**TỔNG KẾT HỆ THỐNG WEB — USER / ADMIN MANAGEMENT & PAYMENT FLOW**

**⚙️ 1. CẤU TRÚC THƯ MỤC DỰ ÁN**

**📁 back\_end/**

Chứa toàn bộ mã nguồn Node.js + Express + MongoDB + Email SMTP.

back\_end/

│

├── .env # Biến môi trường (DB, JWT, SMTP)

├── index.js # File chính khởi động server

├── package.json / package-lock.json # Cấu hình thư viện npm

├── test\_mail.js # Script kiểm tra gửi mail SMTP

│

├── admin\_model/ # Controllers xử lý logic admin & user

│ ├── authController.js # Đăng ký, đăng nhập, refresh token, logout

│ ├── middlewareController.js # Xác thực JWT & quyền admin

│ └── userController.js # CRUD user, duyệt tài khoản, gửi mail

│

├── routes/ # Định nghĩa các endpoint API

│ ├── auth.js # /v1/auth (đăng nhập, đăng ký, logout)

│ ├── user.js # /v1/user (user info, admin delete)

│ ├── order.js # /v1/order (quản lý đơn hàng & thanh toán)

│

├── user\_model/ # Mongoose models

│ └── User.js # Định nghĩa schema người dùng

│

├── models/

│ └── Order.js # Định nghĩa schema đơn hàng (thanh toán)

│

└── utils/

└── mailer.js # Hàm gửi mail (dùng nodemailer)

**🌐 front\_end/**

Chứa giao diện người dùng (HTML + JS thuần).

front\_end/

│

├── admin.html # Giao diện chính admin dashboard

├── user\_management.html # Quản lý user, duyệt tài khoản

├── payment\_management.html # Quản lý đơn hàng chờ xác nhận thanh toán

├── user\_dashboard.html # Trang user: hiển thị sản phẩm, mua hàng

├── payment.html # Trang thanh toán (QR, xác nhận)

├── login.html # Đăng nhập

├── register.html # Đăng ký

│

└── js/

└── auth.js # Xử lý login, register, token

**🔄 2. WORKFLOW — LUỒNG HOẠT ĐỘNG HỆ THỐNG**

sequenceDiagram

participant User

participant FrontEnd

participant BackEnd

participant MongoDB

participant Admin

participant MailServer

User->>FrontEnd: Mở trang Register

FrontEnd->>BackEnd: POST /v1/auth/register (username, email, ...)

BackEnd->>MongoDB: Lưu user (status="pending")

BackEnd->>MailServer: Gửi email cho Admin (có user mới chờ duyệt)

MailServer-->>Admin: Email thông báo "Có user mới cần duyệt"

Admin->>FrontEnd: Mở trang user\_management.html

FrontEnd->>BackEnd: GET /v1/user/list (users pending)

Admin->>BackEnd: POST /v1/user/accept/:id

BackEnd->>MongoDB: Cập nhật user (status="active")

BackEnd->>MailServer: Gửi email thông báo kích hoạt cho user

MailServer-->>User: Email "Tài khoản đã được duyệt"

User->>FrontEnd: Login

FrontEnd->>BackEnd: POST /v1/auth/login

BackEnd->>MongoDB: Kiểm tra thông tin, tạo JWT token

BackEnd-->>FrontEnd: Trả token + info user

FrontEnd->>User: Mở user\_dashboard.html

User->>FrontEnd: Chọn sản phẩm → Thanh toán

FrontEnd->>BackEnd: POST /v1/order/new

BackEnd->>MongoDB: Lưu đơn hàng (status="pending\_payment")

User->>FrontEnd: Xác nhận đã thanh toán

FrontEnd->>BackEnd: PUT /v1/order/confirm/:id

BackEnd->>MailServer: Gửi mail báo admin có đơn chờ duyệt

MailServer-->>Admin: Email "Có đơn hàng chờ xác nhận"

Admin->>FrontEnd: Mở payment\_management.html

Admin->>BackEnd: PUT /v1/order/approve/:id (hoặc reject)

BackEnd->>MongoDB: Cập nhật trạng thái

BackEnd->>MailServer: Gửi mail OTP (nếu approve)

MailServer-->>User: Email "Đơn hàng xác nhận, OTP: ####"

**🔑 3. CẤU HÌNH**

Trong**.env**

# MongoDB connection

MONGODB\_URL=mongodb+srv://username:password@cluster0.mongodb.net/?retryWrites=true&w=majority

# JWT secrets

SECRETKEY=your\_jwt\_secret

REFRESHTOKEN=your\_refresh\_token\_secret

# SMTP (Gmail)

SMTP\_HOST=smtp.gmail.com **-> Host server của smtp là google.com**

SMTP\_PORT=465 **-> Nếu port ảo thì là 587 (Ethereal)**

SMTP\_USER=Sender@gmail.com **-> Tài khoản dùng để gửi mail trên server**

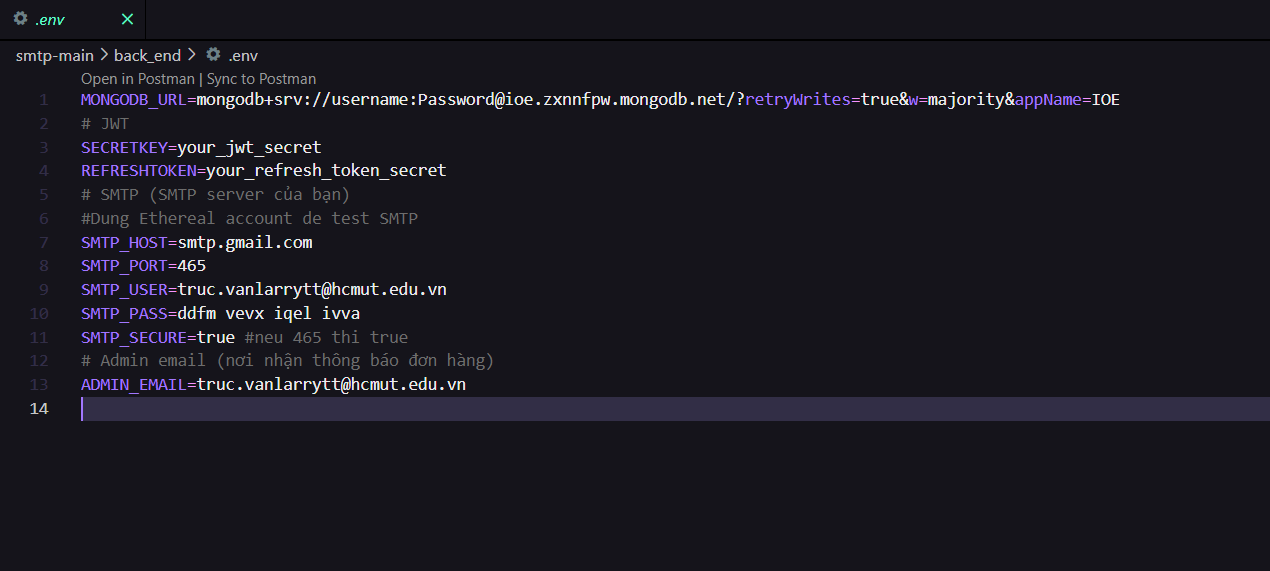
SMTP\_PASS=your\_16\_digit\_app\_password **-> Password được tạo ra sau các bước hướng dẫn bên dưới**

SMTP\_SECURE=true **-> false nếu port là 587**

# Admin email

ADMIN\_EMAIL=truc.vanlarrytt@hcmut.edu.vn **-> email admin để gửi mail thông báo user mới**

⚠️ SMTP\_PASS là **App Password 16 ký tự**, không phải mật khẩu Gmail thật.



App Password được tạo theo flow sau:

**🔐 Cách tạo App Password (chỉ mất 1 phút)**

1.Truy cập: https://myaccount.google.com/security  
2️.Bật **2-Step Verification** (xác minh 2 bước)  
3️.Sau khi bật, sẽ thấy mục **App passwords**  
4️.Chọn:

* App: **Mail**
* Device: **Other (Custom)** → nhập NodeMailer
* Bấm **Generate**

5️.Bạn sẽ thấy mã 16 ký tự, ví dụ:

ddfm vevx iqel ivva

(chính là chuỗi bạn đang có)

6️.Dán chuỗi đó vào .env như sau:

SMTP\_PASS=ddfmvevxiqelivva

❗Không có khoảng trắng — loại bỏ hết cách dòng và khoảng giữa các nhóm ký tự.

**🚀 4. HƯỚNG DẪN CHẠY HỆ THỐNG**

**🧩 Bước 1: Cài đặt backend**

cd back\_end

npm install

Các thư viện sẽ tự động cài:

* express
* mongoose
* nodemailer
* bcryptjs
* jsonwebtoken
* cookie-parser
* cors
* dotenv
* morgan

**🧩 Bước 2: Chạy server**

node index.js

Khi thành công:

CONNECTED TO MONGODB

Server is running

Server hoạt động tại:  
👉 <http://localhost:8000>

Ví dụ: http://localhost:8000/login.html

**🧩 Bước 3: Chạy test mail (tùy chọn)**

Test mail để kiểm tra SMTP Server đề phòng trường hợp gặp lỗi hoặc nghẽn server.

