# ĐẠI HỌC BÁCH KHOA HÀ NỘI KHOA TOÁN – TIN



BÁO CÁO BÀI TẬP LỚN CUỐI KỲ

# HỌC PHẦN KĨ THUẬT LẬP TRÌNH - MI3310

# Chủ đề 6: Xây dựng chương trình làm bài trắc nghiệm

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | : TS . Vũ Thành Nam |
| Họ và tên sinh viên | : Nguyễn Doãn Anh Minh |
| MSSV | : 20227200 |
| Lớp | : Hệ thống thông tin 01-K67 |
| Mã lớp | : 150327 |
| Học kì | : 20232 |

*Hà Nội, ngày 16 tháng 6 năm 2024*

0

*Project cuối kì – Học phần Kỹ thuật lập trình – MI3310 Chủ đề 6 – Xây dựng chương trình làm bài thi trắc nghiệm Họ và tên : Nguyễn Doãn Anh Minh*

*MSSV : 20227200*

*Lớp : Hệ thống thông tin quản lý 01- K67 Mã lớp : 150327*

*Email:* [*minh.nda227200@sis.hust.edu.vn*](mailto:minh.nda227200@sis.hust.edu.vn)

## Mục lục

[Lời nói đầu 2](#_bookmark0)

1. [Mô tả cấu trúc chương trình. 3](#_bookmark1)
   1. [Chủ để lựa chọn, mô tả bài toán. 3](#_bookmark2)
   2. [Cấu trúc của chương trình. 3](#_bookmark3)
2. [Các kĩ thuật đã sử dụng 5](#_bookmark4)
   1. [Các nguyên lý/ hình mẫu lập trình đã được áp dụng. 5](#_bookmark5)
   2. [Các giải thuật đã được áp dụng 7](#_bookmark6)
   3. [Các kĩ thuật thiết kế chương trình. 7](#_bookmark7)
      1. [Các nguyên tắc thiết kế cơ bản đã được áp dụng 7](#_bookmark8)
      2. [Các thiết kế giải thuật đã được áp dụng 9](#_bookmark9)
         1. [Cơ chế top-down và modularity 9](#_bookmark10)
         2. [Kỹ thuật xác định hàm, thủ tục, tham số 9](#_bookmark11)
         3. [Sử dụng biến 10](#_bookmark12)
3. [Chức năng chương trình 11](#_bookmark13)
   1. [Chức năng thêm người dùng admin. 11](#_bookmark14)
   2. [Chức năng đăng kí đăng nhập (phân quyền admin – user) 11](#_bookmark15)
   3. [Chức năng của admin 13](#_bookmark16)
   4. [Chức năng của user 17](#_bookmark17)
4. [Kiểm thử chương trình và kết quả chạy. 19](#_bookmark18)
   1. [Kiểm thử, gỡ lỗi. 19](#_bookmark19)
   2. [Chạy và phát hành chương trình. 21](#_bookmark20)
5. [Đề xuất nâng cấp. 22](#_bookmark21)
6. [Mã nguồn. 25](#_bookmark22)
7. [Tổng kết 32](#_bookmark23)
8. [Tài liệu tham khảo 33](#_bookmark24)

# Lời nói đầu

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc đến thầy TS Vũ Thành Nam. Sự giảng dạy, hỗ trợ và tạo điều kiện của thầy đã giúp nhóm em hoàn thành tốt nhất báo cáo này.

***Về học phần Kĩ thuật lập trình - MI3310:*** Qua môn học Kỹ thuật lập trình, em đã hiểu rõ hơn về các hành động, thao tác và lối suy nghĩ của bản thân khi lập trình. Những điều này, từ vô thức nay đã được diễn giải thành lời, giúp em nắm bắt chi tiết hơn về trình tự, mục đích, cũng như ưu và khuyết điểm của các hoạt động lập trình.

