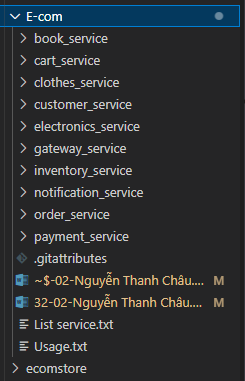
Bài 5:

Cấu trúc hệ thống:



Chi tiết code hệ thống được đặt tại: <https://github.com/thanhchauns2/E-com>

**I - Mô tả các service:**

1. Gateway Service: 1000

Chức năng:

1.1. Đăng kí

\* JSON truyền vào: {Username, Password, Email, Address, City, Country, First Name, Last Name}

\* Các chức năng được gọi: 4.1, 4.3, 4.4, 4.5

1.2. Đăng nhập

(Tạm thời chưa cần)

1.3. Quản lí kho hàng:

1.3.1. Khởi tạo kho hàng + cập nhật kho hàng

\* JSON truyền vào (có thể có hoặc không): {Address, Book Quantity, Clothes Quantity, Electronic Quantity, Status}

\* Các chức năng được gọi: 8.1, 8.2

1.3.2. Thêm hàng vào kho

\* JSON truyền vào: {Product ID, Quantity}

\* Các chức năng được gọi: 8.3

1.3.3. Xoá hàng khỏi kho

\* JSON truyền vào: {Product ID, Quantity}

\* Các chức năng được gọi: 8.5

1.3.4 Kiểm tra kho hàng

\* JSON truyền vào: {}

\* Các chức năng được gọi: 8.4

1.3.5 Đăng kí một mặt hàng mới

\* JSON truyền vào: {Category Name, Author Name, Author Email, Author Address, Book Title, Publishded Date, Price, Description}

\* Các chức năng được gọi: 2.1.1, 2.1.2, 2.1.3

1.4. Quản lí giỏ hàng:

1.4.1. Thêm hàng vào giỏ hàng

\* JSON truyền vào: {User ID, Product ID, Quantity}

\* Các chức năng được gọi: 5.2

1.4.2. Xoá hàng khỏi giỏ hàng

\* JSON truyền vào: {User ID, Product ID, Quantity}

\* Các chức năng được gọi: 5.3

1.4.3. Kiểm tra giỏ hàng

\* JSON truyền vào: {User ID}

\* Các chức năng được gọi: 5.4

1.4.4. Mua hàng

\* JSON truyền vào: {User ID}

\* Các chức năng được gọi: 3.1, 5.5

1.5. Quản lí hoá đơn

1.5.1 Theo dõi đơn hàng

\* JSON truyền vào: {Order ID}

\* Các chức năng được gọi: 3.3

1.5.2 Cập nhật đơn hàng

\* JSON truyền vào: {Order ID, Status}

\* Các chức năng được gọi: 3.2, 3.3

2. Product Service:

Chia ra thành các chức năng nhỏ hơn:

- Book: 2100 - done

2.1.1: Tạo author:

\* JSON truyền vào: {Author Name, Email, Address}

2.1.2: Tạo category

\* JSON truyền vào: {Category Name}

2.1.3: Tạo book:

\* JSON truyền vào: {Title, Author ID, Category ID, Published Date, Price, Description}

2.1.4: Xoá book:

\* JSON truyền vào: {Book ID}

2.1.5: Tìm author

\* JSON truyền vào: {Author Name}

2.1.6: Tìm category

\* JSON truyền vào: {Category Name}

2.1.7: Tìm book

\* JSON truyền vào: {search\_term}, search\_term có thể là: id sách, tên sách, tên category, tên author

- Electronics: 2200 - Not neccessary

- Clothes: 2300 - Not neccessary

3. Order Service: 3000 - done

Chức năng:

3.1. Tạo hoá đơn:

\* JSON truyền vào: {User ID}

\* Các chức năng được gọi: 5.4, 5.5

3.2. Cập nhật tình trạng hoá đơn

\* JSON truyền vào: {Order ID, Status}

3.3. Kiểm tra hoá đơn

\* JSON truyền vào: {Order ID}

4. Customer Service: 4000 - done

Chức năng:

4.1. Tạo tài khoản mới:

\* JSON truyền vào: {Username, Password, Email}

4.2. Xoá tài khoản:

\* JSON truyền vào: {Account ID}

4.3. Tạo fullname:

\* JSON truyền vào: {First Name, Last Name}

4.4. Tạo address:

\* JSON truyền vào: {Address, City, Country}

4.5. Tạo customer:

\* JSON truyền vào: {Account ID, Fullname ID, Address ID}

\* Các chức năng được gọi: 5.1

Công nghệ sử dụng: Django ORM

5. Cart Service: 5000 - done

Chức năng:

5.1. Tạo một cart:

\* JSON truyền vào: {User ID}

5.2. Thêm item vào cart:

\* JSON truyền vào: {User ID, Product ID, Quantity}

5.3. Xoá item khỏi cart:

\* JSON truyền vào: {User ID, Product ID, Quantity}

5.4. Xem thông tin cart:

\* JSON truyền vào: {User ID}

\* Các chức năng được gọi: 2.1.7

5.5. Clear tất cả item trong cart

\* JSON truyền vào: {User ID}

6. Payment Service: 6000

Chức năng: Xử lý thanh toán cho các đơn hàng, bao gồm thông tin thanh toán, xác nhận giao dịch, v.v.

Công nghệ sử dụng: Django ORM

7. Notification Service: 7000

Chức năng: Gửi thông báo đến khách hàng về các hoạt động trong hệ thống, bao gồm cập nhật đơn hàng, thanh toán, v.v.

Công nghệ sử dụng: Django ORM

8. Inventory Service: 8000

Chức năng:

8.1. Khởi tạo kho hàng

\* JSON truyền vào: {}

8.2. Cập nhật thông tin kho hàng (địa chỉ, kích thước, status):

\* JSON truyền vào (có thể có hoặc không): {Address, Book Quantity, Clothes Quantity, Electronic Quantity, Status}

8.3. Thêm một món hàng vào kho

\* JSON truyền vào: {Product ID, Quantity}

8.4. Trả về thông tin kho hàng

\* JSON truyền vào: {}

8.5. Xoá một món hàng khỏi kho

\* JSON truyền vào: {Product ID, Quantity}

8.6. Kiểm tra dung tích còn thừa của kho hàng

\* JSON truyền vào: {Product ID, Quantity}

**II – Sử dụng API Gateway để điều khiển project**

{

// Đăng kí

// "Username" : "abc1332313",

// "Password" : "1233334361",

// "Email" : "33313@gmail.com",

// "Address" : "1ab333cxyz",

// "City" : "Ha3n33i1",

// "Country" : "Vie3t333am1",

// "First Name" : "Ng3333yen1",

// "Last Name" : "33331"

// Khởi tạo kho hàng + cập nhật kho hàng

// "Address" : "Hanoi",

// "Book Quantity" : 50000,

// "Clothes Quantity" : 50000,

// "Electronic Quantity" : 50000,

// "Status" : "Ready"

// Đăng kí một mặt hàng mới

// "Category Name" : "Horror",

// "Author Name" : "Edgar Allan Poe",

// "Author Email" : "a@gmail.com",

// "Author Address" : "ab22c3xyz",

// "Book Title" : "book",

// "Publishded Date" : "2020-10-10",

// "Price" : "10",

// "Description" : "abasdasdasd"

// Thêm hàng vào kho

// "Product ID" : 74827708490362,

// "Quantity" : 100

// Xoá hàng khỏi kho

// "Product ID" : 74827708490362,

// "Quantity" : 100

// Kiểm tra kho hàng

// "User ID" : 14886562145040

// Thêm hàng vào giỏ hàng

// "User ID" : 14886562145040,

// "Product ID" : 74827708490362,

// "Quantity" : 100

// Xoá hàng khỏi giỏ hàng

// "User ID" : 14886562145040,

// "Product ID" : 74827708490362,

// "Quantity" : 100

// Kiểm tra giỏ hàng

// "User ID" : 14886562145040

// Mua hàng

// "User ID" : 14886562145040

// Theo dõi đơn hàng

// "Order ID" : 29662756138573

// Cập nhật trạng thái đơn hàng

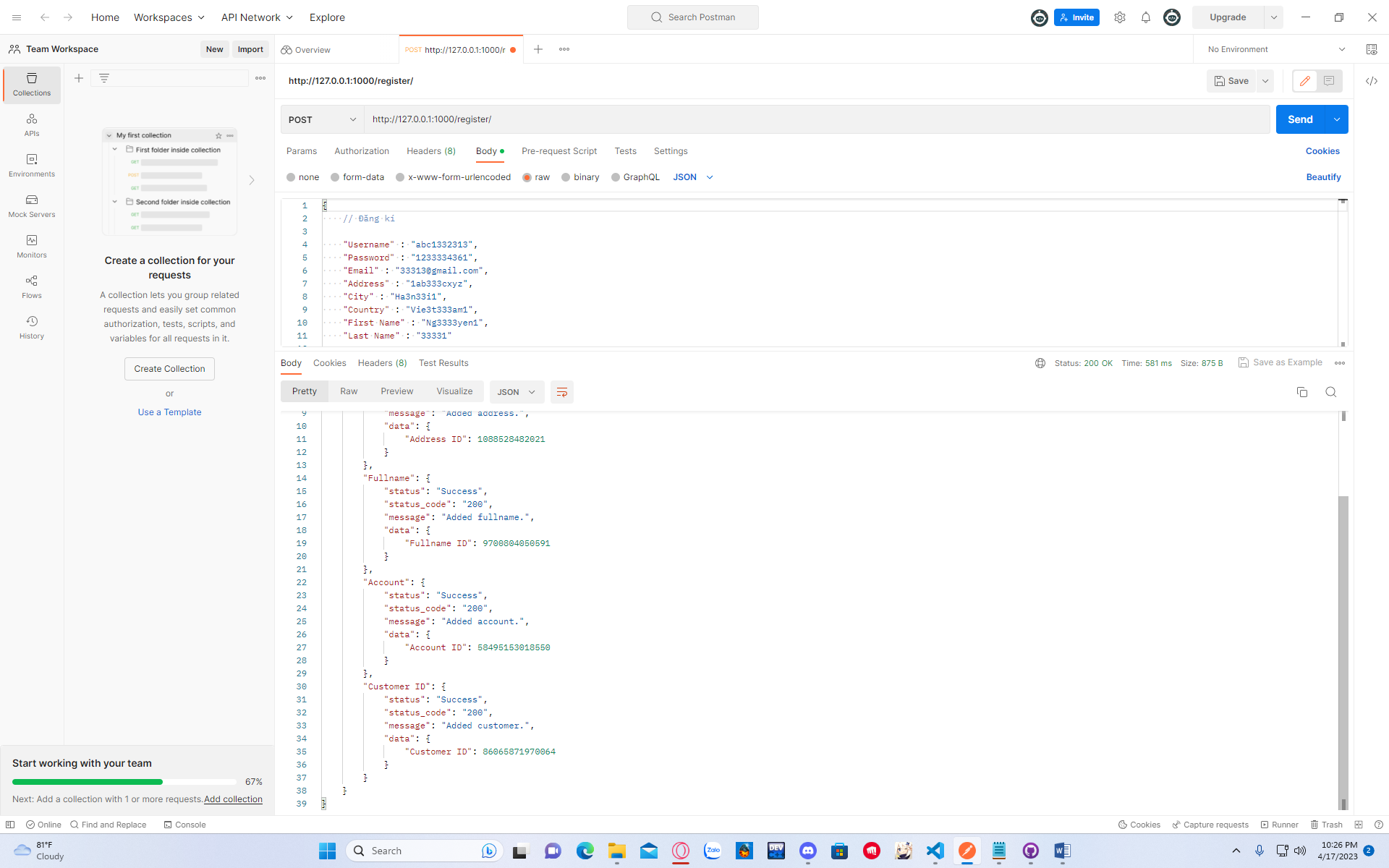
// "Order ID" : 29662756138573,

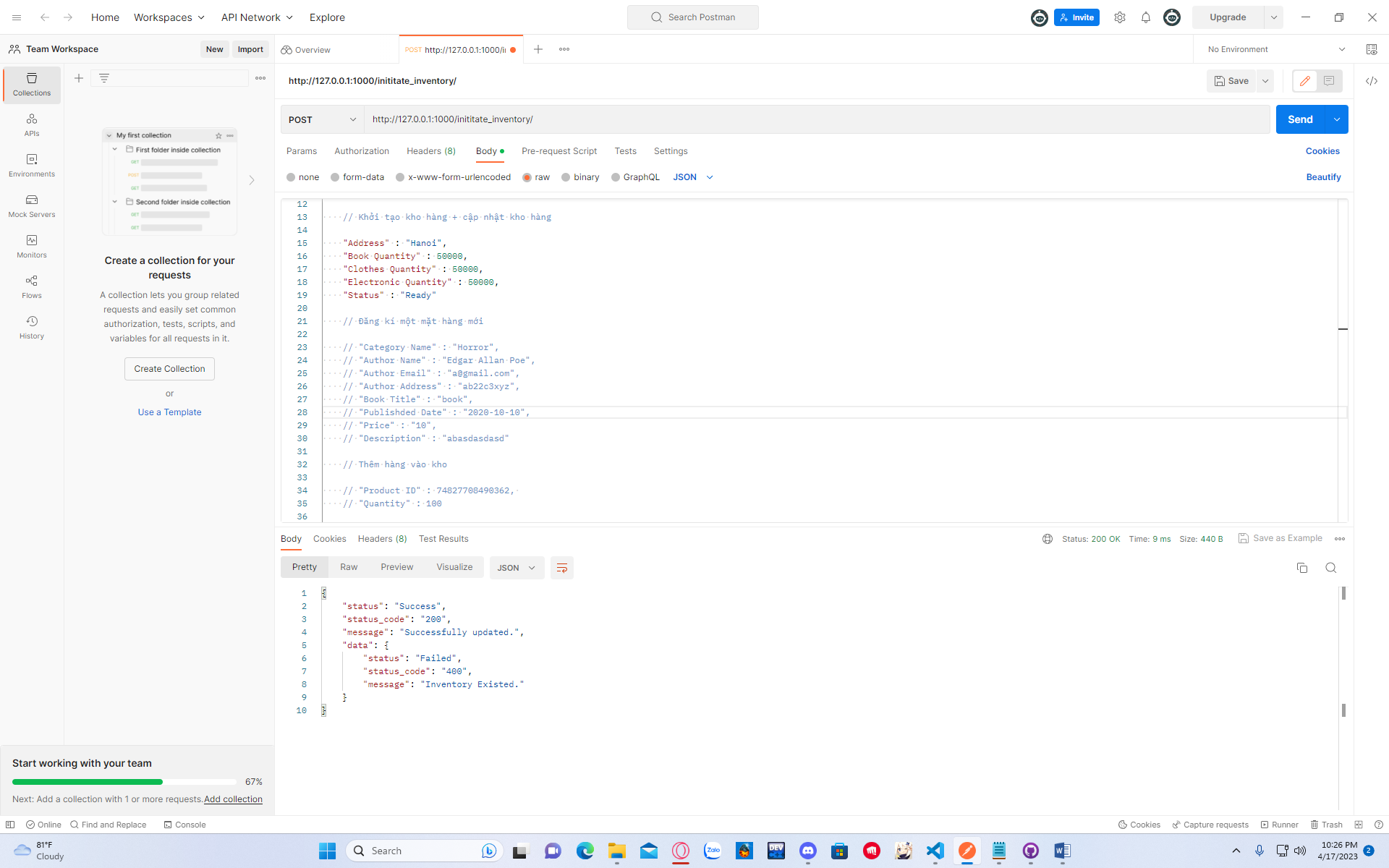
// "Status" : "Done"