Ngoài ra kiểm tra SMTP để đảm bảo tính chắn chắn cho các bugs khác.

node test\_mail.js

→ Nếu “Mail sent successfully”, nghĩa là SMTP OK.

**🧩 Bước 4: Mở giao diện**

1. Mở thư mục front\_end/
2. Mở file login.html hoặc register.html bằng trình duyệt

Backend đã phục vụ file tĩnh, nên bạn có thể truy cập qua:  
👉 <http://localhost:8000/login.html>

**🧪 5. HƯỚNG DẪN TEST TOÀN BỘ LUỒNG**

**NOTE:**

* **Tài khoản admin đã tạo trước:**

*Email:* [*tester@gmail.com*](mailto:tester@gmail.com)

*Password: Password*

* **Tài khoản user đã tạo trước:**

*Email:* [*user@gmail.com*](mailto:user@gmail.com)

*Password: Password*

**🧍‍♂️ User luồng**

1. Mở register.html → đăng ký tài khoản mới
2. Email admin (ADMIN\_EMAIL) nhận được thông báo chờ duyệt
3. User chưa thể đăng nhập (vì status = pending)

**👨‍💼 Admin luồng**

1. Mở user\_management.html
2. Thấy user mới “Chờ duyệt”
3. Click “Accept” → tài khoản được kích hoạt
4. User nhận email “Tài khoản của bạn đã được duyệt”

**🛍 User mua hàng**

1. User đăng nhập → mở user\_dashboard.html
2. Chọn sản phẩm → thanh toán → mở payment.html
3. Quét QR, nhấn “Xác nhận đã thanh toán”
4. Email admin nhận “Có đơn hàng chờ xác nhận”

**💰 Admin xác nhận thanh toán**

1. Mở payment\_management.html
2. Click “Xác nhận” → hệ thống gửi email OTP cho user
3. Nếu “Từ chối” → email báo đơn hàng bị hủy
4. Trạng thái đơn hàng cập nhật lại trên cả giao diện user & admin

**📤 6. CÁC MAIL GỬI TRONG HỆ THỐNG**

| **Tình huống** | **Người nhận** | **Tiêu đề mail** | **Nội dung** |
| --- | --- | --- | --- |
| User đăng ký | Admin | 📨 “User mới chờ duyệt” | Thông tin user đăng ký |
| Admin duyệt user | User | “Tài khoản đã kích hoạt” | Lời chào & thông báo đăng nhập |
| User xác nhận thanh toán | Admin | “Đơn hàng chờ xác nhận” | Thông tin đơn & user |
| Admin xác nhận đơn hàng | User | “Mã OTP xác nhận thanh toán” | Gửi mã OTP |
| Admin từ chối đơn hàng | User | “Đơn hàng bị từ chối” | Lý do & hướng xử lý |

**🧱 7. MONGODB — CÁC COLLECTION**

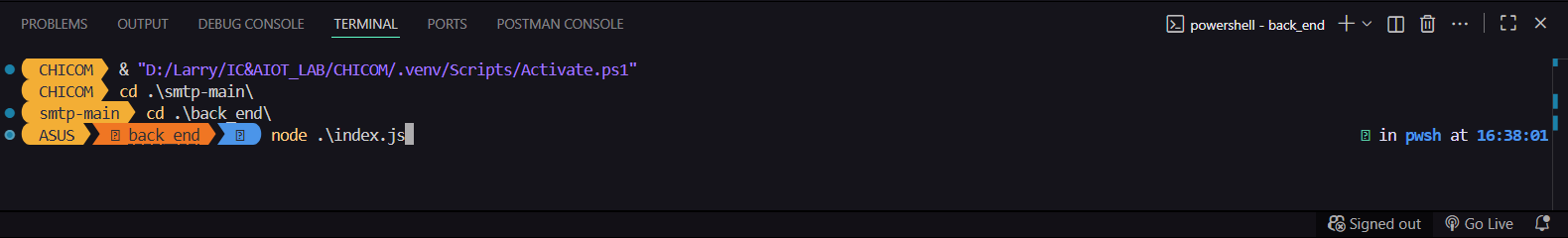
| **Collection** | **Trường dữ liệu chính** | **Mục đích** |
| --- | --- | --- |
| users | username, email, password, admin, status | Lưu thông tin user |
| orders | userId, product, total, status | Lưu đơn hàng, trạng thái thanh toán |

**🧰 8. MẸO DEBUG NHANH**

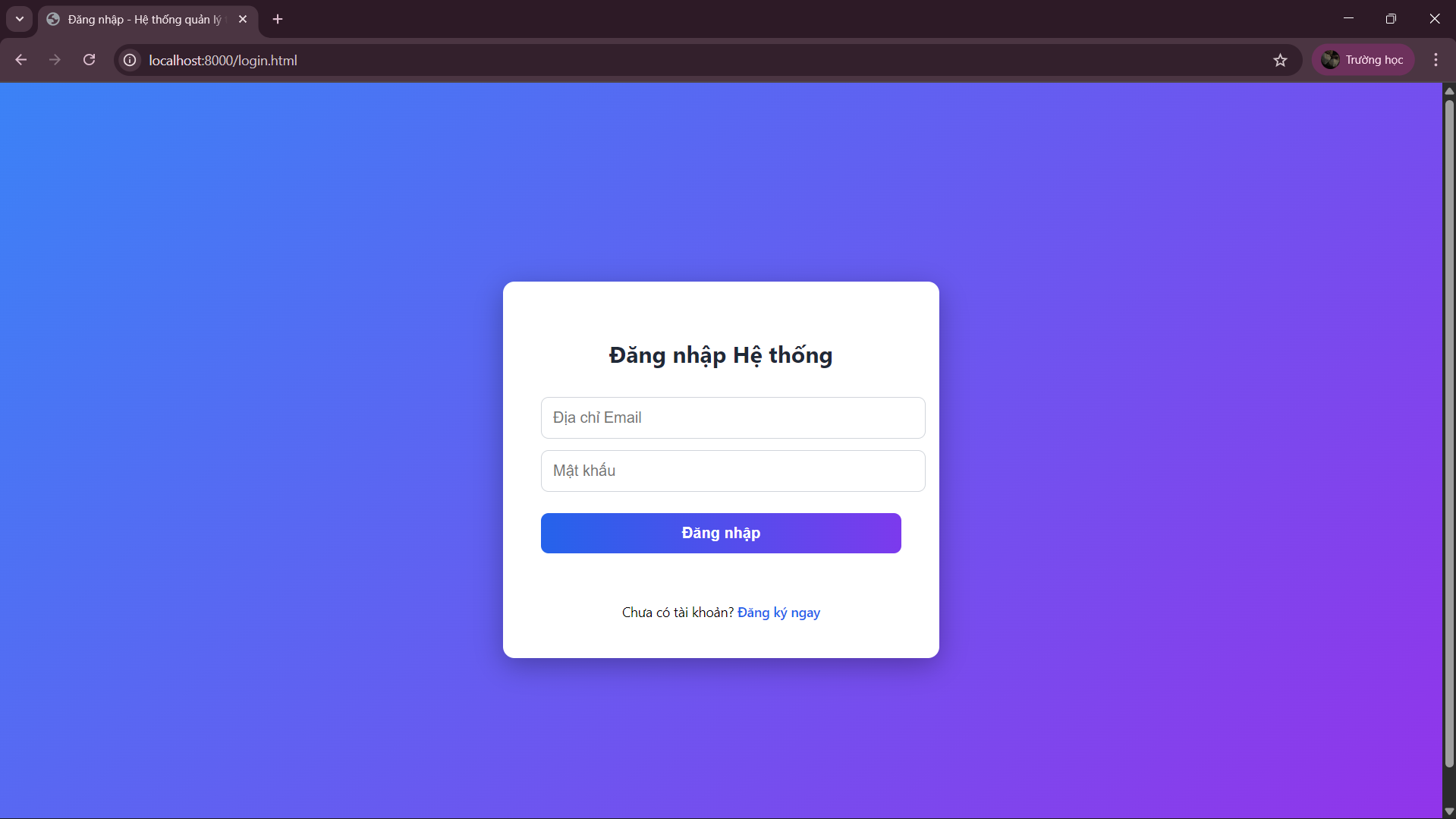
| **Vấn đề** | **Cách xử lý** |
| --- | --- |
| Không gửi được mail | Kiểm tra .env SMTP, App Password |
| Lỗi CORS | Đặt origin: "http://localhost:5500" trong index.js |
| Lỗi JWT hết hạn | Dùng API /v1/auth/refresh để lấy access token mới |
| Không truy cập được HTML | Kiểm tra express.static("../front\_end") đúng đường dẫn |