***Về Project này:*** Khi thầy có bật mí về các chủ đề, em đã rất có hứng thú với chủ đề làm app trắc nghiệm và thật may mắn sau khi được phân bố chủ đề thì em đã may mắn nhận được chủ đề này. Đây cũng là 1 dự án mà em đã đặt nhiều tâm huyết, thời gian và công sức để hoàn thành một cách chỉnh chu nhất. Dù chương trình của em không đạt đến mức độ hoàn thiện như các chương trình chuyên về làm bài trắc nghiệm, nhưng qua dự án lần này, em đã học được cách vận hành và hoạt động của các chương trình tương tự.

Trong quá trình thực hiện, khó tránh khỏi về những sai sót và các thuật toán chưa tối ưu. Em rất mong nhận được bỏ qua cho những sai sót sự góp ý từ thầy, để bài làm của em có thể được hoàn thiện và tối ưu hơn nữa.

Em xin chân thành cảm ơn!

|  |
| --- |
| *Hà Nội, ngày 16 tháng 6 năm 2024*  Sinh viên thực hiện |
| Nguyễn Doãn Anh Minh |

## Mô tả cấu trúc chương trình.

### Chủ để lựa chọn, mô tả bài toán.

*Chủ đề lựa chọn :* Xây dựng chương trình làm bài trắc nghiệm.

*Mô tả bài toán :* Chương trình đọc dữ liệu câu hỏi và đáp án từ một file cho trước sau đó hiển thị câu hỏi theo thứ tự ngẫu nhiên ( không lặp lại), cuối cùng in ra kết quả :

* + - File cauhoi.bin : ngân hàng câu hỏi thi gồm: mã câu hỏi, câu hỏi, 4 phương án, phương án đúng.
    - Quản lý câu hỏi thi (thêm mới, sửa, xóa) câu hỏi thi (không có 2 câu hỏi trùng mã).
    - Sinh đề thi (chọn ngẫu nhiên N câu hỏi, N xác định trước) và cho thi.
    - Hiển thị kết quả bài thi và cho phép lưu bài làm + kết quả.

*Các hạng mục đã thực hiện :* Tất cả các yêu cầu của thầy đã được em thực hiện trong bài báo cáo. Ngoài ra em đã thêm 1 số phần để hoàn thiện chương trình hơn.

* + - Sinh mã đề 1 cách ngẫu nhiên khi sinh viên làm bài để thuận tiện cho việc quản lý.
    - Phân quyền người dùng, admin. Thêm các file để lưu dữ liệu người dùng.
    - Đảo câu hỏi, đảo đáp án cho mỗi đề khác nhau cho sinh viên làm bài thi.
    - Chức năng phê duyệt người dùng.
    - Và 1 số chức năng khác sẽ được mô tả kĩ hơn ở trong phần các chức năng của chương trình.

Công cụ lập trình : Visual Studio Code.

### Cấu trúc của chương trình.

Đầu tiên em đã phát triển chương trình của em có cấu trúc như sau : quiz/

├── data/

│ ├── questions.bin

│ ├── users.bin

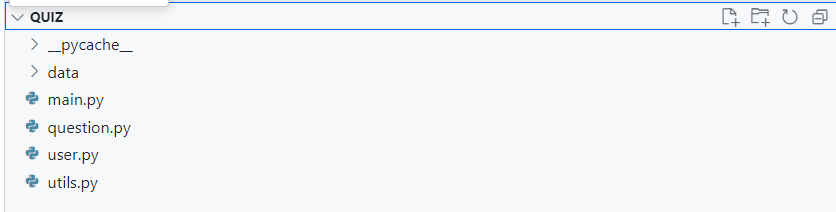
│ └── các file chứa dữ liệu điểm thi/ bài làm của người dùng.

├── question.py # gồm các hàm xử lý câu hỏi.

├── user.py # gồm các hàm xử lý người dùng.

├── utils.py #gồm các hàm tiện ích.

└── main.py # hàm chạy chương trình.