}

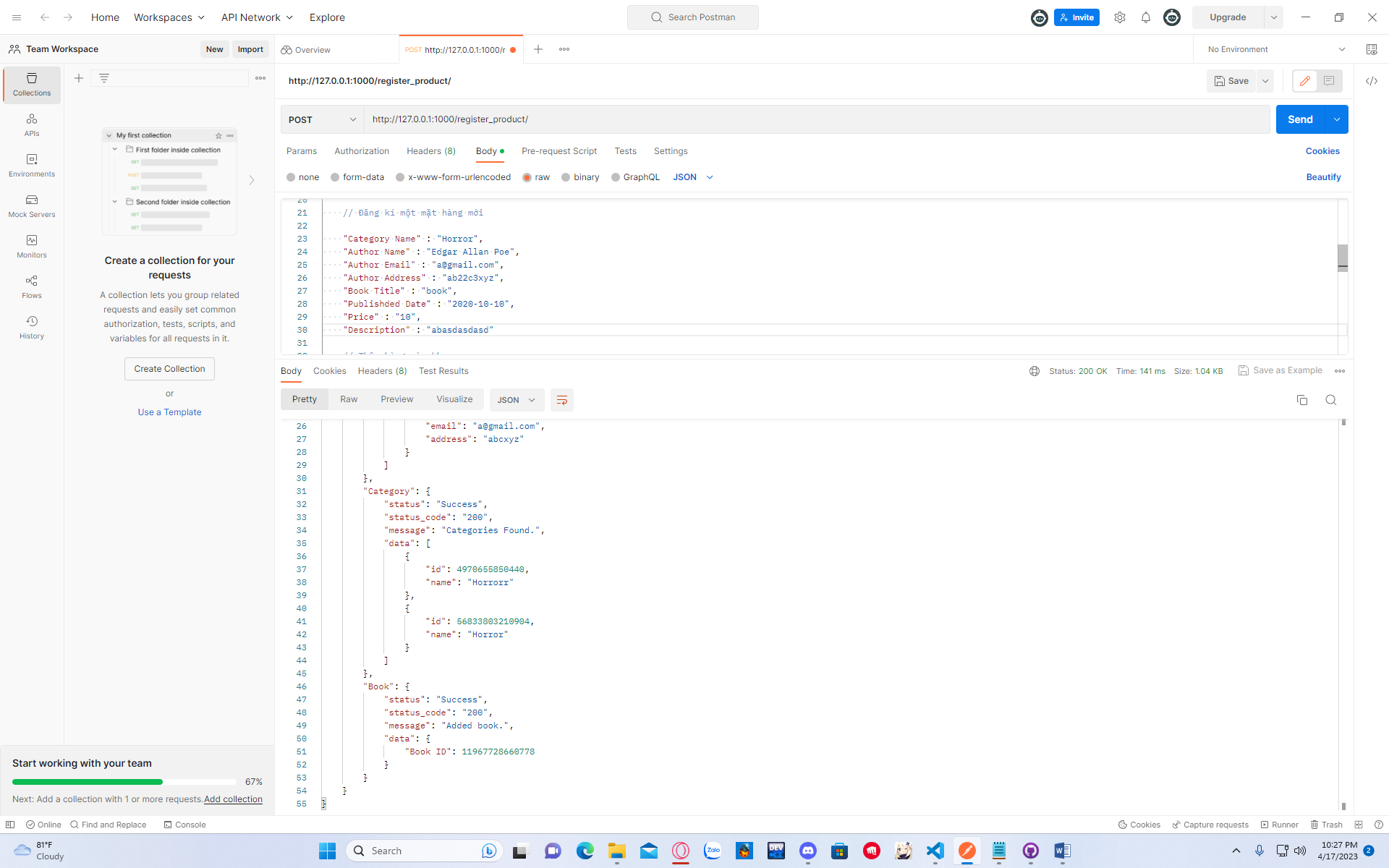
**III - Ảnh chụp màn hình khi chạy**

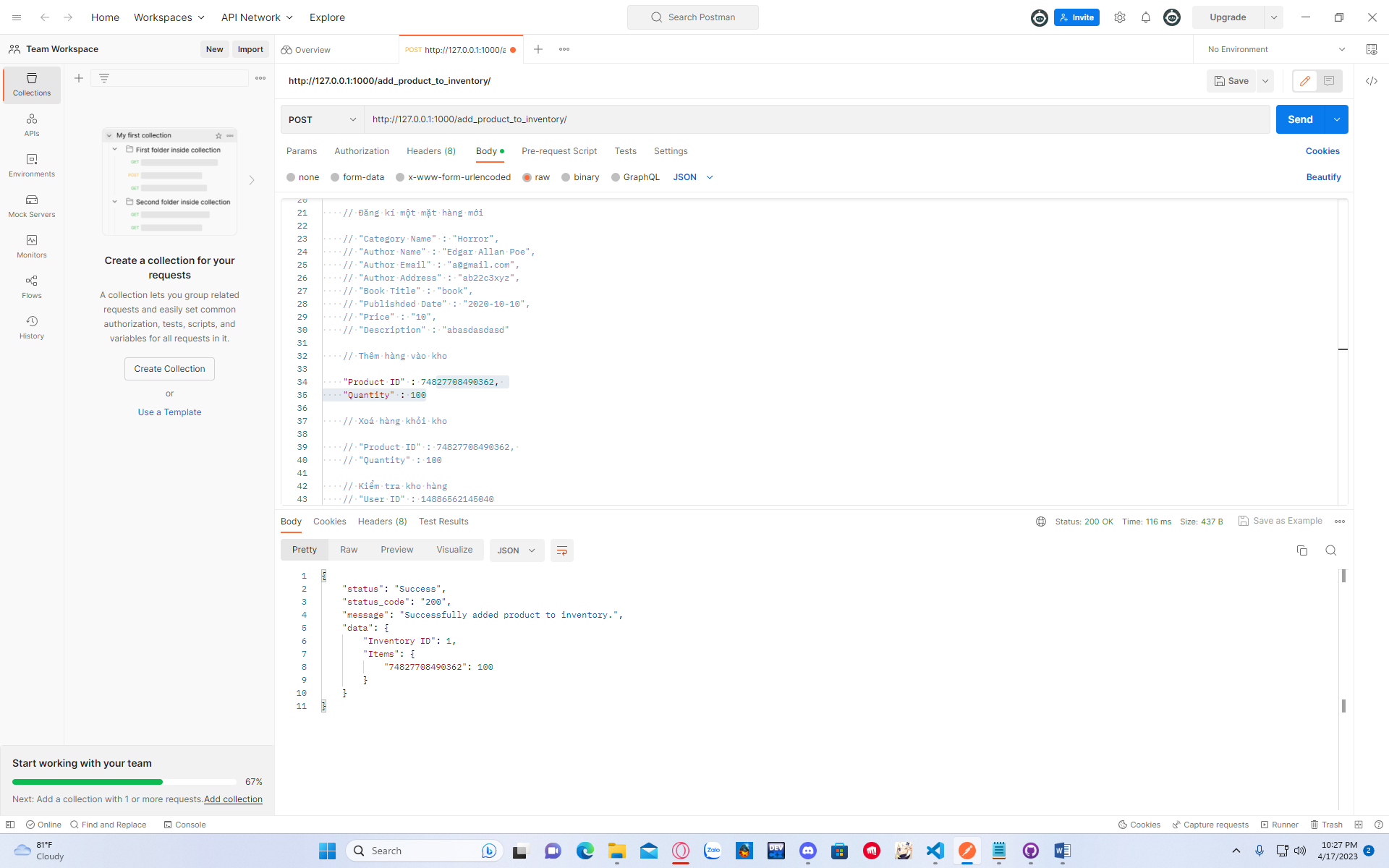
1. Đăng kí



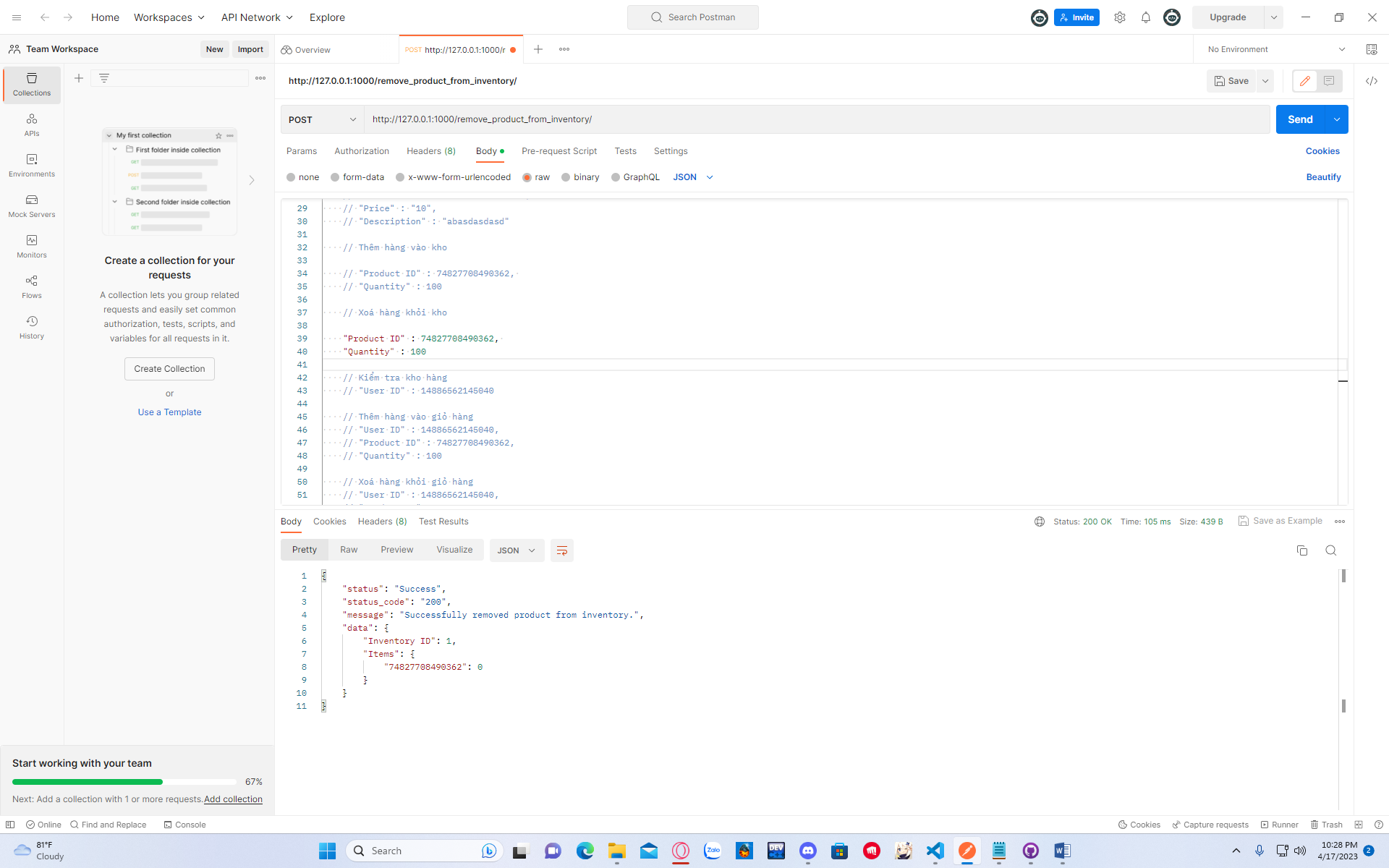
2. Khởi tạo kho hàng

3. Đăng kí một mặt hàng mới



4. Thêm hàng vào kho

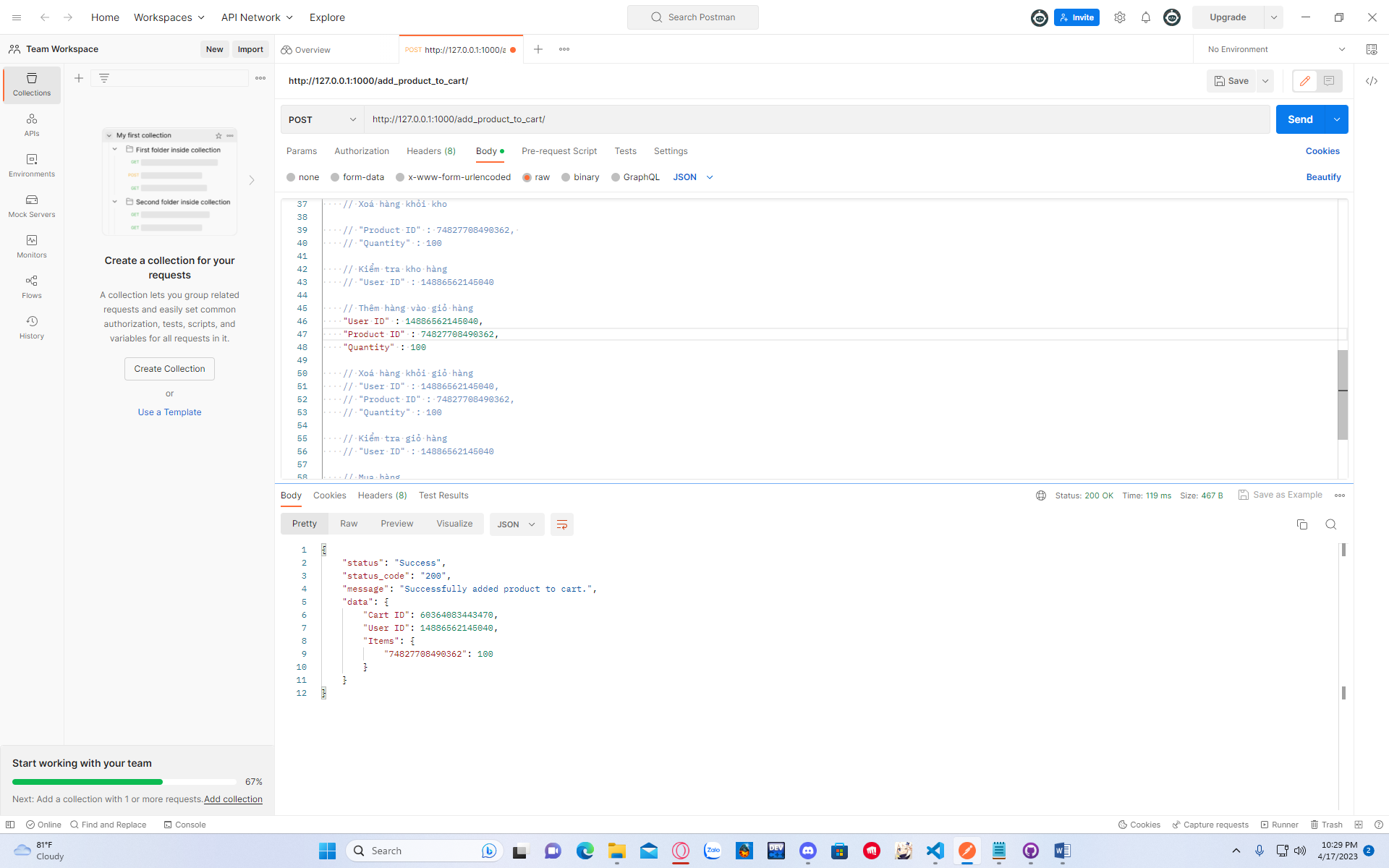
5. Xoá hàng khỏi kho



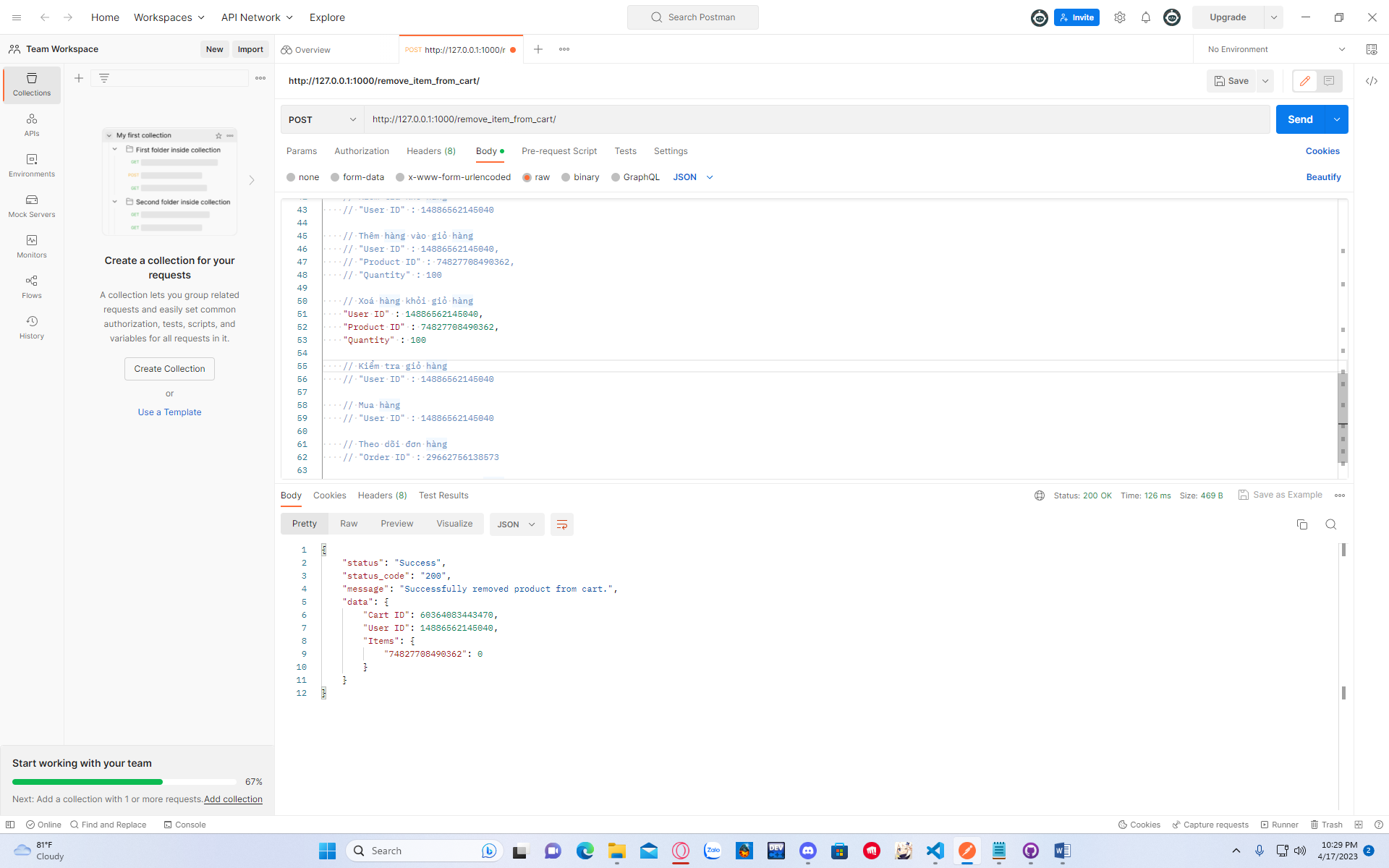
6. Kiểm tra kho hàng



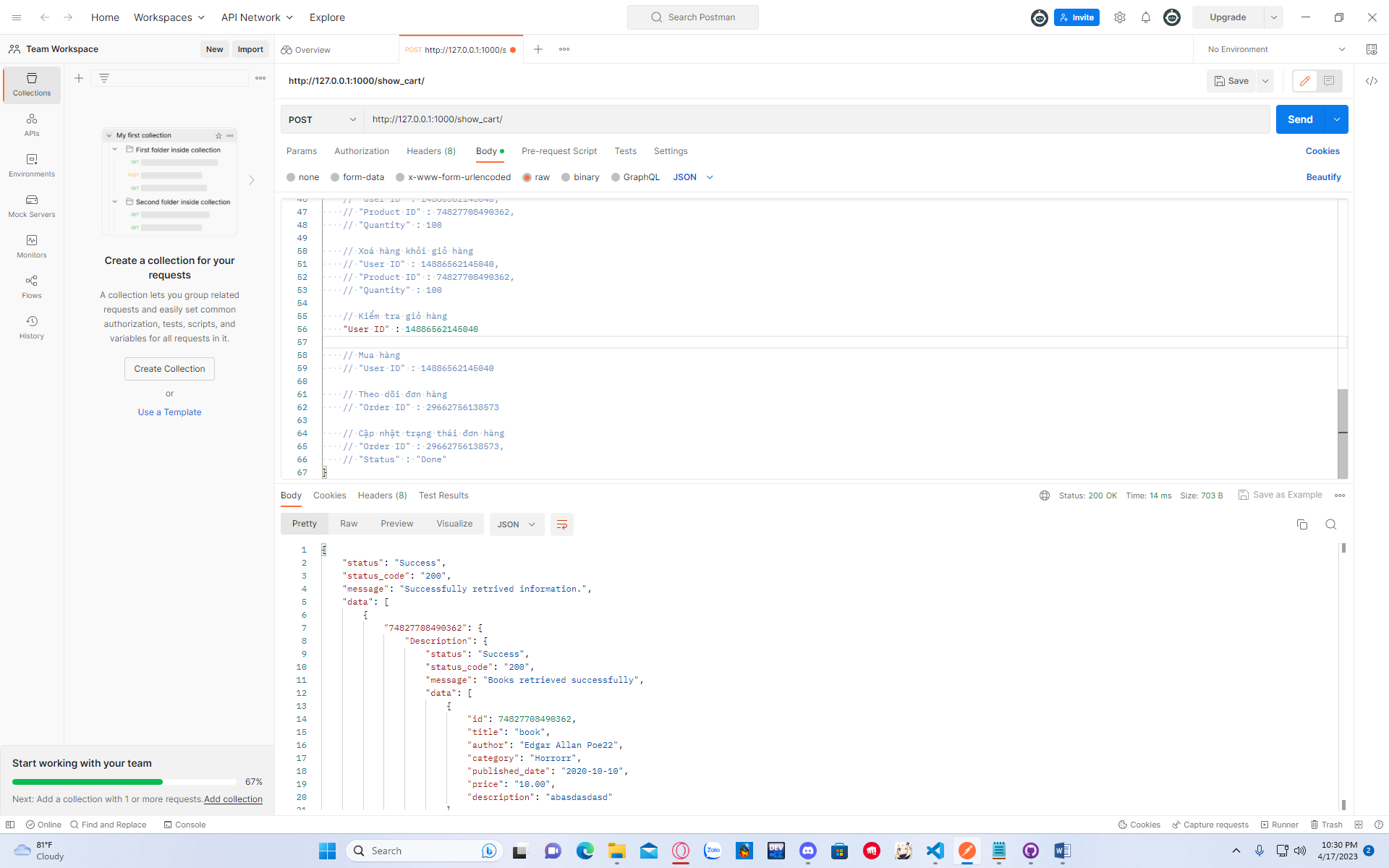