**QUY TRÌNH THAO TÁC TRÊN WEB**

Kiểm tra đường dẫn thư mục chứa back\_end

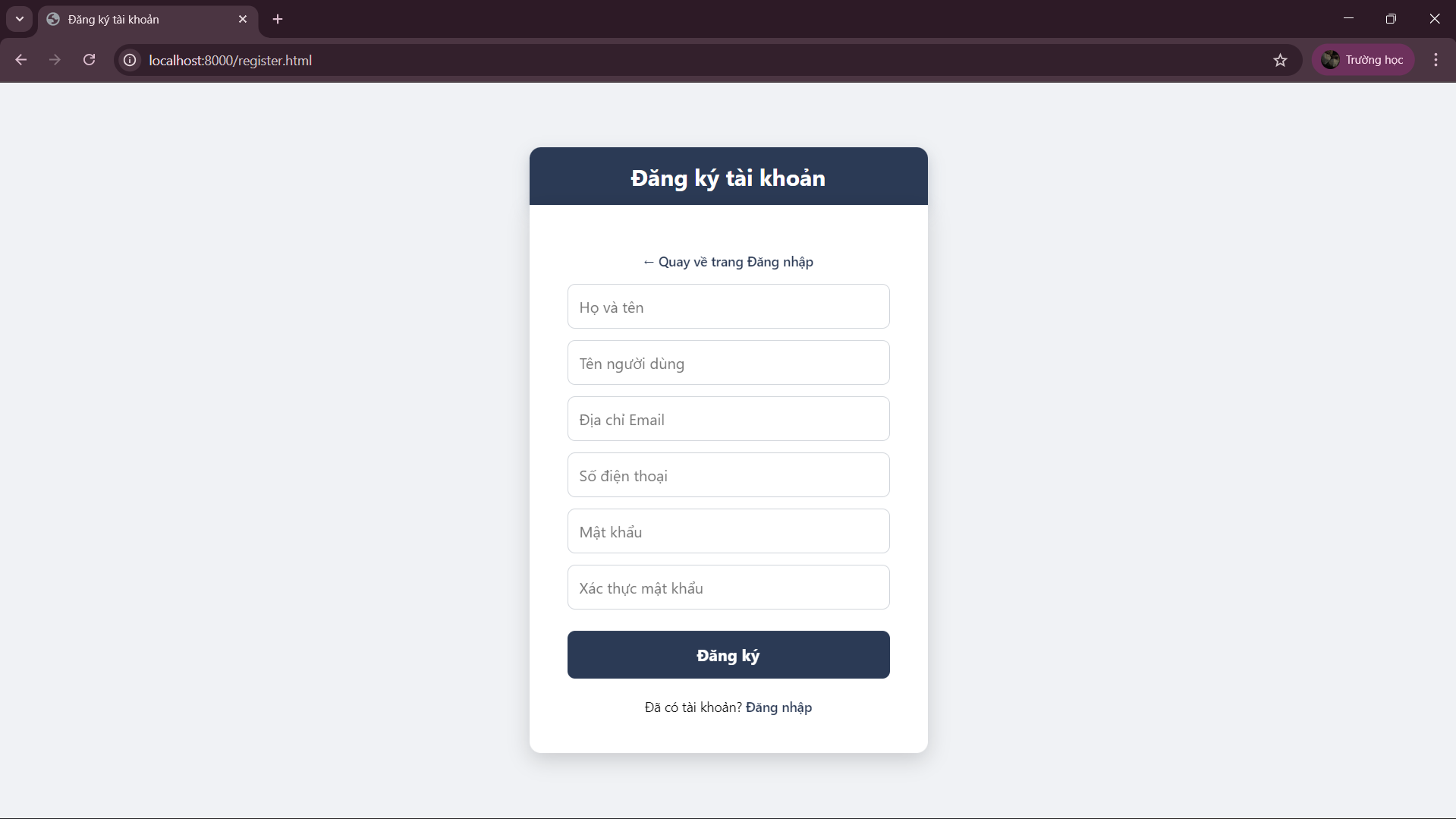


-Sau khi chạy localhost thì ta sẽ trực tiếp mở trang login.html



Đây chính là giao diện trang login.html để Đăng nhập

Nếu chưa có tài khoản, chọn Đăng ký ngay



Đây là giao diện đăng ký tài khoản, điền các field (bắt buộc).

TEST CASE:

* Nếu bỏ trống, báo lỗi
* Nếu nhập ‘Họ và tên’ :

+ dưới 4 ký tự, báo lỗi…

+ Trùng với tên có sẵn, báo lỗi

* Nếu nhập ‘Tên người dùng’:

+ dưới 4 ký tự, báo lỗi…

+ Trùng tên người khác, báo lỗi

* Nếu nhập địa chỉ email:

+ không đúng đuôi @domain, báo lỗi…

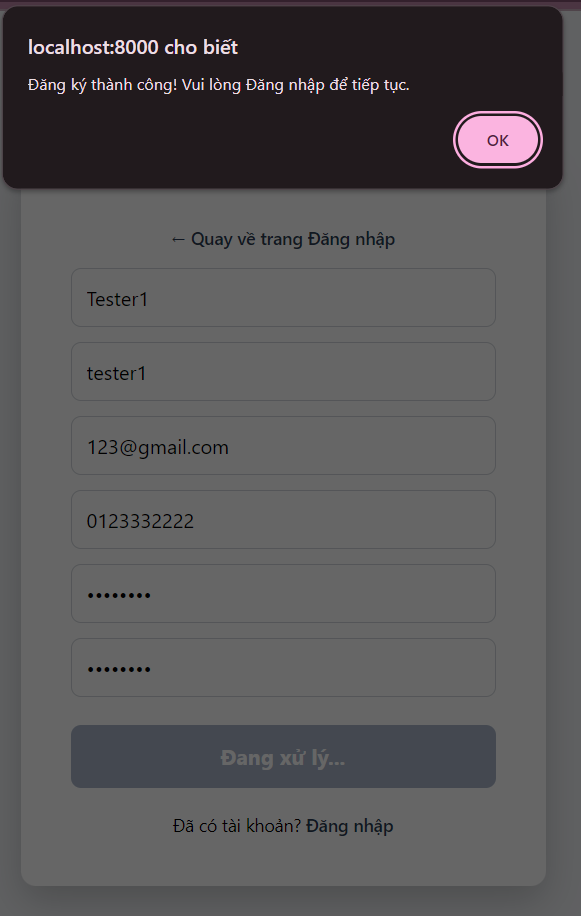
+ Trùng email có sẵn, báo lỗi

* Nếu nhập ‘Số điện thoại’:

+ dưới 10 số, báo lỗi…

+ Trùng số điện thoại, báo lỗi…

* Nếu nhập mật khẩu: dưới 6 ký tự, báo lỗi
* Confirm mật khẩu: nhập sai, báo lỗi.



Sau khi điền các fields và nhấn Đăng ký, web sẽ hiện thông báo Đăng ký thành công

**Tuy nhiên sau bước Đăng ký này, user mới vẫn chưa được đăng nhập vào mà phải đợi admin duyệt tài khoản thì mới được đăng nhập.**

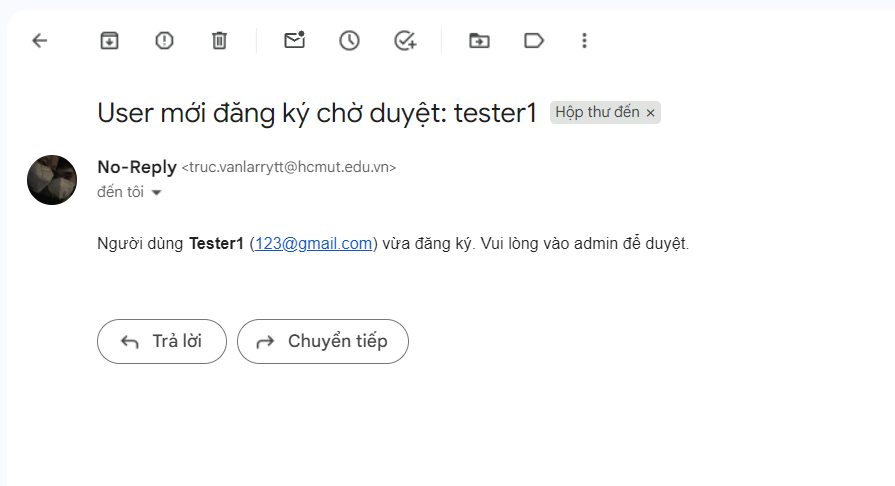
TEST: Nếu chưa được duyệt thì khi đăng nhập sẽ thế nào