Sau đó nhận thấy để cho tiện quản lý và nâng cấp sau này, em đã tìm hiểu và sửa đổi mã nguồn theo mô hình MVC (Model, Views, Controllers): MVC\_quizupdate/

├── data/

│ ├── questions.bin # Ngân hàng đề.

│ ├── users.bin # Dữ liệu người dùng.

│ ├── admin.bin #dữ liệu admin.

│ └── results/ #Lưu kết quả thi của học sinh/ sinh viên.

├── models/

│ ├── question.py # quản lý các câu hỏi.

│ ├── user.py # quản lý người dùng.

│ └── result.py #quản lý các bài kiểm tra.

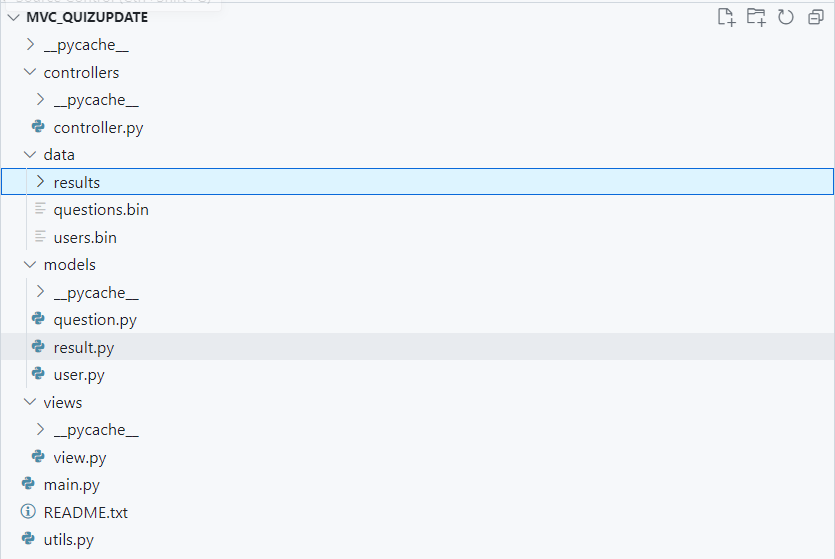
├── views/

│ └── view.py #quản lý giao diện.

├── controllers/

│ └── controller.py #điều phối hoạt động giữa model và view.

├── utils.py # quản lý các hàm tiện ích.

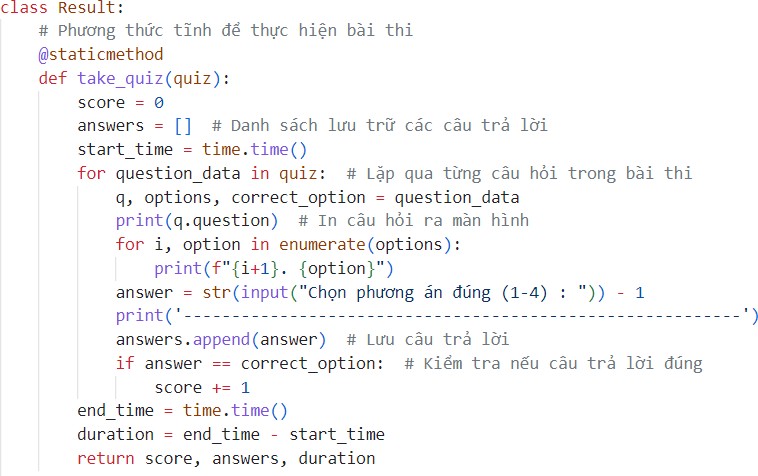
└── main.py # hàm chính để chạy chương trình.