7. Thêm hàng vào giỏ hàng



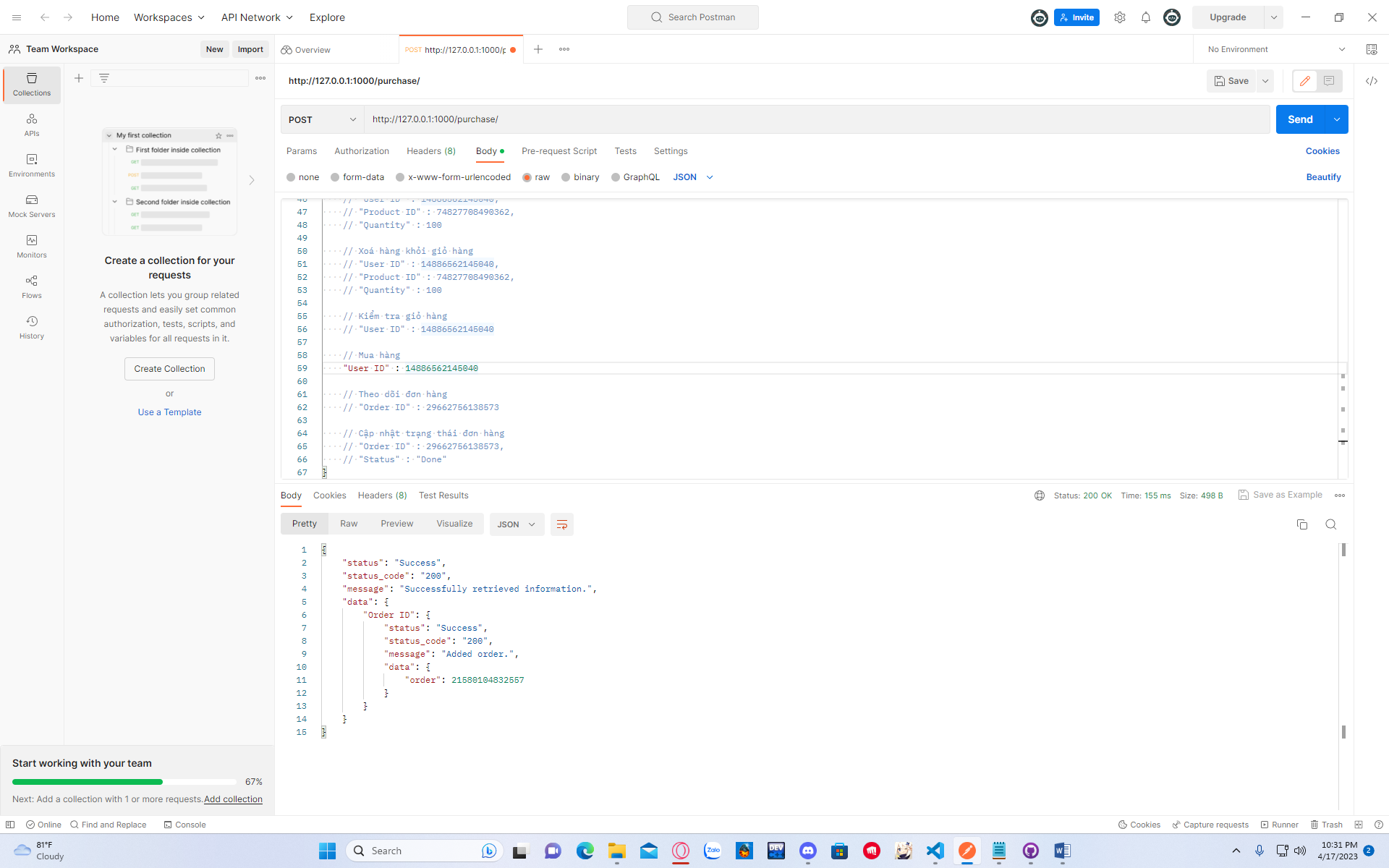
8. Xoá hàng khỏi giỏ hàng



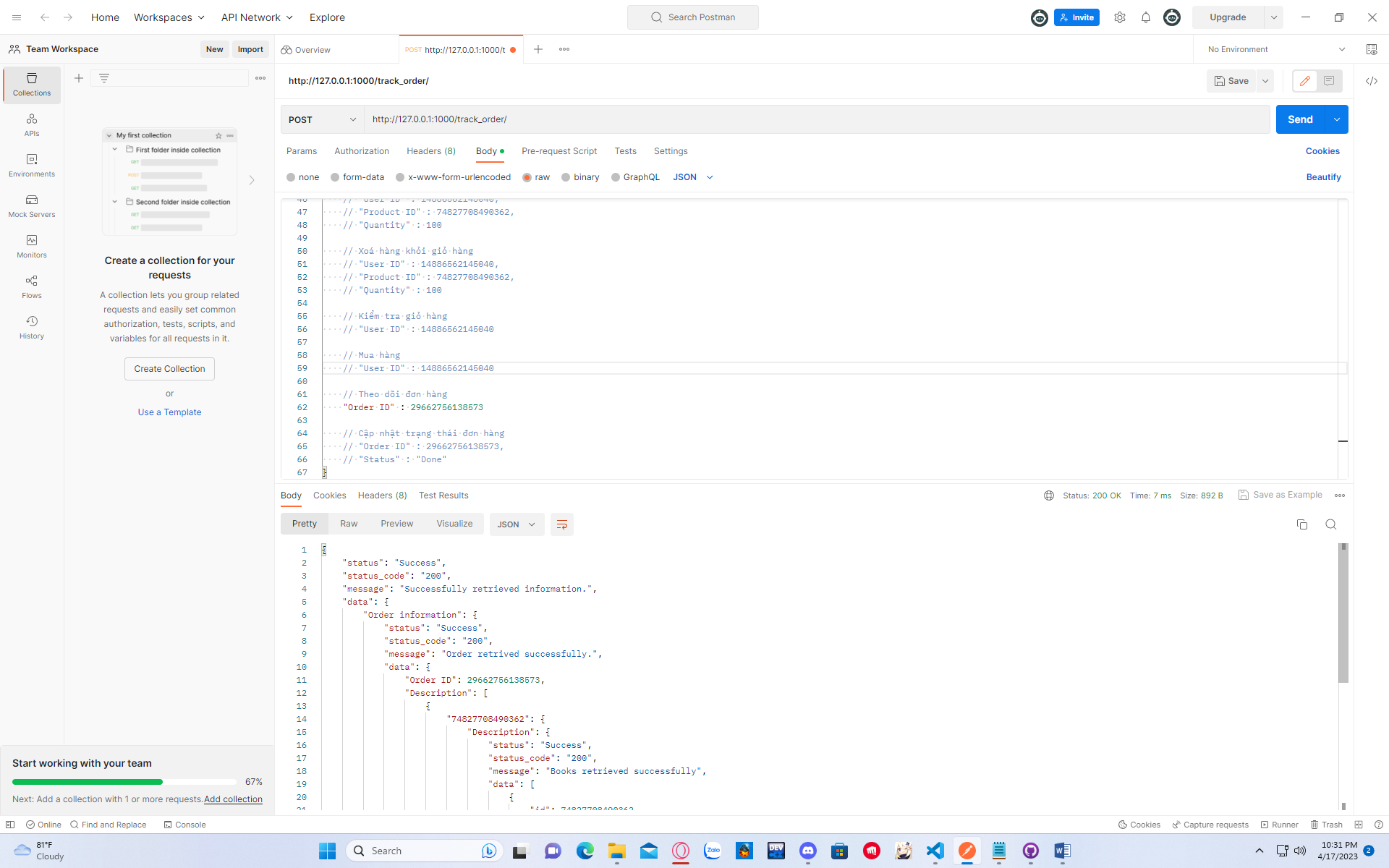
9. Kiểm tra giỏ hàng



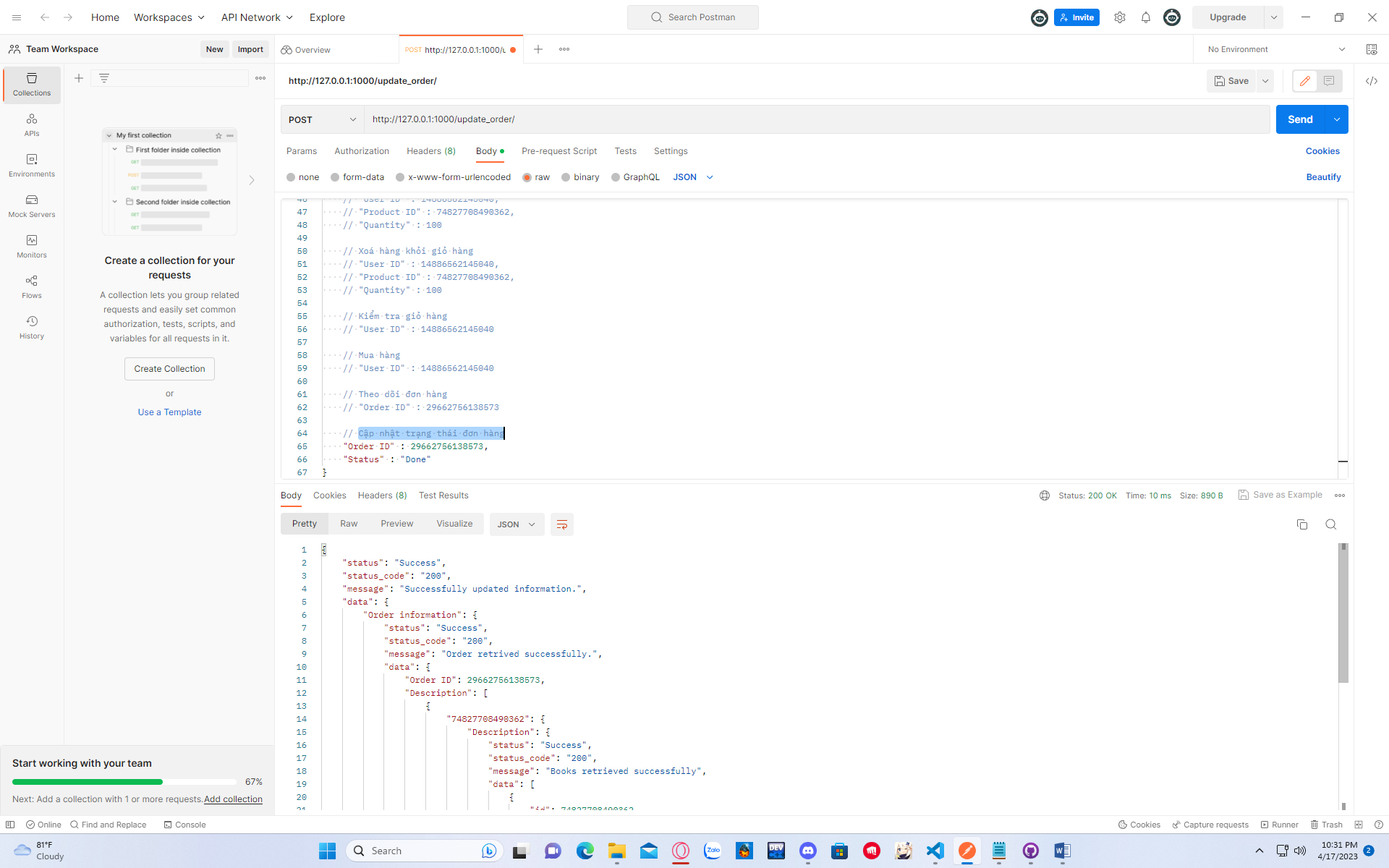
10. Mua hàng



11. Theo dõi đơn hàng

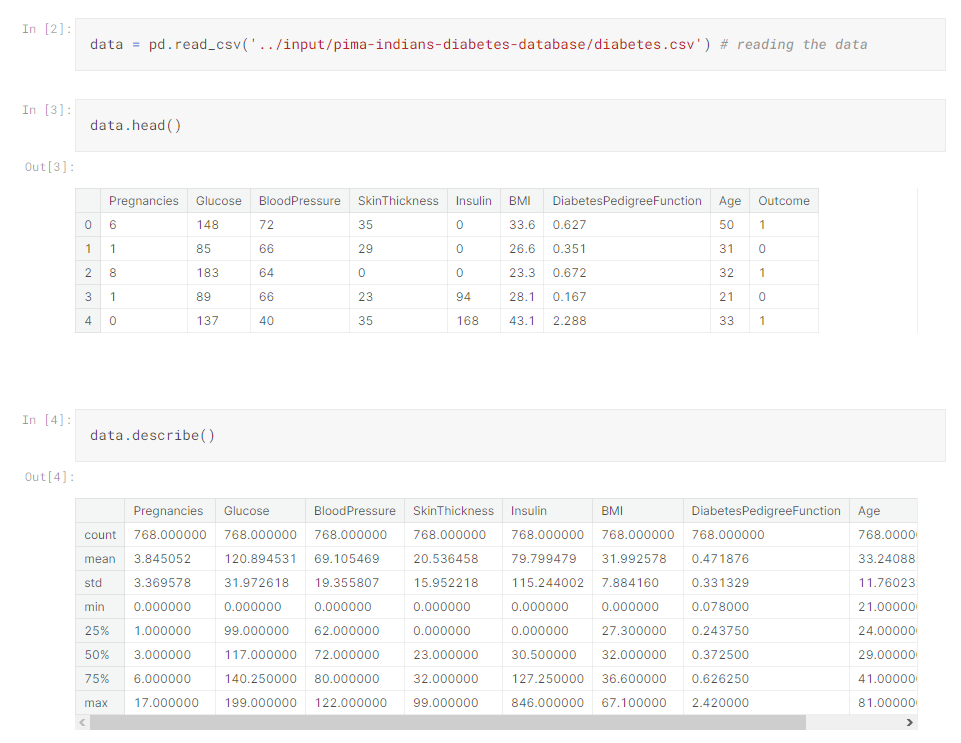


12. Cập nhật trạng thái đơn hàng

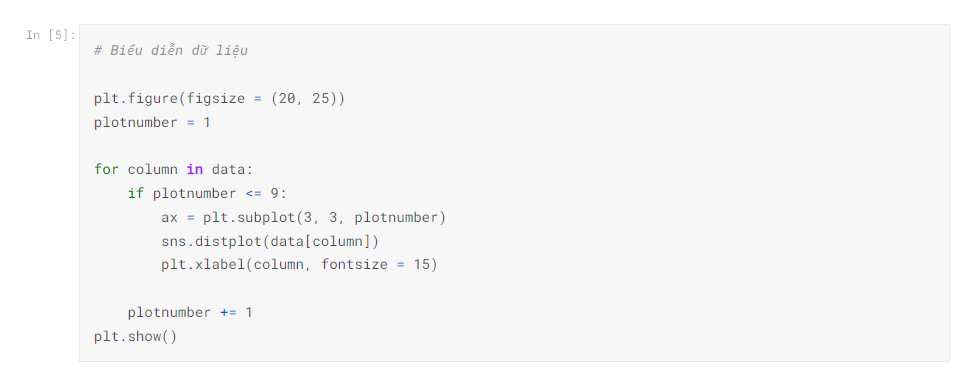


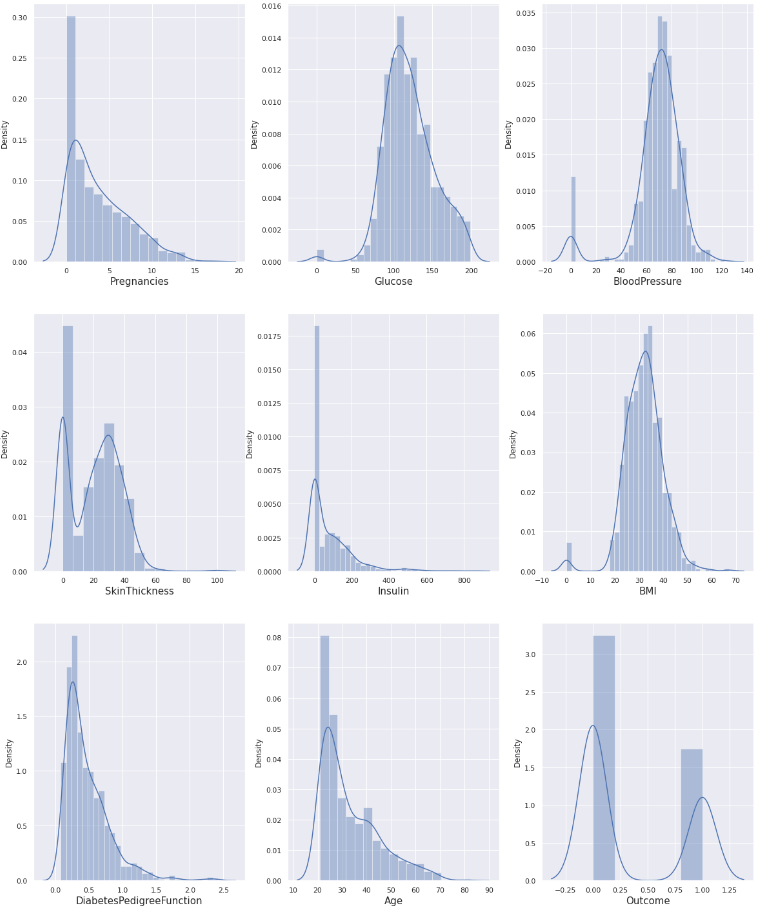