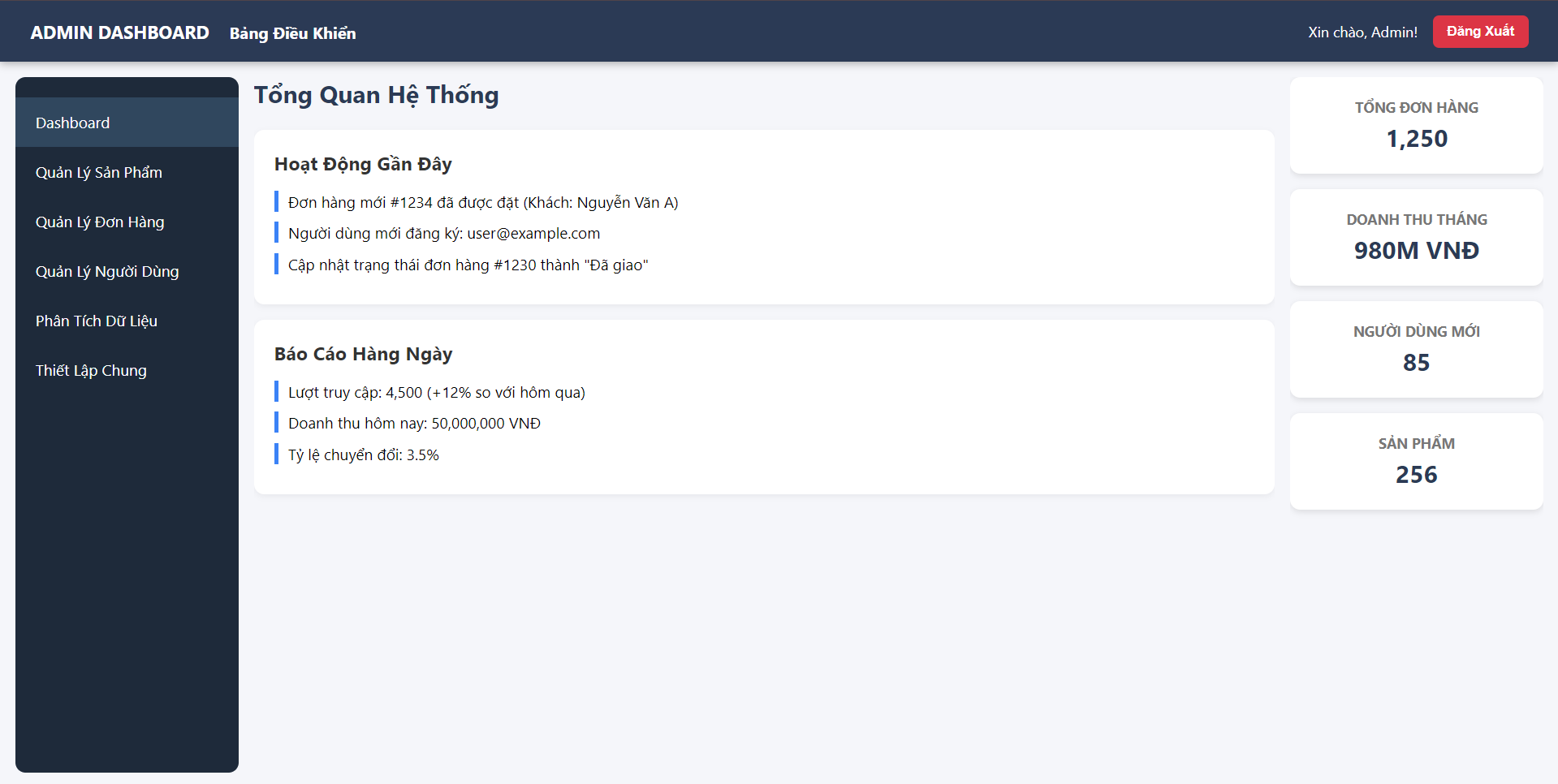
* Có thể thấy khi chưa được admin accept thì hệ thống sẽ không cho phép đăng nhập.

ADMIN:

Sau khi user mới đăng ký thì admin sẽ được nhận thông báo qua mail:

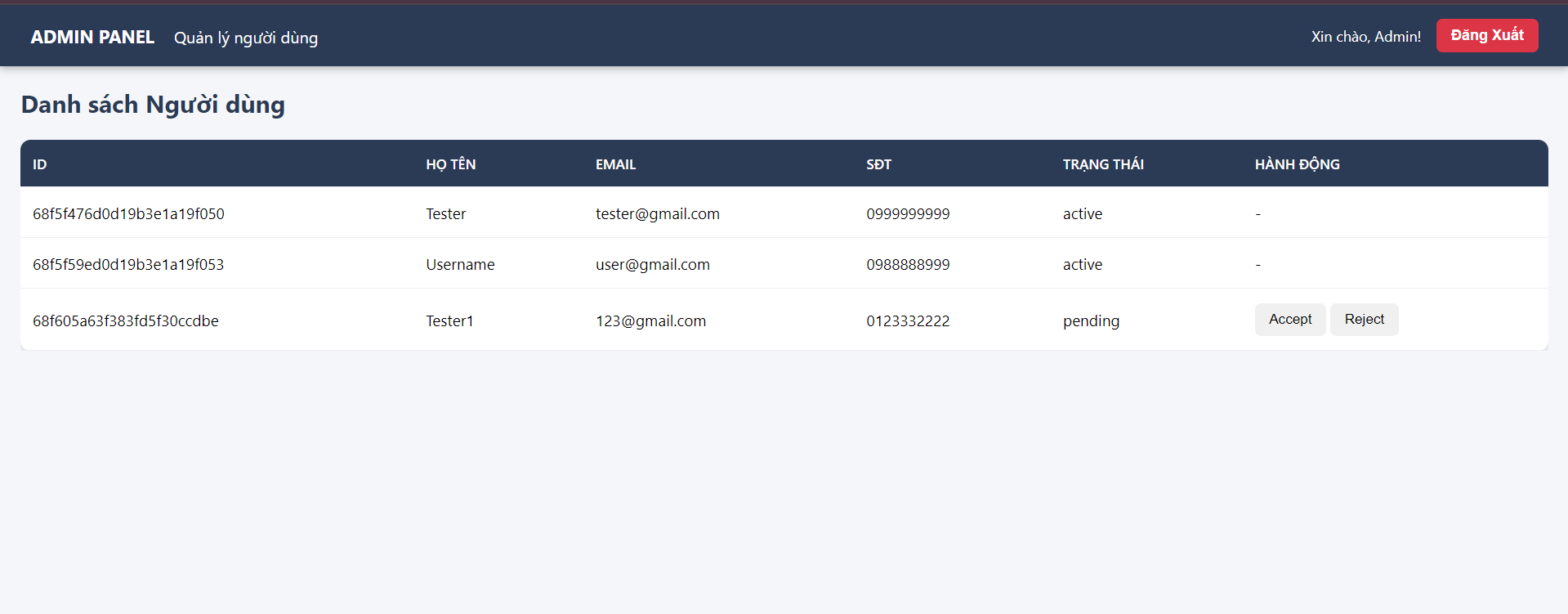


Với email admin được thiết lập sẵn, sau khi được nhận thông báo nêu rõ về user mới, ta Đăng nhập vào với tài khoản admin.



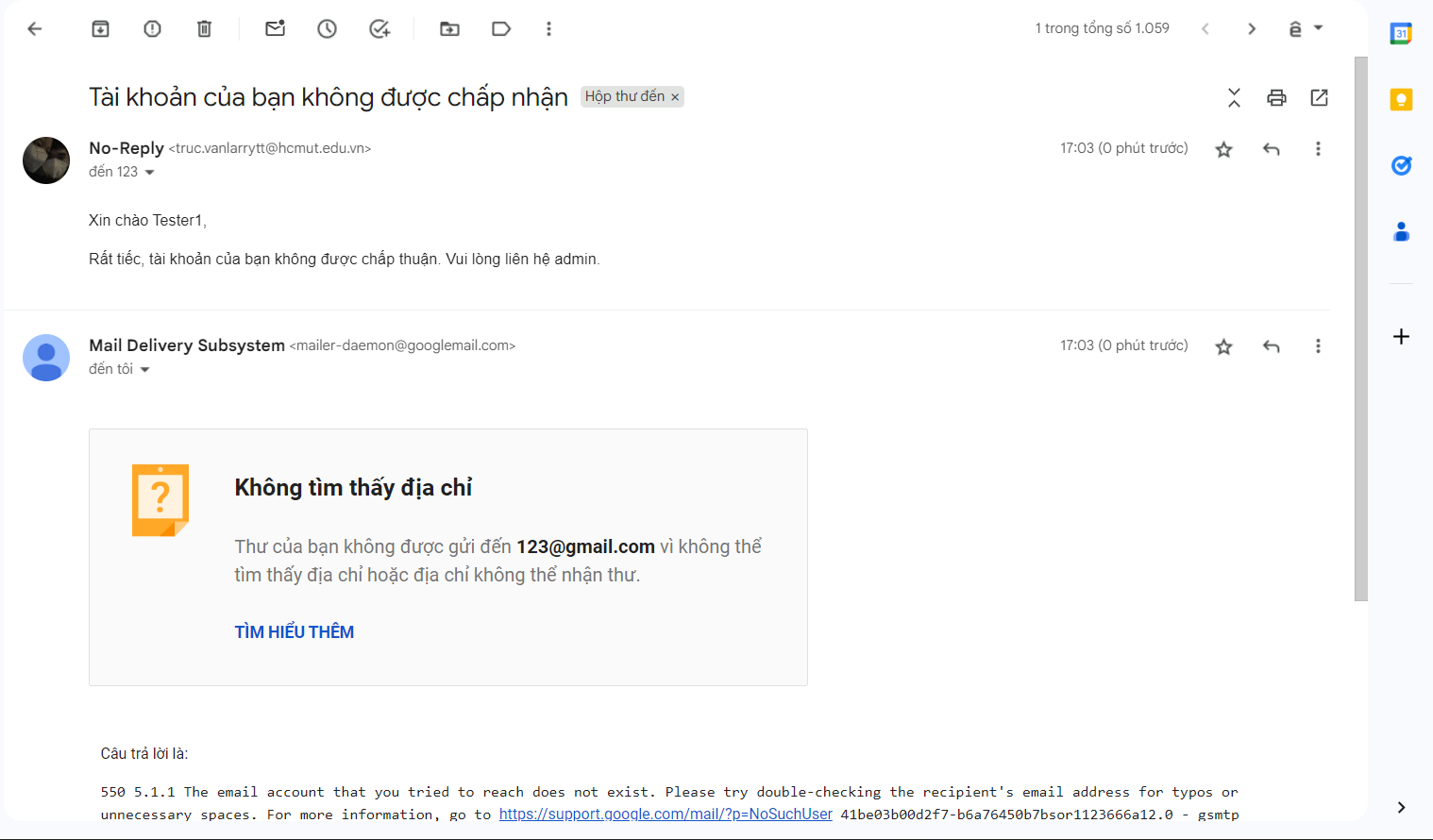
Giao diện tài khoản admin.html sẽ khác với giao diện của user\_dashboard.html, có các mục Quản lý.