## Các kĩ thuật đã sử dụng .

### Các nguyên lý/ hình mẫu lập trình đã được áp dụng.

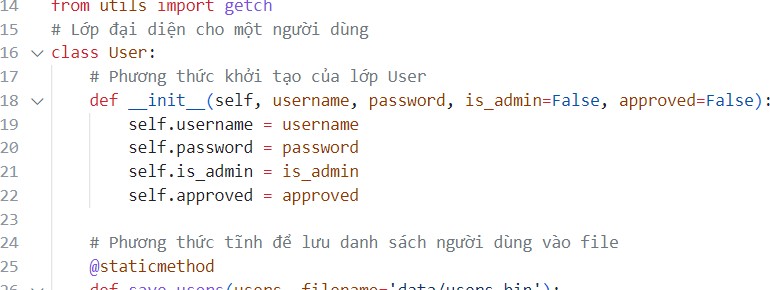
Các hình mẫu lập trình chủ yếu được áp dụng trong Project này gồm 3 hình mẫu :

*+ Lập trình mệnh lệnh :* Lập trình mệnh lệnh tập trung vào việc mô tả các bước mà chương trình cần thực hiện để đạt được mục tiêu mong muốn. Các bước này thường được thể hiện qua các câu lệnh tuần tự.

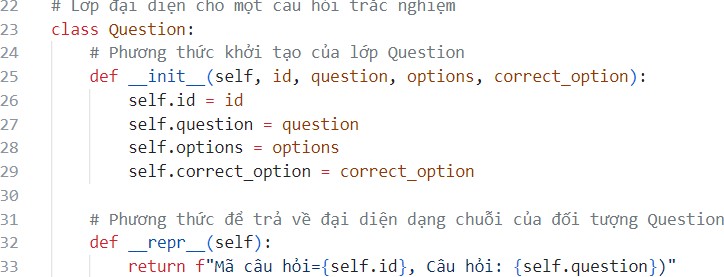
*Ví dụ*: Trong phương thức take\_quiz của lớp Result, chúng ta mô tả tuần tự các bước để thực hiện bài kiểm tra:

*+ Lập trình hướng đối tượng:* Lập trình hướng đối tượng tập trung vào việc sử dụng các đối tượng để đại diện cho dữ liệu và các phương thức để thao tác với dữ liệu đó. Ứng dụng trắc nghiệm của chúng ta sử dụng lập trình hướng đối tượng một cách rõ ràng.

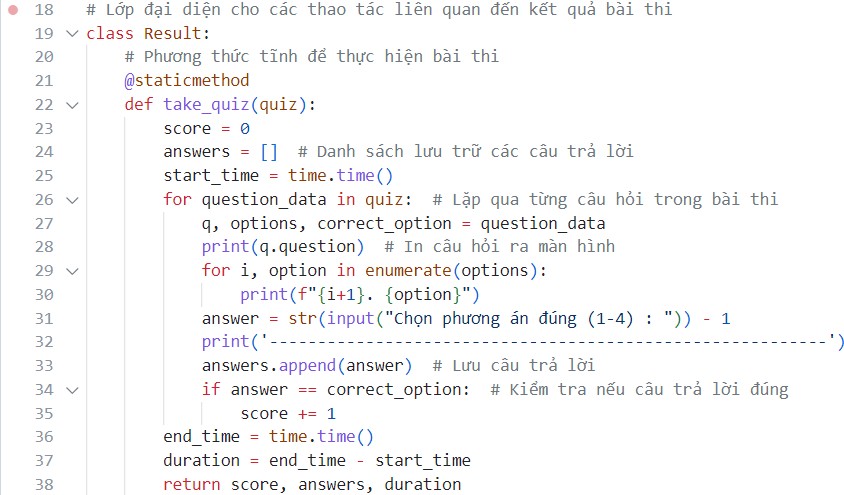
*Ví dụ*: Các lớp User, Question và Result là các ví dụ của lập trình hướng đối tượng.



