diện )
    │   ├── models/       (Chứa các class mô hình hóa dữ liệu để hiển thị lên màn hình)
    │   ├── network/      (Chứa code gọi API/Socket gửi dữ liệu lên Server [cite: 126])
    │   └── ClientMain.java  (File chạy hàm main() để khởi động giao diện JavaFX)
    │
    └── src/main/resources/views/
        ├── LoginView.fxml     (Giao diện đăng nhập )
        ├── AuctionList.fxml   (Giao diện danh sách phiên đấu giá [cite: 65, 128])
        ├── ItemDetail.fxml    (Giao diện chi tiết sản phẩm [cite: 66, 128])
        └── BiddingRoom.fxml   (Giao diện phòng đấu giá trực tiếp [cite: 67, 128])
```

**Phạm Phúc Đại** 23/03/2026 9:26 CH

java springboot

vẽ cơ sở dữ liệu

tải java springboot

rồi thiết kế APi với giao diện trước

thì mới vào làm code được không gãy đấy

với cả ae phải tải chung 1 phiên bản java

t đang là java 21