Bài 6: Em sử dụng kaggle cho bài tập 6. Em xin phép để url notebook ở đây:   
<https://www.kaggle.com/code/chaunguyenthanh/assignment-6-ml>  
<https://www.kaggle.com/code/chaunguyenthanh/assignment-6-dl>  
  
1. Khai báo thư viện



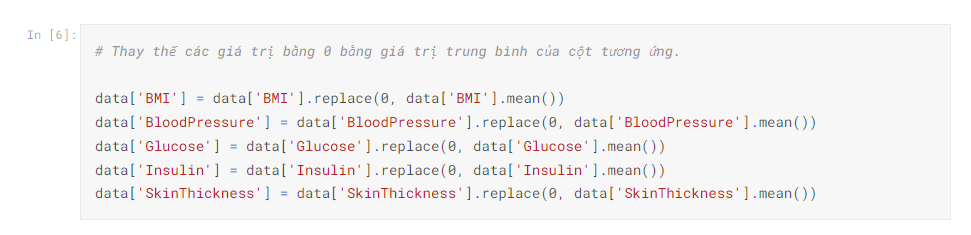
2. Review data  


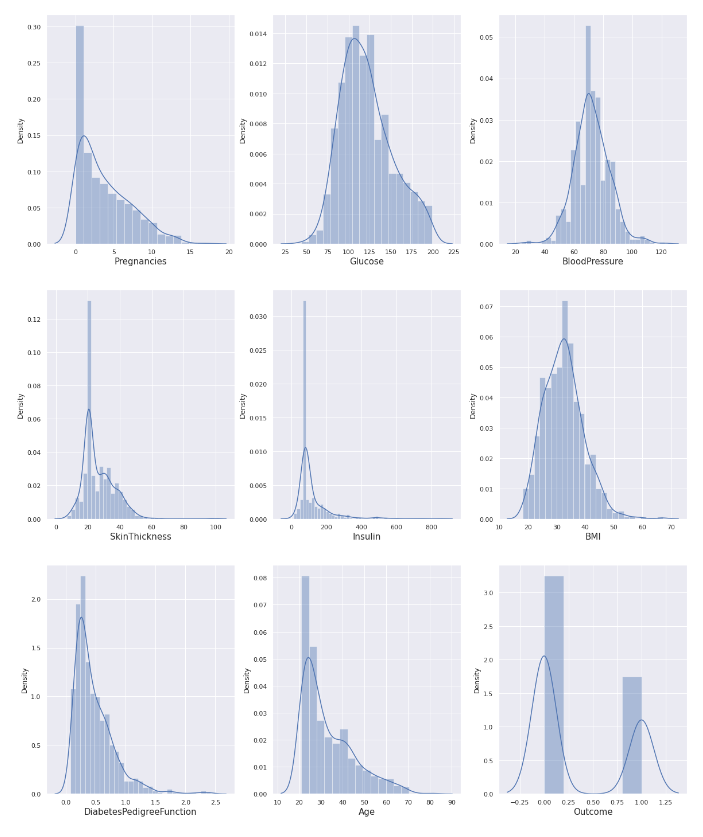
3. Biểu diễn dữ liệu



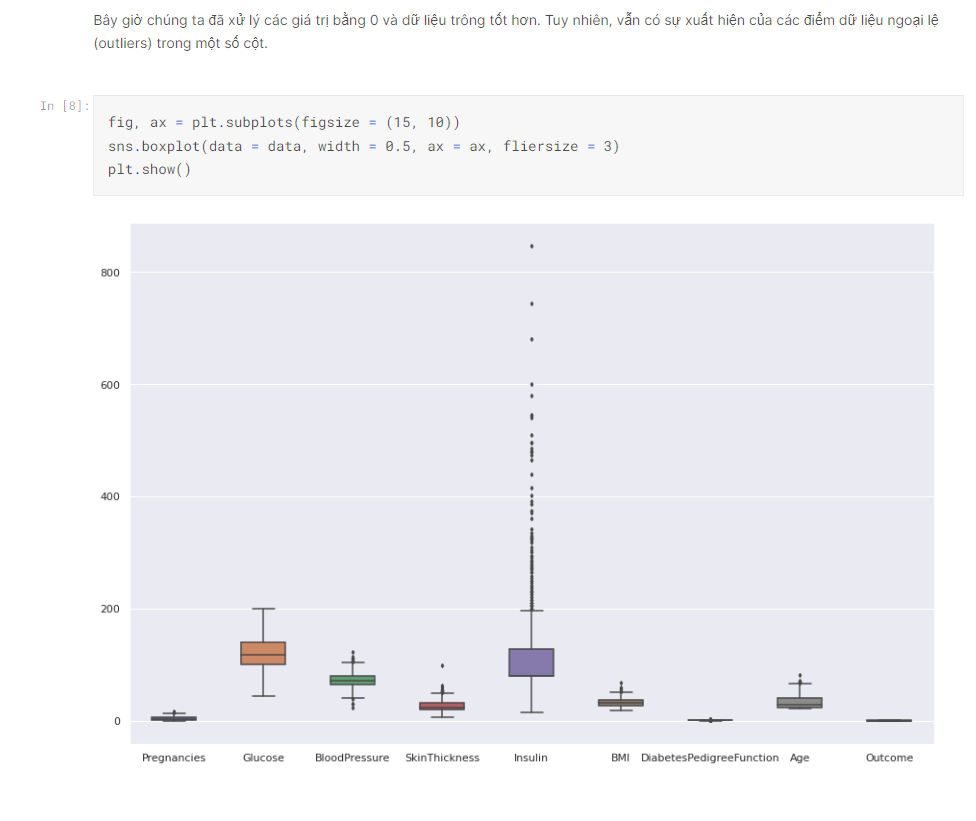


Một vài cột có giá trị bằng 0. Điều này không bình thường.