*Lớp user*



*Lớp question*



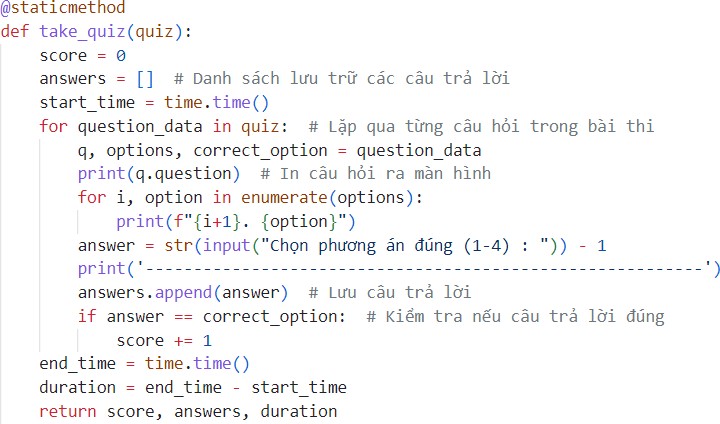
*Ví dụ về các lớp – lập trình hướng đối tượng*

*+ Lập trình hướng sự kiện:* Lập trình hướng sự kiện tập trung vào việc xử lý các sự kiện, thường được sử dụng trong các ứng dụng GUI. Ứng dụng này có một số yếu tố của lập trình hướng sự kiện, đặc biệt là trong cách xử lý các tương tác của người dùng thông qua các lựa chọn menu.

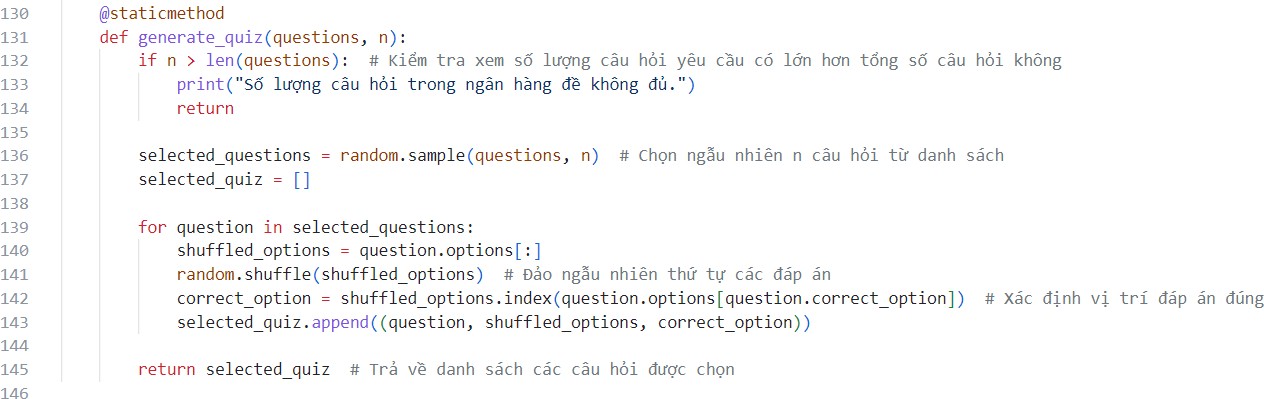
*Ví dụ*: Controller xử lý các lựa chọn của người dùng, một dạng lập trình hướng sự kiện cơ bản.



### Các giải thuật đã được áp dụng.

Thuật toán làm bài thi: Em đã thiết kế ra giải thuật này để tạo ra hàm làm bài thi của học viên. Ở đây có dùng thuật toán duyệt toàn bộ để lặp qua từng câu hỏi trong bài thi.

Giải thuật đảo đáp án: Giải thuật này được em thiết kế ra trên nền tảng giải thuật ngẫu nhiên và, việc chọn ngẫu nhiên các câu hỏi từ danh sách câu hỏi sử dụng hàm random.sample() đảm bảo rằng không có câu hỏi nào bị trùng lặp trong bài thi được tạo ra. Hàm random.sample() sẽ chọn ngẫu nhiên n mục từ danh sách questions mà không có sự lặp lại, đảm bảo rằng mỗi câu hỏi chỉ được chọn một lần.