Chọn Quản lý người dùng

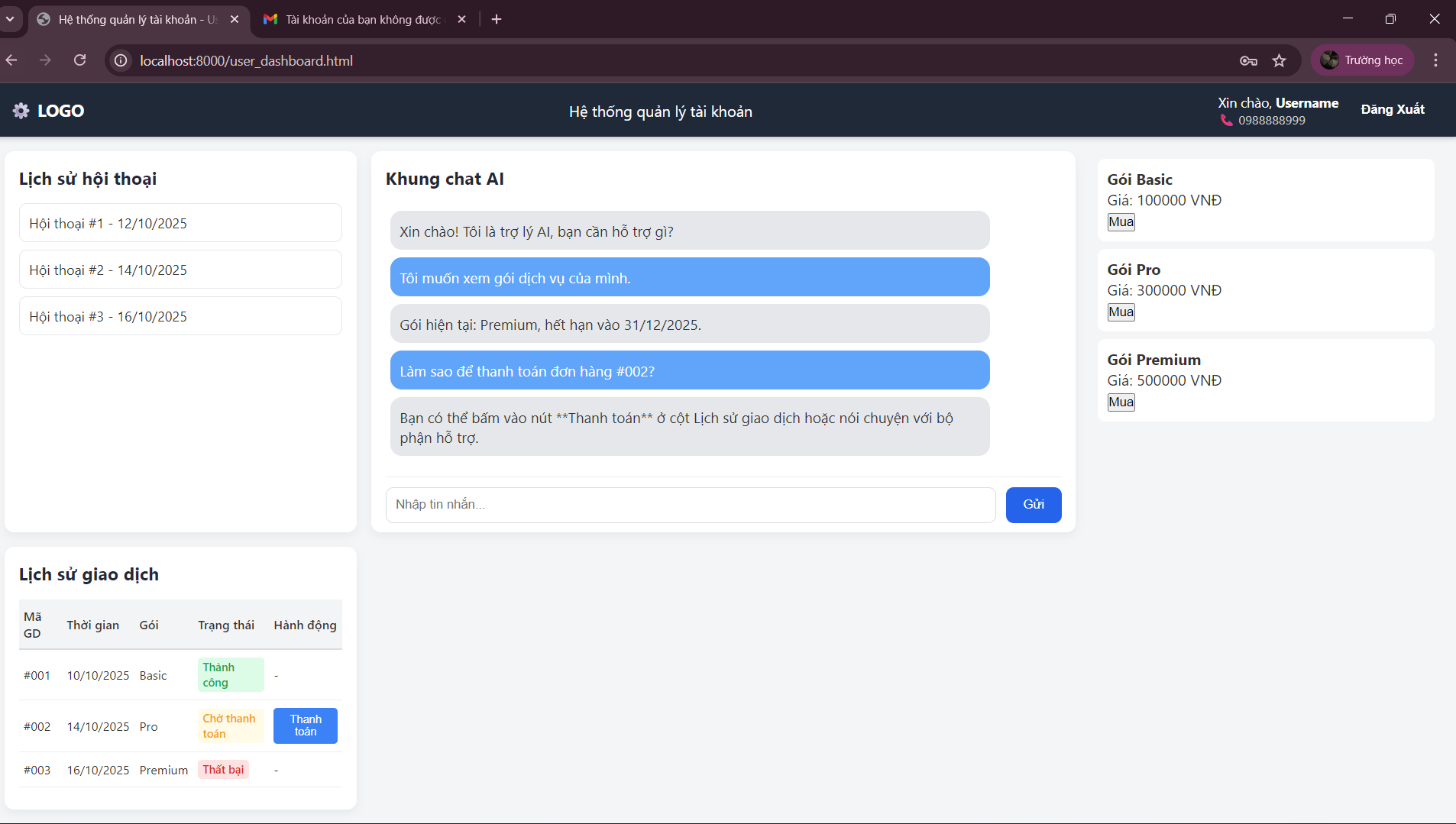


Ở đây ta sẽ thấy user mới đang hiện thông báo trạng thái là “pending” và cần admin chọn các tác vụ là Accept hay Reject

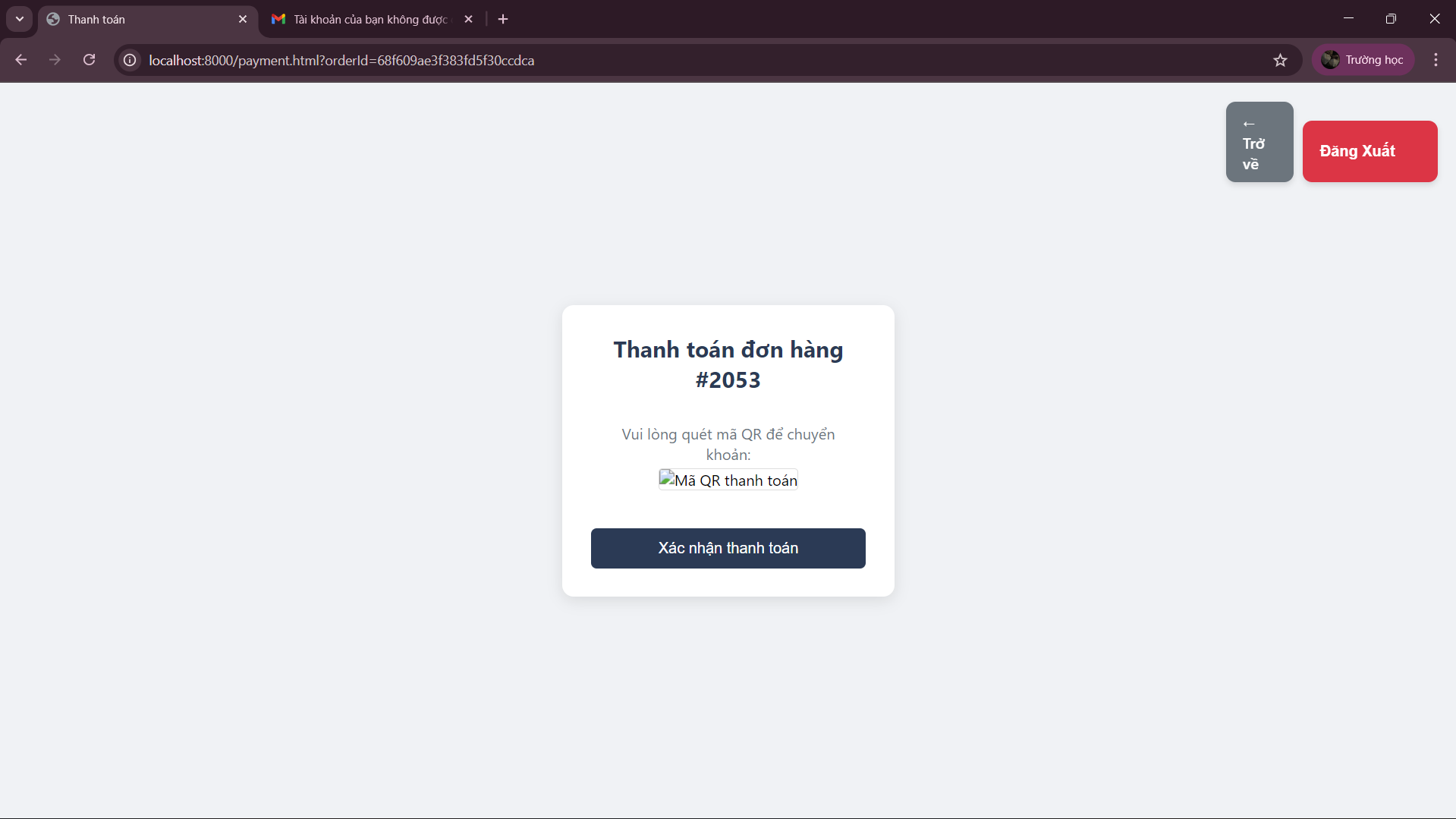
TEST: Nếu chọn Reject, tài khoản mới sẽ bị xóa và không lưu lại trong hệ thống.