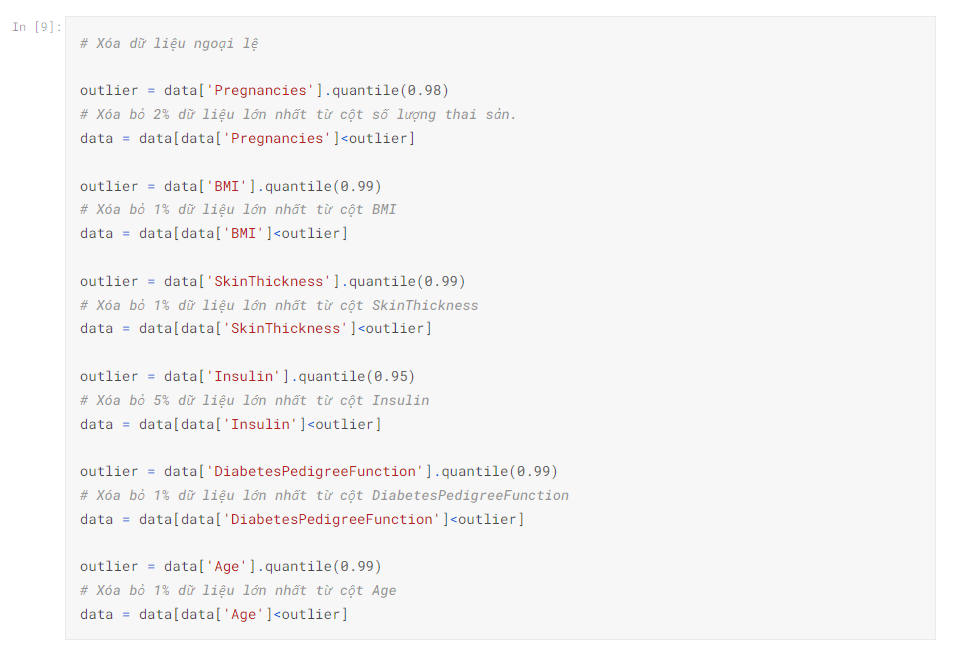
4. Xử lí các giá trị bị khuyết5/

5. Review data một lần nữa

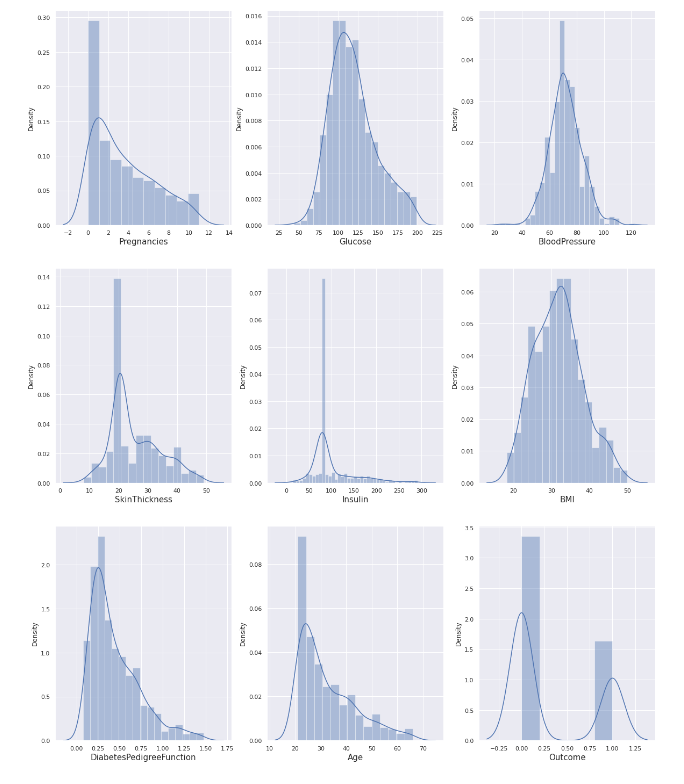
6. Phân tích ngoại vi



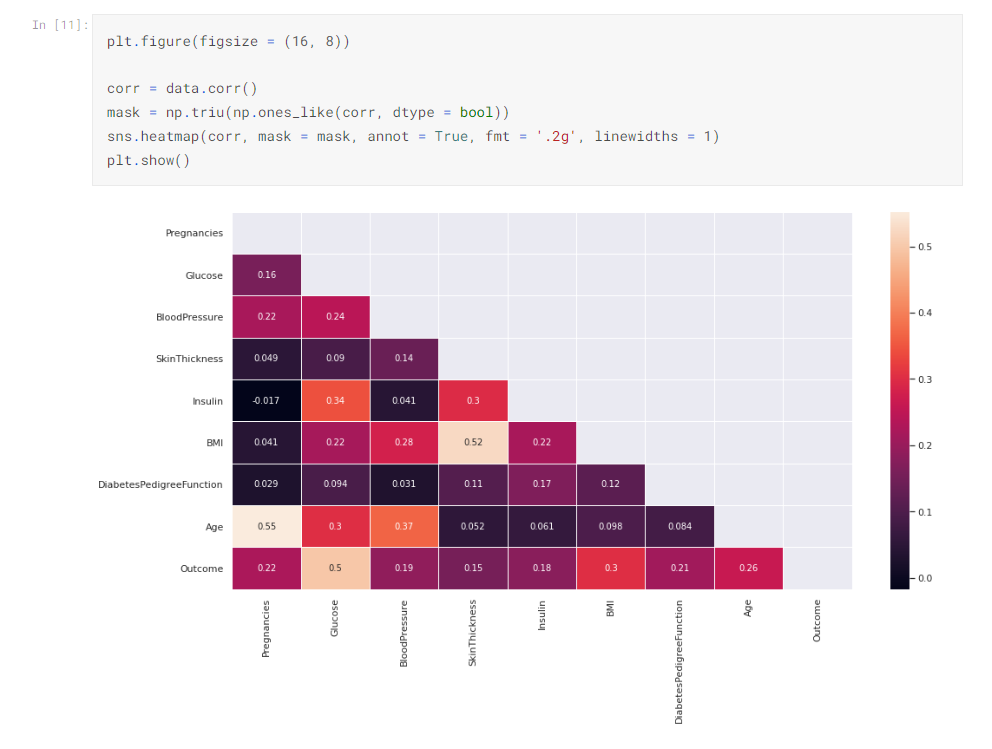
7. Loại bỏ ngoại vi



8. Biểu diễn lần nữa



9. Heatmap

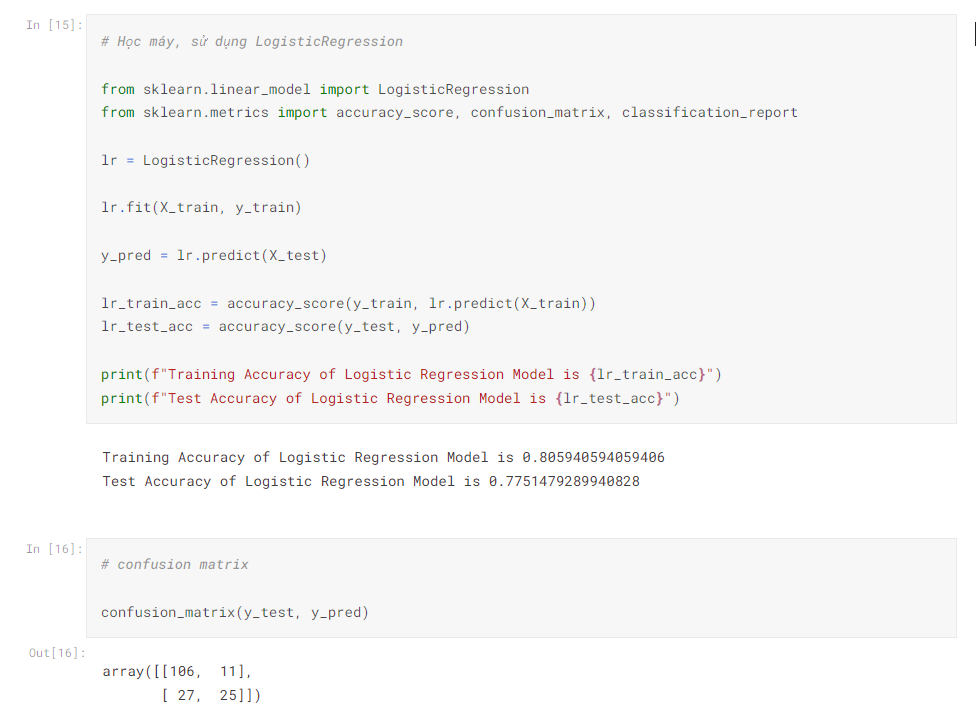


10. Phân chia dữ liệu

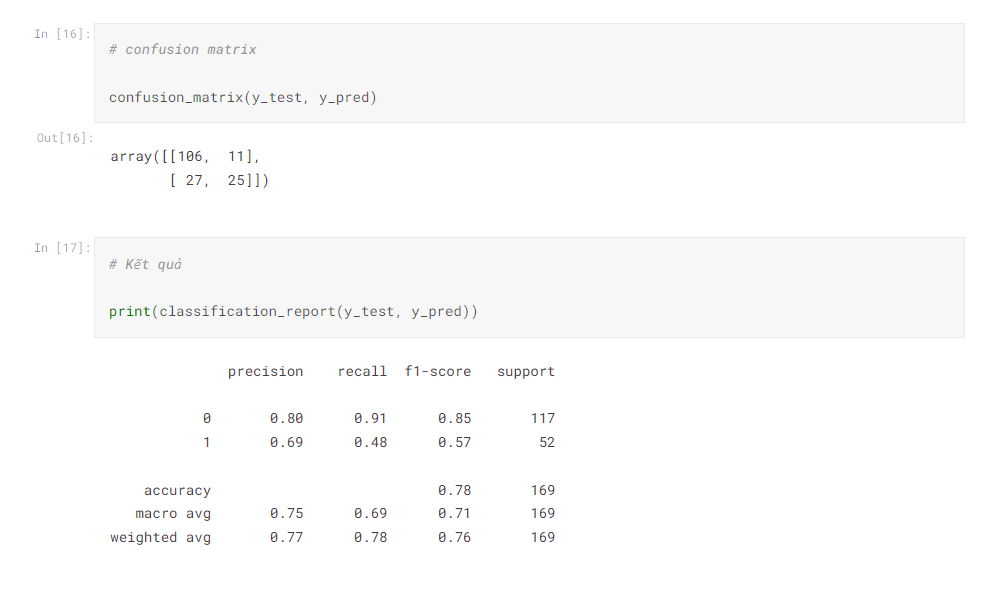


I – Sử dụng Machine Learning

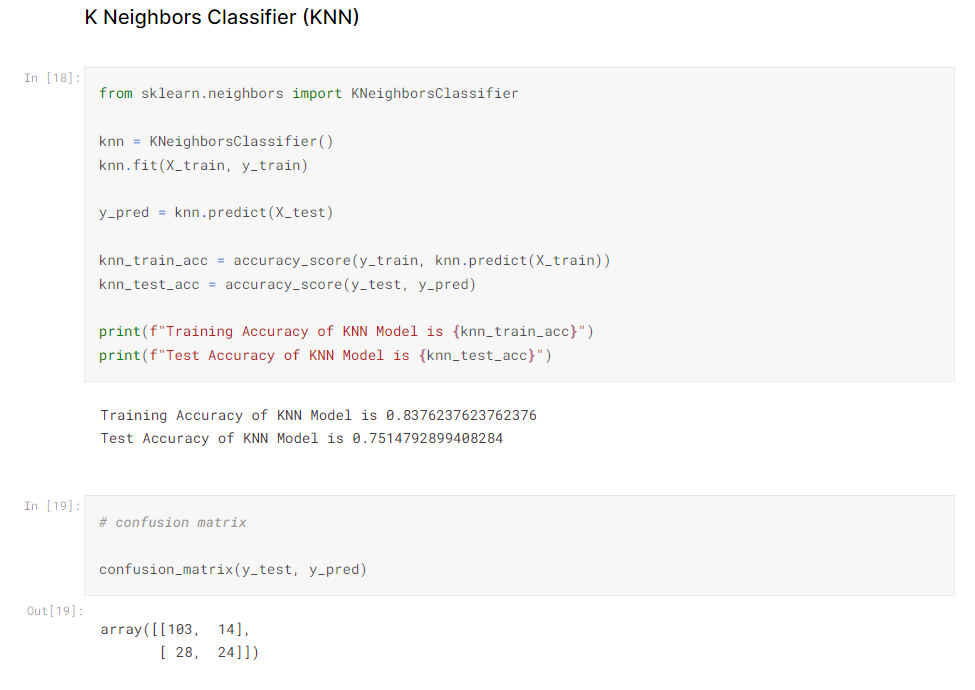
1. Sử dụng Logistic Regression



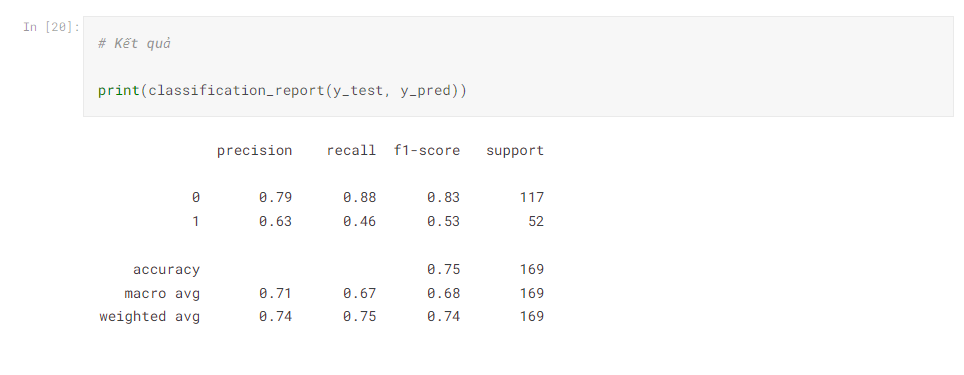
Kết quả

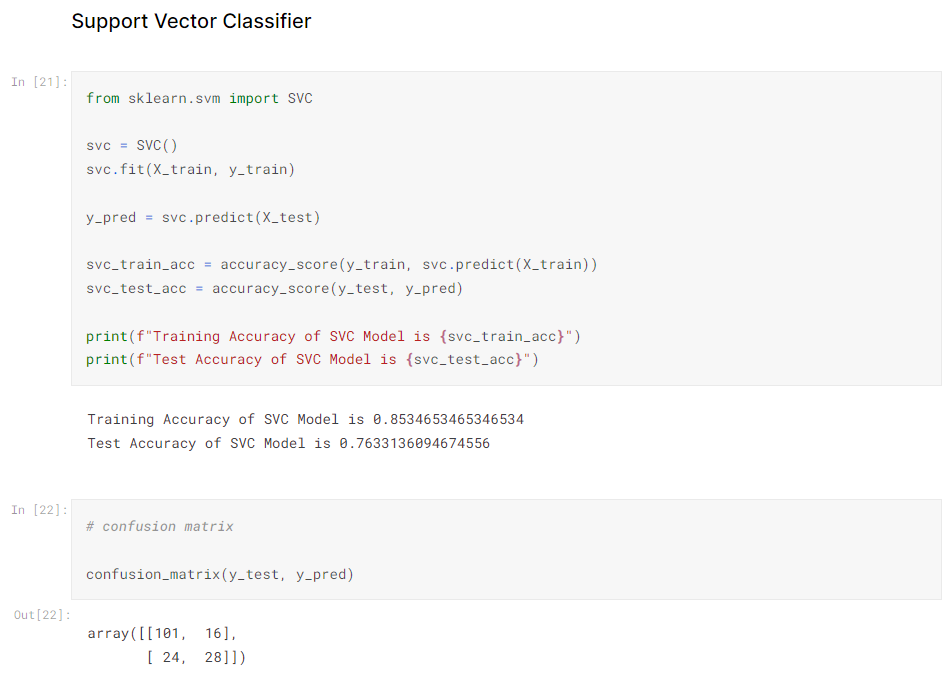


2. Sử dụng KNN

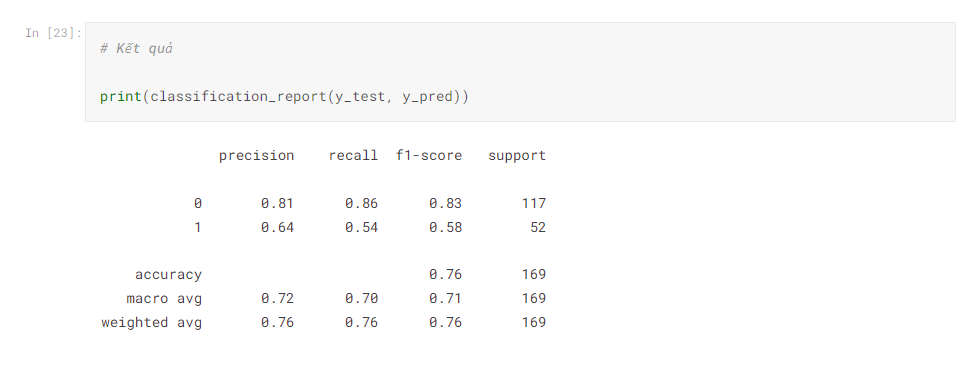


Kết quả



3. Sử dụng SVM 

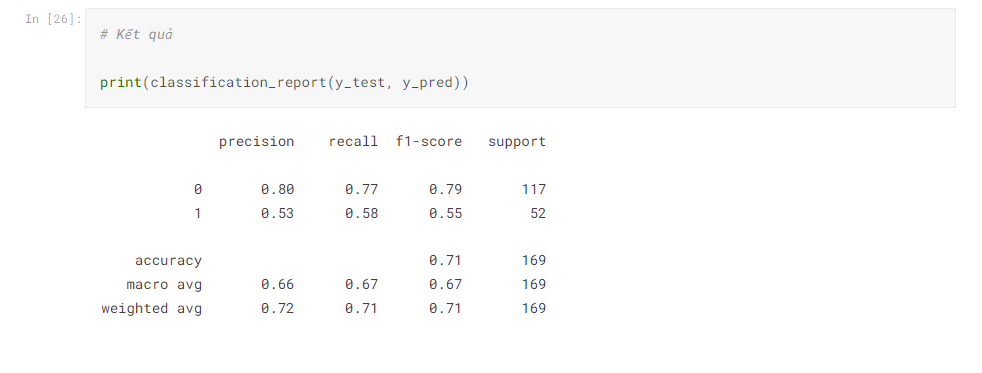
Kết quả



4. Sử dụng Decision Tree

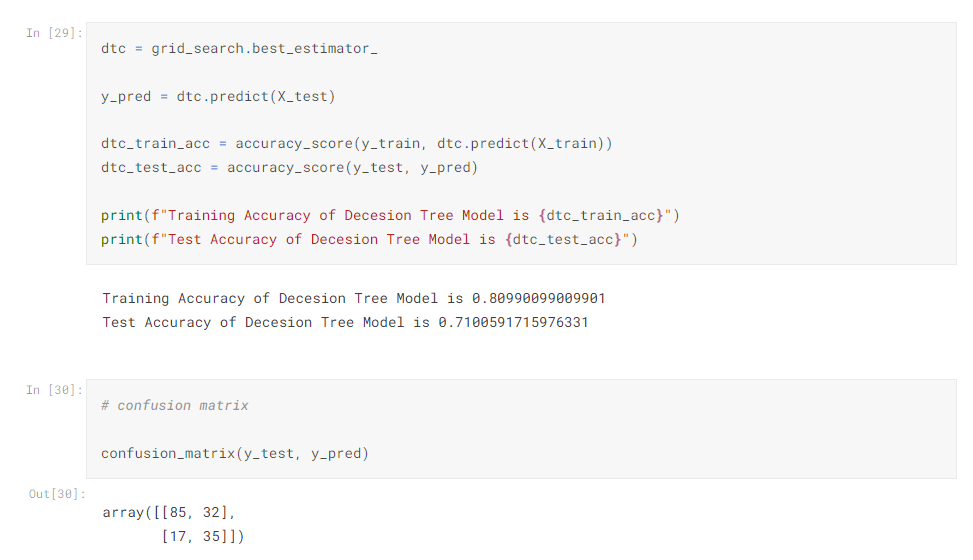


Kết quả

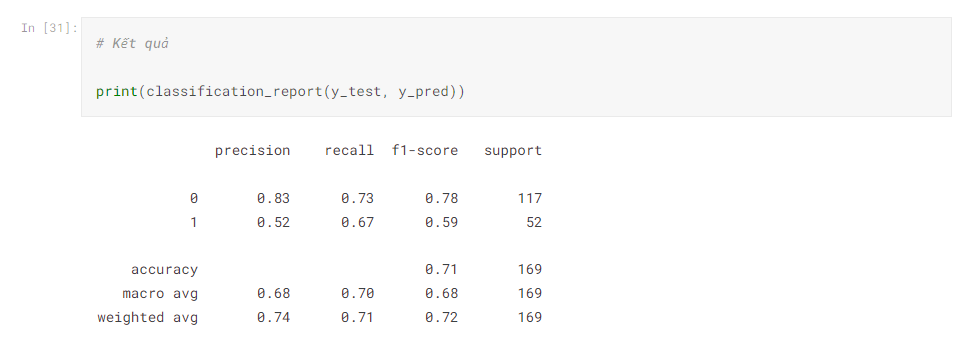


5. Sử dụng Grid Search CV

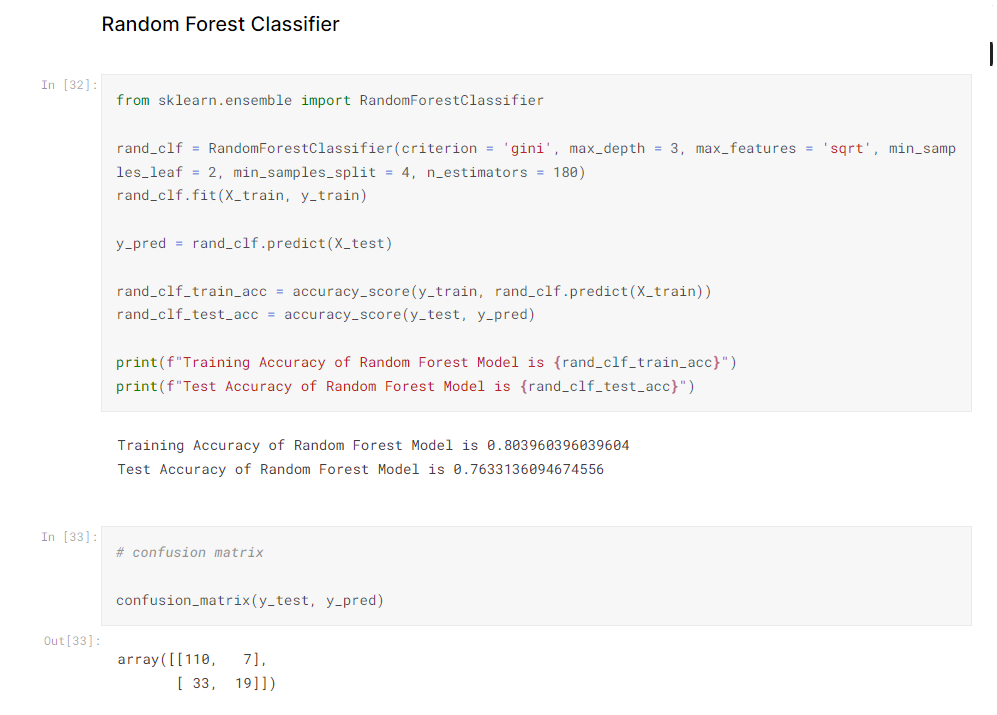