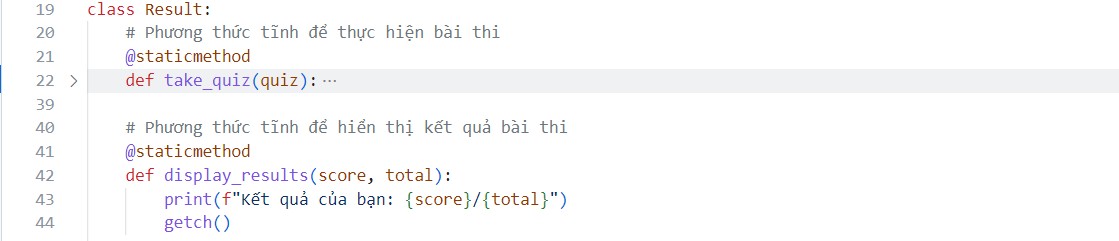


### Các kĩ thuật thiết kế chương trình.

* 1. *Các nguyên tắc thiết kế cơ bản đã được áp dụng*

*Nguyên tắc đơn giản :* Nguyên tắc đơn giản tập trung vào việc giữ cho mã nguồn đơn giản và dễ hiểu. Tránh sự phức tạp không cần thiết và đảm bảo rằng mỗi phần của mã thực hiện một nhiệm vụ rõ ràng.

Trong ứng dụng trắc nghiệm, phương thức display\_results được thiết kế đơn giản:



=> Phương thức display\_results có nhiệm vụ duy nhất là hiển thị kết quả và đợi người dùng nhấn phím.

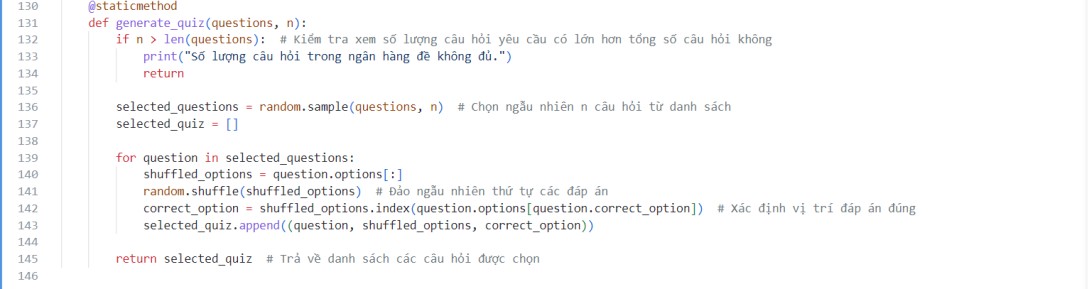
*Nguyên tắc trực tiếp :* Nguyên tắc trực tiếp nhấn mạnh việc viết mã nguồn rõ ràng và trực tiếp, tránh các bước thừa hoặc vòng vèo không cần thiết. Mã nguồn nên thể hiện rõ ràng ý định của lập trình viên.

*Ví dụ*: Trong phương thức register\_user của lớp User, việc kiểm tra tên đăng nhập và thêm người dùng mới được thực hiện trực tiếp:



*Nguyên tắc rõ ràng :* Nguyên tắc rõ ràng đảm bảo rằng mã nguồn dễ đọc và dễ hiểu, không chỉ cho người viết mà còn cho người khác đọc và bảo trì. Sử dụng các tên biến và hàm rõ ràng, và viết các comment khi cần thiết.

*Ví dụ*: Phương thức generate\_quiz của lớp Question được viết rõ ràng để mô tả cách thức tạo ra một bài kiểm tra ngẫu nhiên:



*Nguyên tắc cấu trúc tốt :* Nguyên tắc cấu trúc tốt nhấn mạnh việc tổ chức mã nguồn theo các khối logic, giúp mã nguồn dễ hiểu và dễ bảo trì. Sử dụng các lớp, phương thức và module để chia nhỏ và tổ chức mã nguồn.