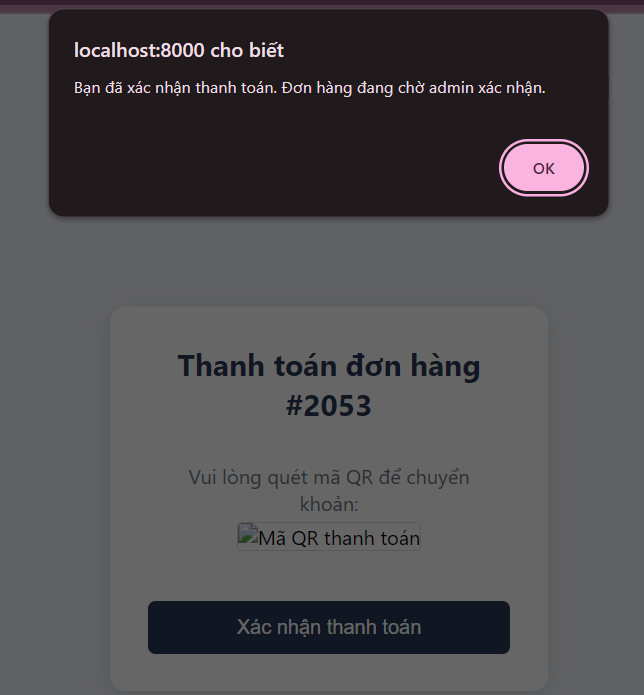


Nếu chọn Accept, tài khoản mới sẽ được chấp thuận đăng nhập vào user\_dashboard

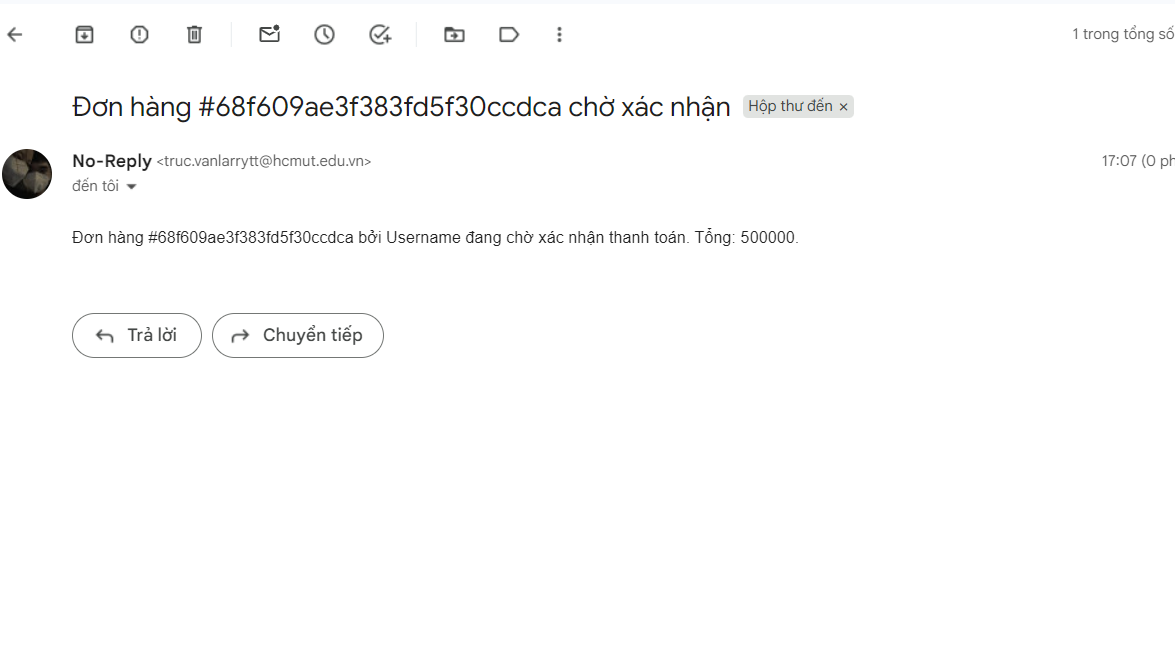


Giao diện user\_dashboard sẽ có các mục giao dịch và thanh toán. Nếu chọn mua gói nào thì sẽ báo giá Xác nhận thanh toán, sau đó sẽ gửi Email thanh toán đến admin về giá gói hàng.

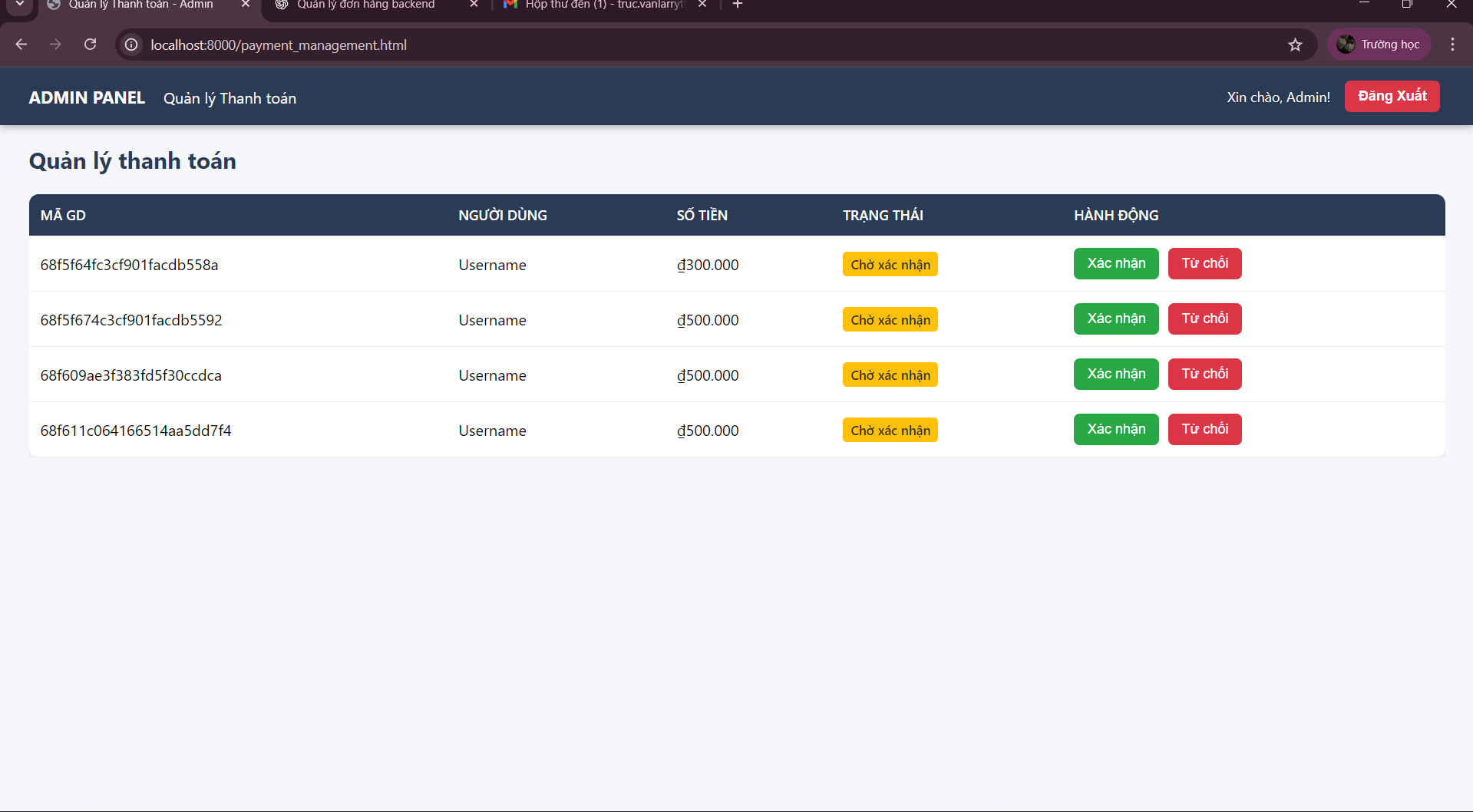




Ngay lập tức tài khoản email admin sẽ nhận thông báo:

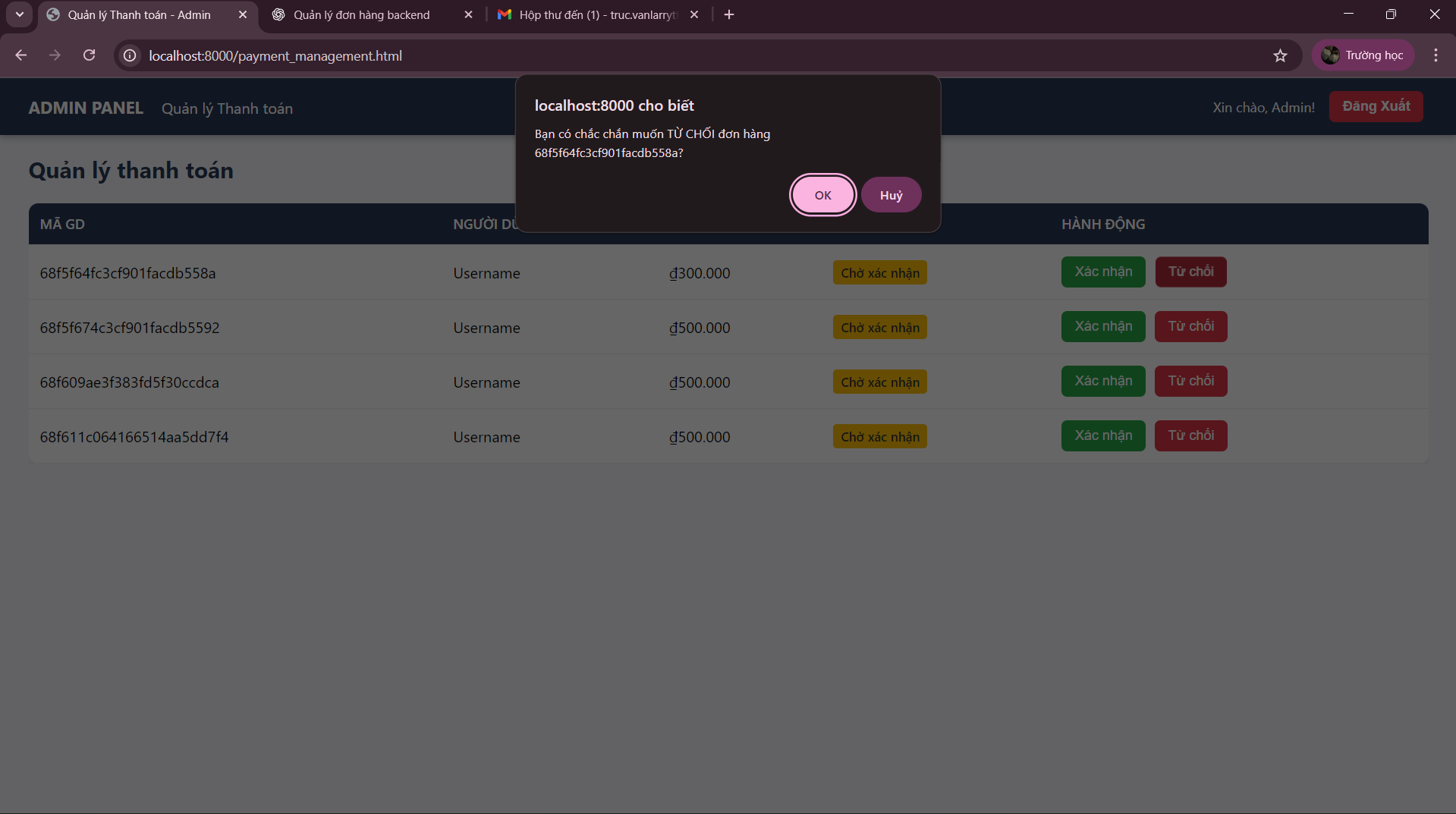


Sau khi nhận email, admin sẽ thao tác truy cập vào Quản lý đơn hàng để xem xét về đơn hàng:

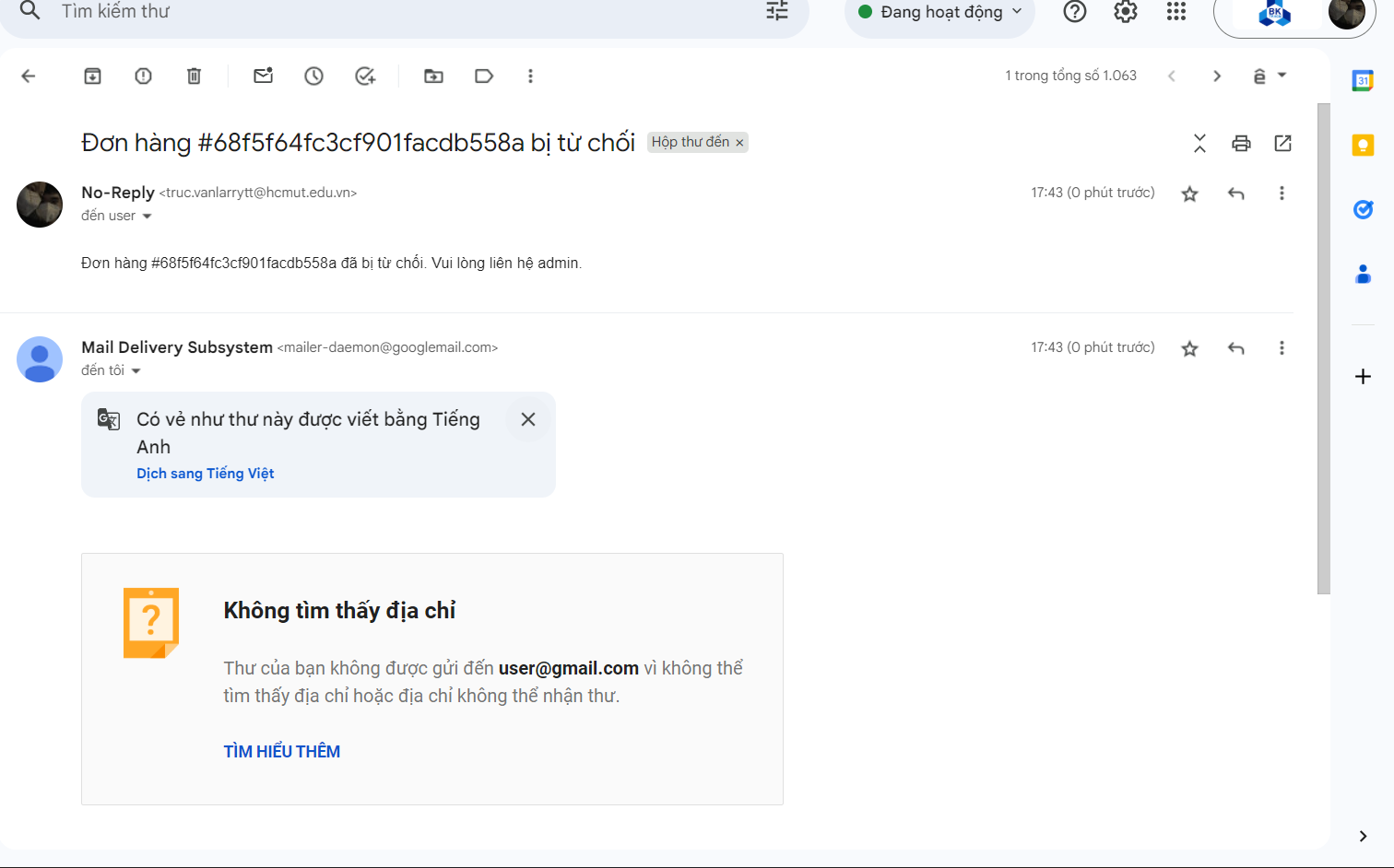


Vì đã mã hóa mã GD, chỉ xem được tên người dùng, có thể thấy các trạng thái của đơn hàng.

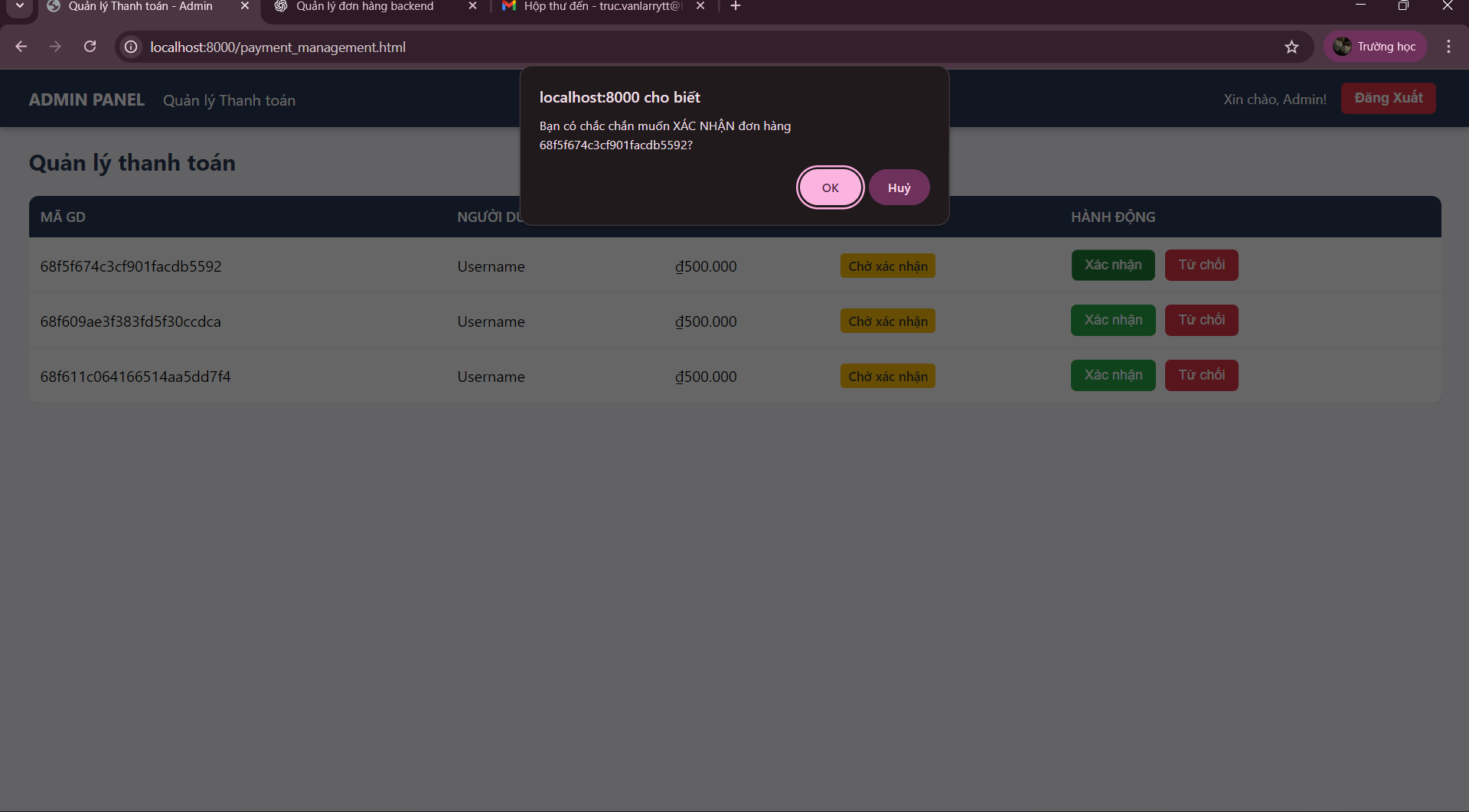
Admin sẽ thao tác Xác nhận hoặc Từ chối:



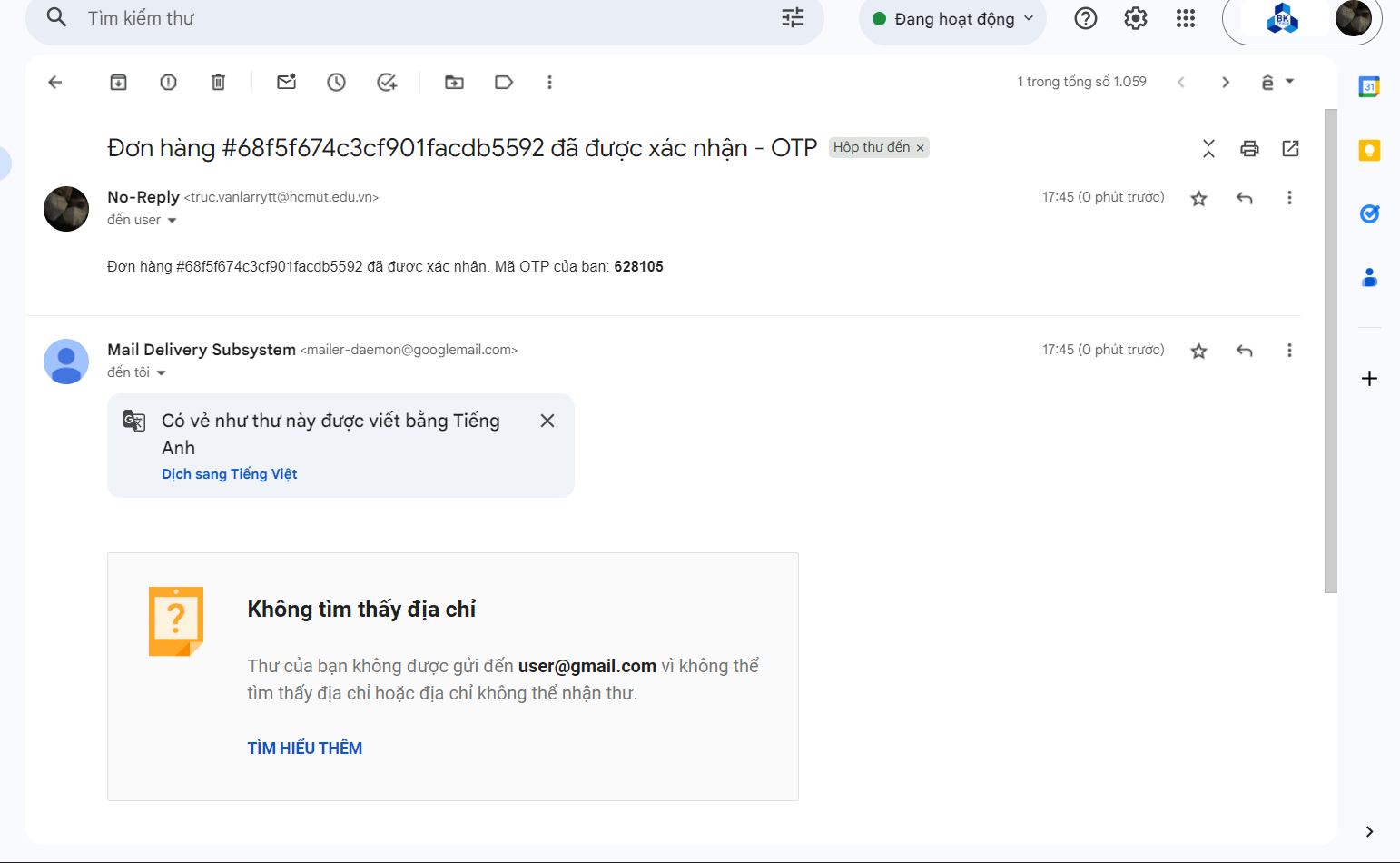
TEST: Từ chối đơn hàng



Sau khi từ chối, lập tức email từ chối đơn hàng được gửi về user.

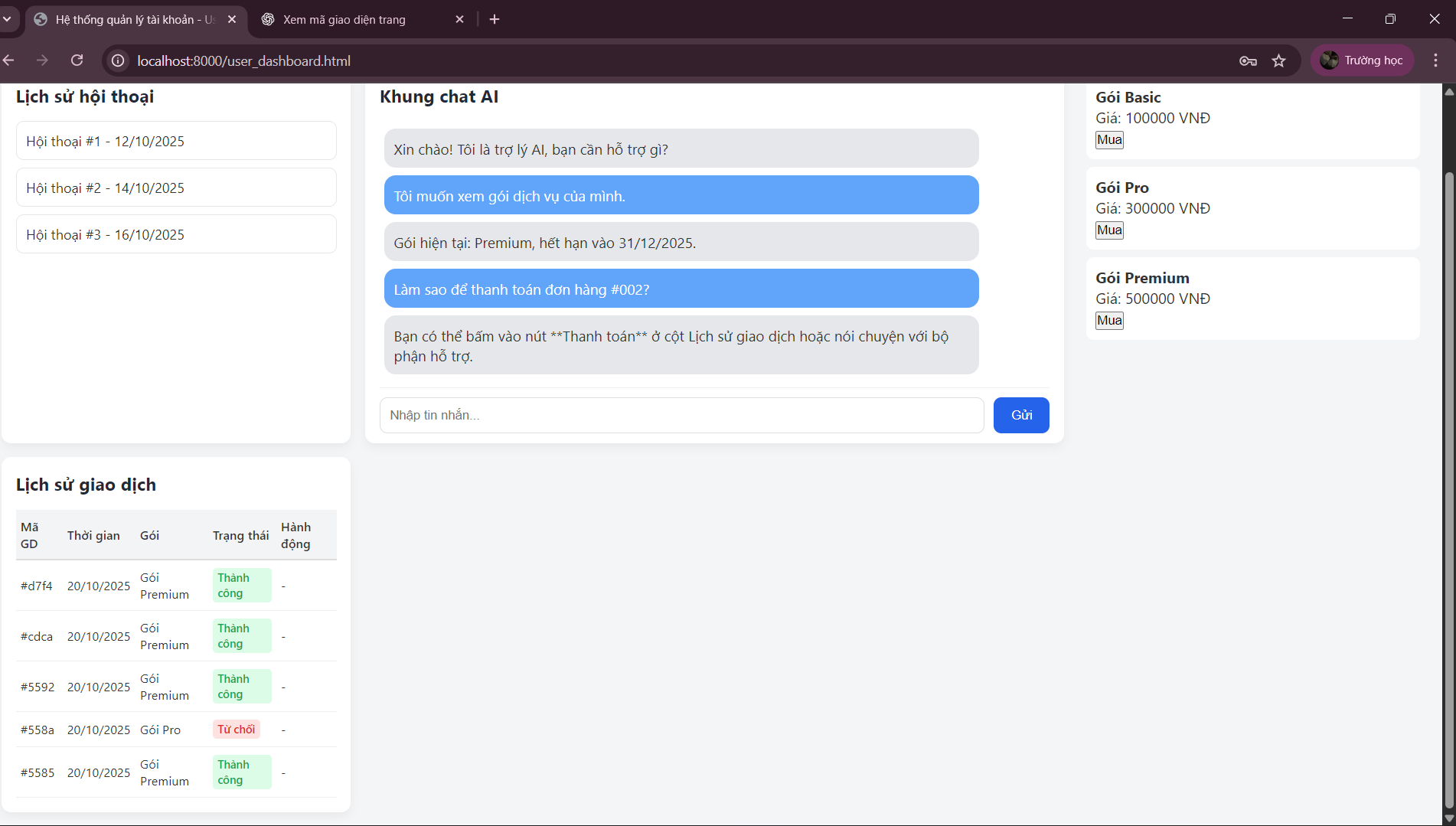


TEST: Xác nhận đơn hàng



Sau khi xác nhận đơn hàng, ngay lập tức email xác nhận được gửi về cho user kèm mã OTP.

Mã OTP này sẽ để phát triển bảo mật sản phẩm cho user, nhằm đảm bảo tính độc quyền của sản phẩm.

****

Kiểm tra lại thông tin thanh toán trong Lịch sử giao dịch của user (user\_dashboard.html).

Tự động cập nhật sau 5s không cần refresh.

**Các tài khoản user nếu Admin muốn xóa đi thì phải truy cập vào MongoDB hoặc cơ sở dữ liệu người dùng liên kết riêng, đảm bảo tránh sự nhầm lẫn.**

**Việc phát triển thêm các tính năng mới như:**

* **OTP**
* **Xóa User**
* **Cập nhật thông tin user**
* **Dùng socket.io để cập nhật liên tục**
* **Các tác vụ khác,…**

**sẽ được phát triển sau.**