Kết quả



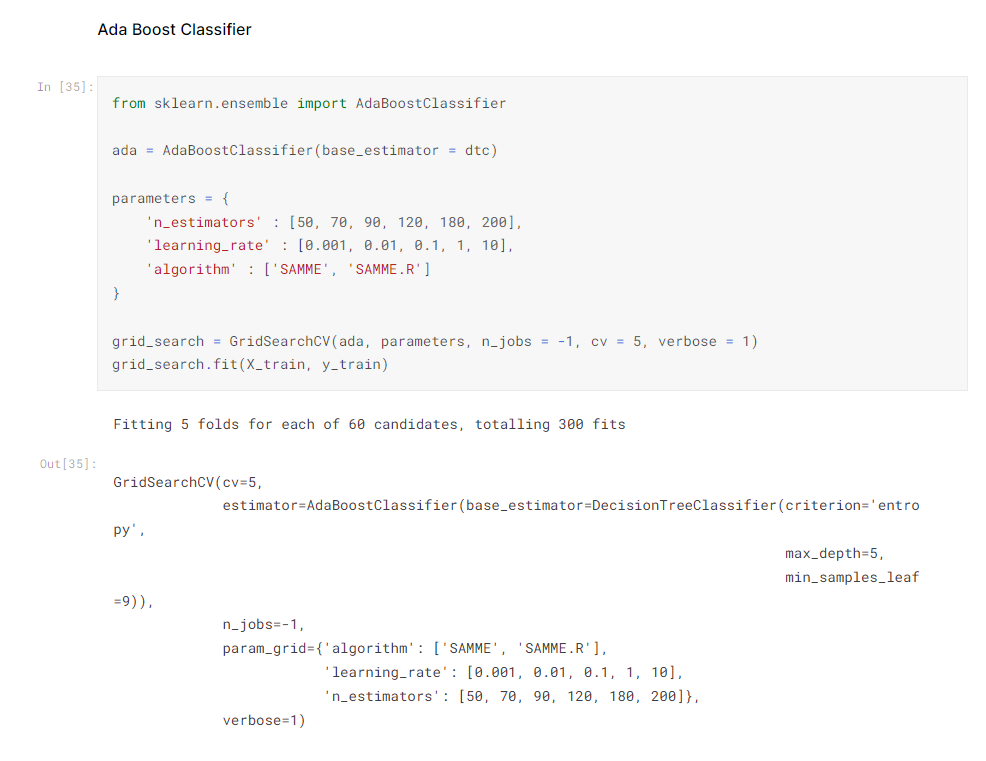
6. Sử dụng Random Forest

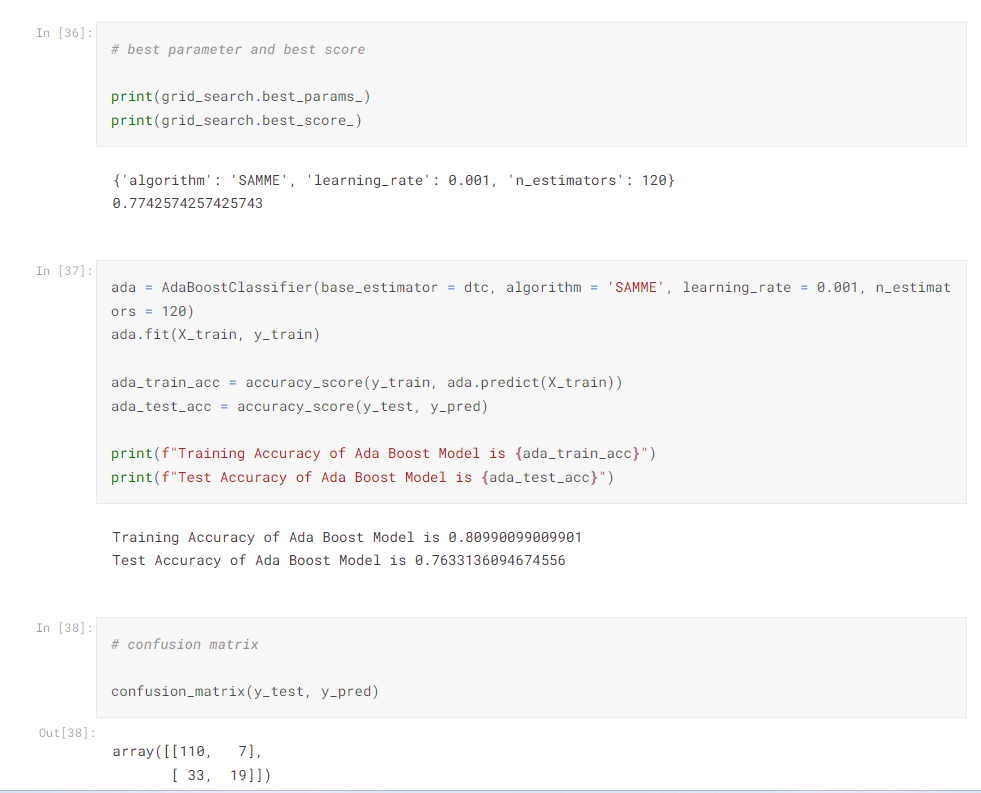


Kết quả



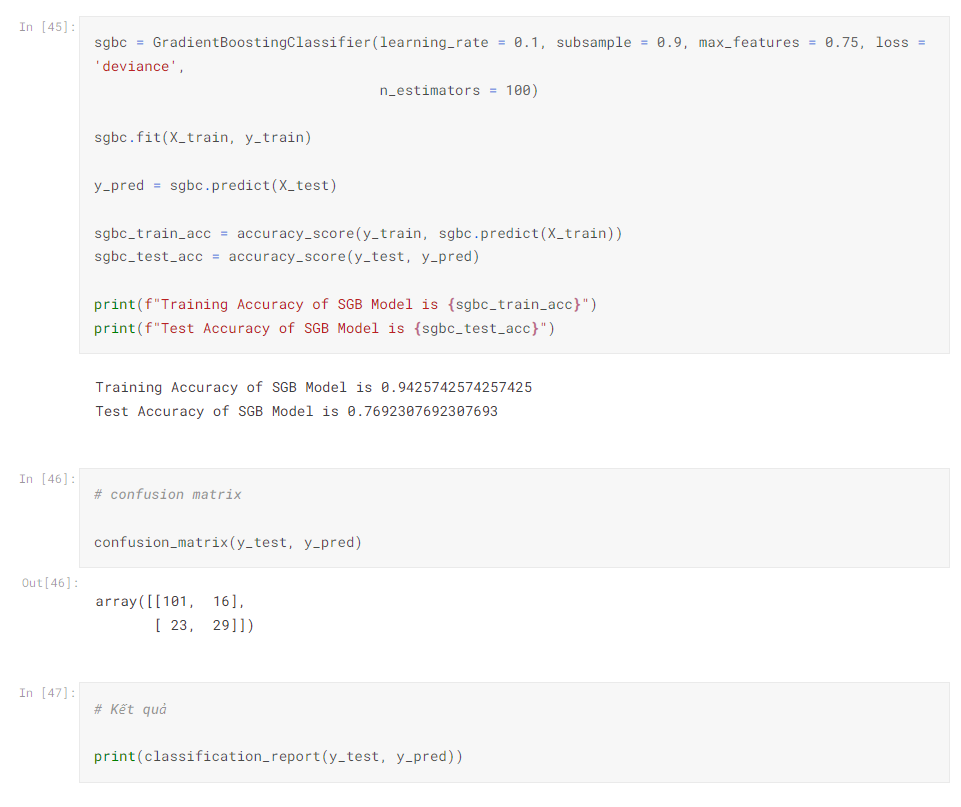
7. Sử dụng Adaboost



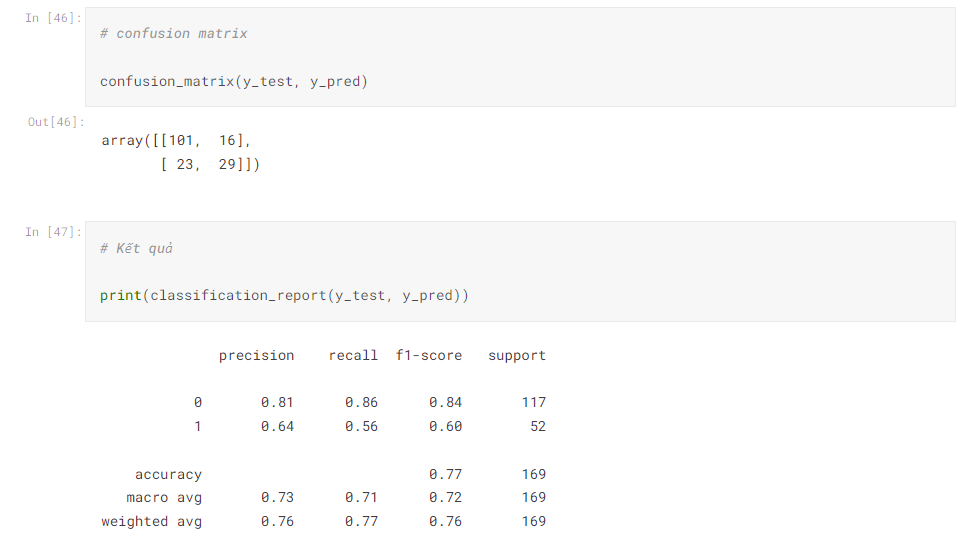




8. Sử dụng Stochastic Gradient Boosting

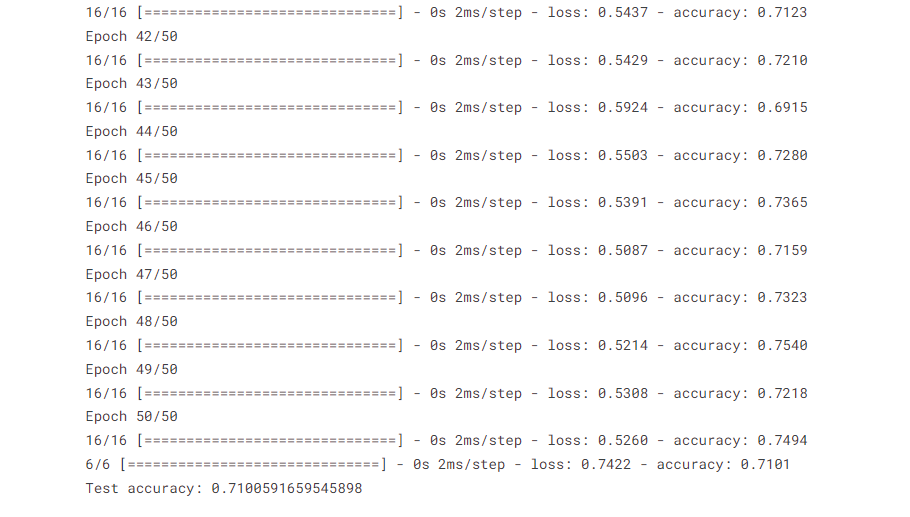


Kết quả



II – Sử dụng Deep Learning

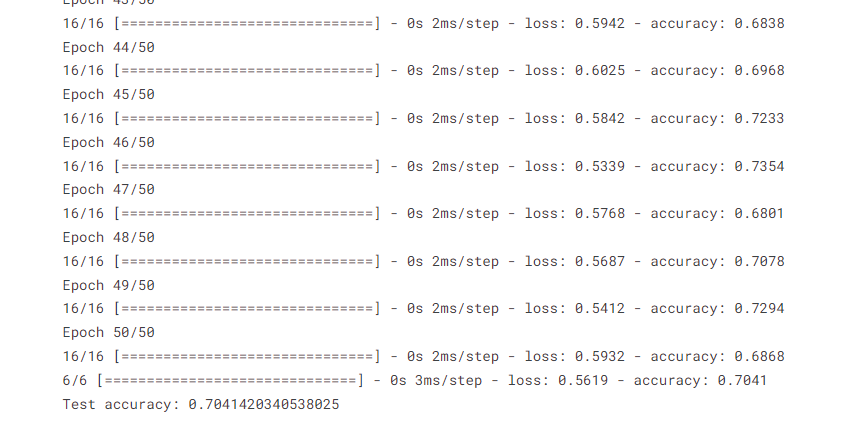
1. Model đầu tiên 

Kết quả 

2. Thử một cách xây dựng khác sâu hơn với nhiều lớp

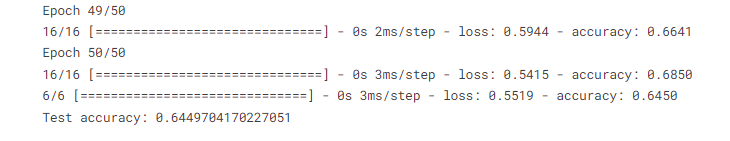


Kết quả



3. Thử một model khác, sử dụng Conv1D và MaxPooling

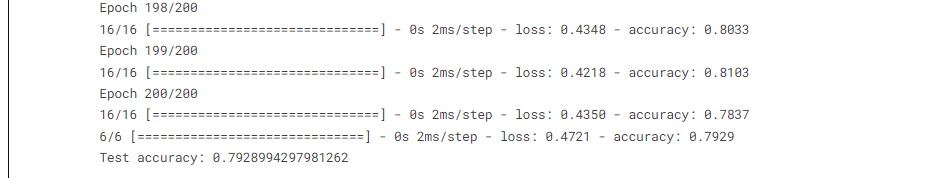




4. Model đầu tiên, tăng số lượng epoch lên 200



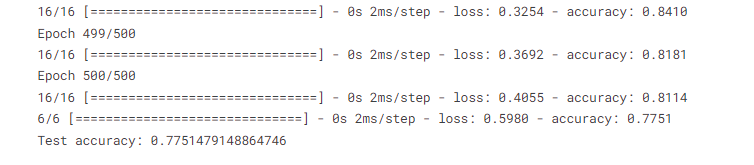
Kết quả



5. Model thứ 2, tăng số epoch lên 500



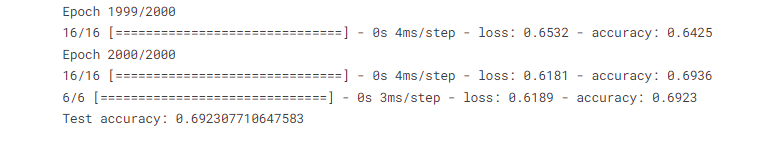
Kết quả



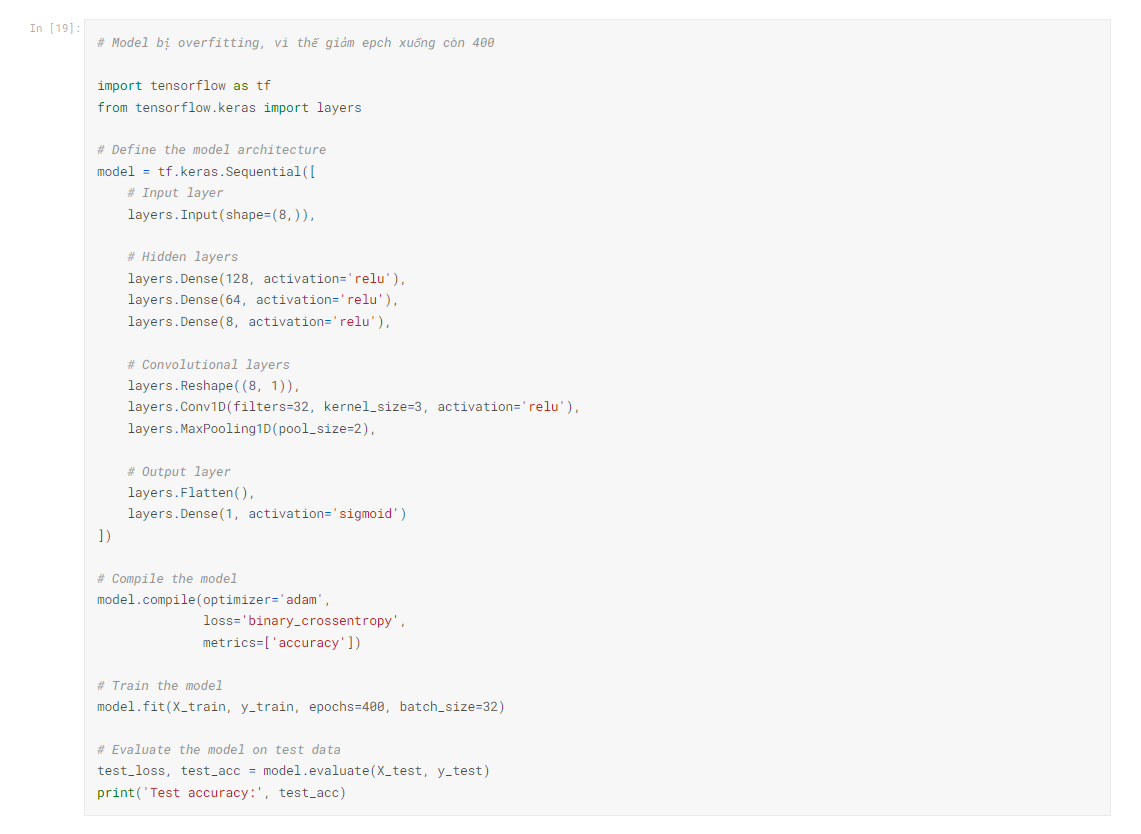
6. Model thứ ba, nhưng với epoch = 2000



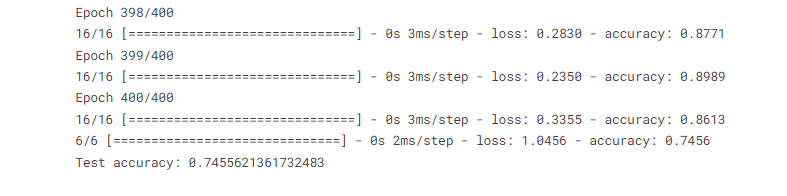
Kết quả:



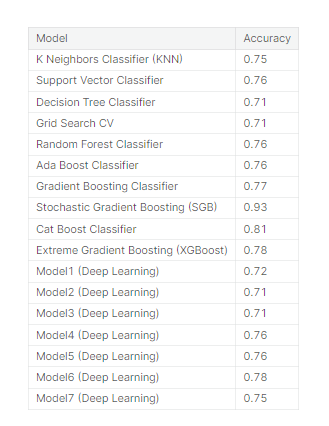
7. Model bị overfitting, vì thế giảm epoch xuống còn 400



Kết quả



III – Bảng so sánh



IV – Kết luận

Dựa trên bảng so sánh, ta thấy các mô hình deep learning không vượt trội hơn so với các mô hình machine learning truyền thống trong bài toán dự đoán tiểu đường trên tập dữ liệu này. Trong số các mô hình deep learning, Model6 đạt kết quả tốt nhất với accuracy là 0.78, tương đương với Gradient Boosting Classifier của machine learning. Tuy nhiên, mô hình Stochastic Gradient Boosting (SGB) của machine learning vẫn là mô hình cho kết quả tốt nhất với accuracy là 0.93. Ta có thể thấy rằng, việc lựa chọn mô hình phù hợp với bài toán là rất quan trọng, và không phải lúc nào cũng cần phải sử dụng các mô hình deep learning phức tạp.

Bài 7:

7.1:

Bài viết trên trang monkeylearn.com giới thiệu về phân tích văn bản, đó là quá trình sử dụng công nghệ để tách, phân tích và hiểu nội dung của một đoạn văn bản. Bài viết giới thiệu các công nghệ phân tích văn bản như sentiment analysis, topic modeling, keyword extraction và text classification. Cụ thể, sentiment analysis là quá trình phân tích ý kiến, đó là cảm nhận của người viết, về một chủ đề hoặc một sản phẩm. Topic modeling là phân tích nội dung của một đoạn văn bản để tìm ra các chủ đề hoặc chủ đề chính trong đó. Keyword extraction là quá trình tìm ra các từ khóa quan trọng nhất trong một đoạn văn bản. Cuối cùng, text classification là quá trình phân loại các đoạn văn bản vào các danh mục được xác định trước. Bài viết cũng giới thiệu các ứng dụng của phân tích văn bản trong thực tế, bao gồm marketing, phân tích đánh giá sản phẩm và phát hiện tin giả.

Phần Text Analysis Basics trình bày khái niệm cơ bản về phân tích văn bản và các ứng dụng của nó. Bài viết giới thiệu cách xử lý văn bản bằng các phương pháp khác nhau, bao gồm phân tích cảm xúc, phân loại văn bản, rút trích thông tin và tóm tắt văn bản. Ngoài ra, bài viết cũng giới thiệu một số công cụ phân tích văn bản phổ biến như NLTK, Gensim và MonkeyLearn. Cuối cùng, bài viết đề cập đến một số thách thức khi xử lý văn bản, bao gồm độ chính xác của kết quả, độ dài của văn bản và đa ngôn ngữ.

Trong phần Methods & Techniques, bài viết giới thiệu các phương pháp và kỹ thuật phân tích văn bản như:

Keyword Extraction: Trích xuất các từ khóa quan trọng từ văn bản để hiểu nội dung chính của văn bản.

Sentiment Analysis: Phân tích cảm xúc của văn bản để xác định tâm trạng hoặc ý kiến của người viết.

Named Entity Recognition: Phân tích và nhận dạng các tên thực thể trong văn bản như tên riêng, địa điểm, thời gian, v.v.

Text Classification: Phân loại văn bản theo chủ đề hoặc thể loại khác nhau.

Topic Modeling: Tìm ra các chủ đề chính trong văn bản.

Text Summarization: Tóm tắt văn bản thành những đoạn văn bản ngắn gọn và dễ hiểu hơn.

Word Embedding: Chuyển đổi các từ thành vector trong không gian đa chiều để xử lý và phân tích văn bản.

Bài viết cũng đề cập đến một số công cụ và thư viện phổ biến được sử dụng để thực hiện các phương pháp và kỹ thuật trên, chẳng hạn như MonkeyLearn, NLTK, SpaCy, Gensim, v.v.

Bài viết "How Does Text Analysis Work?" giải thích về cách phân tích văn bản hoạt động bằng cách sử dụng phân tích cú pháp và phân tích ngữ nghĩa. Phân tích cú pháp sử dụng các công cụ như trích xuất từ khoá, tách câu và phân tích cú pháp để hiểu cấu trúc ngữ pháp của văn bản. Phân tích ngữ nghĩa bao gồm phân tích ý nghĩa của từng từ, cụm từ, câu hoặc đoạn văn bản để hiểu ý nghĩa tổng thể của văn bản.

Bài viết cũng giải thích về các kỹ thuật phân tích văn bản phổ biến bao gồm: phân tích tần số từ, phân tích cảm xúc, phân tích quan điểm và phân tích thực thể. Ngoài ra, bài viết còn giới thiệu về các công cụ và thư viện phân tích văn bản phổ biến như MonkeyLearn, TextBlob, NLTK, SpaCy và Gensim.

Bài viết "How to Analyze Text Data" trình bày một số phương pháp phổ biến để phân tích dữ liệu văn bản, bao gồm:

Preprocessing: Tiền xử lý dữ liệu bao gồm các bước loại bỏ các ký tự đặc biệt, chuyển đổi chữ hoa thành chữ thường, loại bỏ từ dừng và các bước xử lý khác để chuẩn bị dữ liệu cho các phương pháp phân tích tiếp theo.

Sentiment Analysis: Phân tích cảm xúc trong văn bản, bao gồm phát hiện ý kiến, cảm xúc tích cực và tiêu cực.

Topic Modeling: Xác định các chủ đề trong tập dữ liệu văn bản bằng cách sử dụng các phương pháp như Latent Dirichlet Allocation (LDA).

Named Entity Recognition: Nhận dạng các thực thể được đề cập trong văn bản, bao gồm người, địa điểm và sản phẩm.

Text Classification: Phân loại văn bản vào các danh mục đã biết trước.

Word Embeddings: Chuyển đổi các từ thành các vector số để dễ dàng xử lý và tính toán.

Bài viết cũng cung cấp một số công cụ và thư viện phổ biến để thực hiện các phương pháp này trên dữ liệu văn bản.

7.2 :

Em xin phép gửi link chứa code dưới dạng notebook, thay vì gửi toàn bộ code ở đây. Link: <https://colab.research.google.com/drive/10YfjOrNq3sDJp0Q-Bne9jxDiKjrsX-Qm?usp=sharing>

Code được upload từ notebook đã chạy sẵn trên máy local nên sẽ có sẵn file data. Khi chạy lại trên colab sẽ bị lỗi do thiếu file data

7.4

Em xin phép được gửi link notebook chứa code, thay vì gửi toàn bộ code ở đây. Link: <https://colab.research.google.com/drive/1ajUHNXQgCG7Gal6Ovw-213Crmr1IXcUX?usp=sharing>

Code được upload từ notebook đã chạy sẵn trên máy local nên sẽ có sẵn file data. Khi chạy lại trên colab sẽ bị lỗi do thiếu file data

Bài 8:

8.1:

Trang web này cung cấp một bài giới thiệu về học sâu cho xử lý ảnh. Nó trình bày khái niệm cơ bản của mạng nơ-ron tích chập (CNN) và giải thích cách thức hoạt động của chúng trong việc phân loại ảnh và phát hiện đối tượng. Nó cũng trình bày về các bộ dữ liệu được sử dụng trong học sâu cho xử lý ảnh và đưa ra một số ví dụ về ứng dụng của học sâu trong lĩnh vực xử lý ảnh như phát hiện khuôn mặt, phát hiện đối tượng, phân loại ảnh, và xử lý ảnh y tế.

Phần "Traditional image recognition vs. deep learning" so sánh giữa hai phương pháp nhận diện hình ảnh truyền thống và deep learning. Truyền thống sử dụng các thuật toán như phân tách vùng, phân tích màu sắc và kích thước để nhận dạng hình ảnh, trong khi deep learning sử dụng các mạng nơ-ron sâu để học cách trích xuất đặc trưng từ hình ảnh và giải quyết các vấn đề phức tạp hơn như nhận diện khuôn mặt và đối tượng. Deep learning cũng có thể học và cải thiện kết quả theo thời gian và có thể được áp dụng trong nhiều lĩnh vực khác nhau, bao gồm xử lý ảnh y tế, nhận dạng giọng nói và phân loại văn bản.

Phần "What deep learning networks 'learn'" trình bày về cách mà các mạng nơ-ron sâu (deep learning networks) học các đặc trưng của dữ liệu hình ảnh và sử dụng chúng để phân loại hoặc phân đoạn hình ảnh. Các mạng nơ-ron sâu học từ dữ liệu bằng cách điều chỉnh các trọng số của các lớp nơ-ron và tối ưu hóa một hàm mất mát (loss function) để đưa ra các dự đoán chính xác. Các đặc trưng này thường được học tự động và trừu tượng hóa từ dữ liệu, không cần quan tâm đến việc lập trình trực tiếp các đặc trưng này. Các mô hình nơ-ron sâu cũng có thể được sử dụng để trích xuất các đặc trưng để sử dụng trong các mô hình khác hoặc để phân tích dữ liệu hình ảnh.

Phần "The evolution of deep learning" trình bày về sự phát triển của deep learning và một số bước tiến quan trọng trong lịch sử của nó. Bắt đầu từ khởi đầu của neural network, cụ thể là perceptron, qua các giai đoạn đầu thập niên 2000, deep learning đã trải qua nhiều cải tiến như học chuyển tiếp, mạng neuron tích chập (CNN), và mạng nơ-ron hồi quy (RNN). Bằng cách tận dụng những cải tiến này, deep learning đã đạt được những kết quả đáng kinh ngạc trong nhiều lĩnh vực, đặc biệt là trong lĩnh vực xử lý ảnh và thị giác máy tính.

[Perceptron và Convolutional Neural Network (CNN) là hai kiến trúc mạng nơ-ron trong deep learning thường được sử dụng để giải quyết các bài toán xử lý ảnh.]

[Perceptron là một kiến trúc mạng nơ-ron đơn giản bao gồm một hoặc nhiều lớp nơ-ron liên kết một cách tuần tự với nhau. Mỗi nơ-ron trong lớp đầu vào nhận dữ liệu đầu vào, tính toán và truyền tiếp cho lớp tiếp theo. Đầu ra của mỗi nơ-ron được tính bằng cách áp dụng một hàm kích hoạt non-linear, như hàm sigmoid, relu, hay tanh. Perceptron thường được sử dụng trong các bài toán phân loại nhị phân hoặc phân loại đa lớp.]

[CNN là một kiến trúc mạng nơ-ron có khả năng học và phân tích các đặc trưng ảnh một cách hiệu quả. Kiến trúc của CNN bao gồm các lớp convolutional layer, pooling layer, và fully connected layer. Convolutional layer sử dụng các bộ lọc để trích xuất các đặc trưng của ảnh, sau đó áp dụng hàm kích hoạt non-linear để tạo ra các feature maps. Pooling layer được sử dụng để giảm kích thước của feature maps và giảm overfitting. Cuối cùng, fully connected layer được sử dụng để phân loại các đối tượng trong ảnh. CNN thường được sử dụng trong các bài toán nhận dạng đối tượng, phân loại ảnh, và xử lý ảnh y tế.]

Phần "Recent trends" trình bày những xu hướng gần đây trong lĩnh vực deep learning cho xử lý ảnh, bao gồm:

Transfer learning: sử dụng các mô hình đã được huấn luyện trước đó (như trong bài tập này) để tạo ra mô hình mới với ít dữ liệu hơn.

Autoencoders: một loại mô hình deep learning có khả năng học cách biểu diễn dữ liệu mà không cần nhãn, thường được sử dụng để giảm chiều dữ liệu.

Generative adversarial networks (GANs): một loại mô hình deep learning có khả năng tạo ra dữ liệu giả mạo có chất lượng tương đương với dữ liệu thực.

Neural style transfer: một kỹ thuật sử dụng deep learning để tạo ra ảnh mới với phong cách của ảnh khác, thường được sử dụng để tạo ra ảnh nghệ thuật.

Attention mechanisms: một kỹ thuật sử dụng deep learning để chọn ra những phần quan trọng trong dữ liệu đầu vào, thường được sử dụng để giải quyết vấn đề trong xử lý ngôn ngữ tự nhiên, nhưng cũng có thể áp dụng cho xử lý ảnh.

Phần "Keeping up with new developments" nói về cách cập nhật và theo kịp những xu hướng mới trong lĩnh vực học sâu. Bài viết giới thiệu một số tài nguyên trực tuyến để học tập và cập nhật kiến thức như các trang web, blog và tài liệu tham khảo, cùng với việc tham gia các cộng đồng và sự kiện liên quan đến học sâu. Ngoài ra, bài viết cũng nhấn mạnh tầm quan trọng của việc xây dựng các dự án thực tế để áp dụng và rèn luyện kỹ năng trong lĩnh vực này.

8.2: Em xin phép được gửi link notebook, thay vì gửi toàn bộ code ở đây. Link colab: <https://colab.research.google.com/drive/1JjN27i9n1GTWHveGWjkGQrufbzDd69ce>

Một vài chú thích cho link notebook:  
Trang web TensorFlow hướng dẫn cách xây dựng một mô hình nhận diện hình ảnh sử dụng thư viện TensorFlow. Bài hướng dẫn bao gồm các bước sau:

Chuẩn bị dữ liệu: Tải xuống bộ dữ liệu "Hình ảnh thực vật" từ TensorFlow Datasets và tiền xử lý dữ liệu để chuẩn bị cho việc đào tạo mô hình.

Xây dựng mô hình: Sử dụng API Sequential của TensorFlow để xây dựng một mô hình mạng nơ-ron sâu. Mô hình bao gồm các lớp Conv2D và MaxPooling2D để trích xuất đặc trưng từ hình ảnh và lớp Dense để phân loại.

Đào tạo mô hình: Sử dụng phương pháp đào tạo với hàm mất mát categorical crossentropy và thuật toán tối ưu hóa Adam để tối đa hóa độ chính xác của mô hình.

Đánh giá mô hình: Đánh giá mô hình bằng cách sử dụng tập dữ liệu xác thực và tính toán độ chính xác và ma trận lỗi.

Dùng mô hình để dự đoán: Sử dụng mô hình đã huấn luyện để dự đoán lớp của các hình ảnh chưa biết.

Bài hướng dẫn này cũng giới thiệu các khái niệm cơ bản liên quan đến xây dựng mô hình nhận diện hình ảnh, như chuẩn hóa dữ liệu, tăng cường dữ liệu và sử dụng bộ dữ liệu xác thực để kiểm tra mô hình.

tf.data.AUTOTUNE là một tham số trong TensorFlow được sử dụng để tối ưu hóa hiệu suất đọc dữ liệu. Khi sử dụng tf.data.Dataset, việc tải dữ liệu từ đĩa có thể là một lần chặn đối với các tập dữ liệu lớn, làm cho quá trình đào tạo mất thời gian và tốn kém. tf.data.AUTOTUNE cho phép TensorFlow tự động chọn số lượng luồng đọc dữ liệu và kích thước đệm dữ liệu phù hợp với tập dữ liệu và phần cứng hiện có, giúp cải thiện hiệu suất đọc dữ liệu và giảm thời gian huấn luyện mô hình.

Adam có khả năng hội tụ nhanh và hiệu quả trong các bài toán có nhiều tham số, và nó là một lựa chọn phổ biến trong các mô hình deep learning.

8.3: Trang web hướng dẫn cách tải dữ liệu hình ảnh trong TensorFlow sử dụng tf.data.Dataset. Bài hướng dẫn trình bày cách tải dữ liệu từ tệp .zip hoặc tải dữ liệu từ url. Sau đó, nó hiển thị cách chuẩn bị dữ liệu hình ảnh cho việc huấn luyện mô hình và cách tạo và sử dụng tf.data.Dataset để tạo một chuỗi đào tạo. Cuối cùng, nó cung cấp ví dụ về cách huấn luyện mô hình với tf.keras bằng cách sử dụng ImageDataGenerator và flow\_from\_directory để tạo một Dataset.

Em xin phép gửi link google colab chứa code của bài 8.3: <https://colab.research.google.com/drive/1wro20R6vBlwJiA9ua5xrwvvQD_-K1oVb>

8.4: Em xin phép được gửi link colab chứa code của bài 8.4:

<https://colab.research.google.com/drive/1weMfr3egCQu2ijAb0Q805BUhVq3lIoi9>