*Ví dụ:* Việc tổ chức mã nguồn theo mô hình MVC là một ví dụ về cấu trúc tốt:

+ Model: Chứa các lớp User, Question, Result.

+ View: Chứa lớp View để hiển thị giao diện người dùng.

+ Controller: Chứa lớp Controller để điều phối các hoạt động giữa Model và View.

* 1. *Các thiết kế giải thuật đã được áp dụng*
     1. *Cơ chế top-down và modularity*

Cơ chế top-down :Cơ chế top-down (thiết kế từ trên xuống) bắt đầu từ cấp độ cao nhất của hệ thống và dần dần phân rã thành các phần nhỏ hơn. Đây là phương pháp thiết kế mà hệ thống được phân chia thành các mô-đun con để dễ quản lý và phát triển.

Modularity: Modularity (tính mô-đun) liên quan đến việc chia chương trình thành các mô-đun riêng biệt, mỗi mô-đun có chức năng riêng và có thể phát triển, kiểm tra độc lập. Các mô-đun này thường được tổ chức thành các lớp hoặc các file riêng biệt.

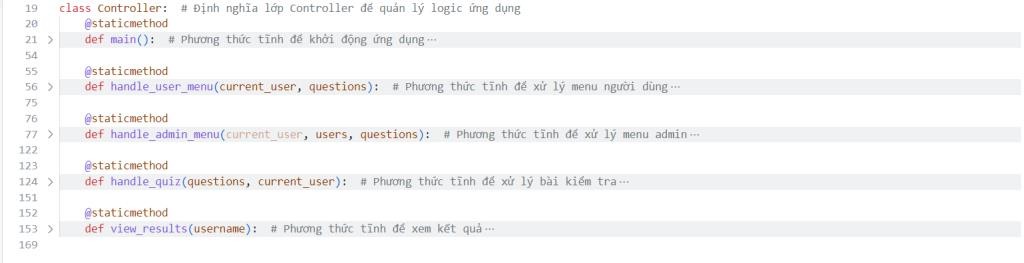
*Ví dụ :*

+ Thiết kế Top-down:

Bắt đầu từ yêu cầu tổng thể, chúng ta phân chia ứng dụng thành các chức năng chính như quản lý người dùng, quản lý câu hỏi, và xử lý kết quả bài kiểm tra.

+ Modularity:

Chúng ta chia ứng dụng thành các mô-đun tương ứng với các lớp User, Question, Result, View, và Controller.



* + 1. *Kỹ thuật xác định hàm, thủ tục, tham số*

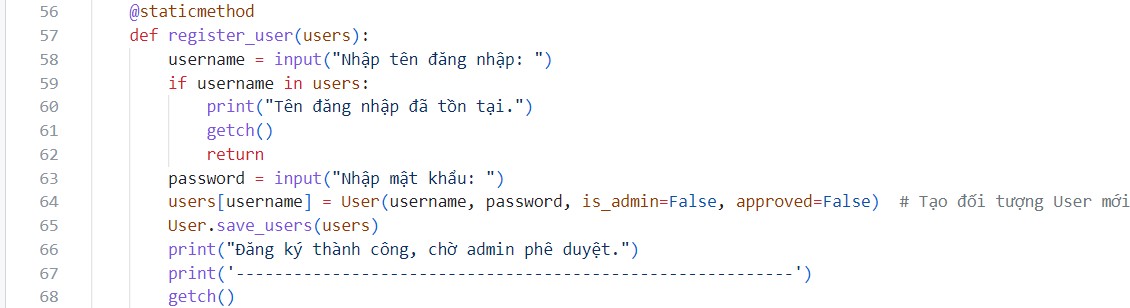
Xác định hàm và thủ tục : Xác định hàm và thủ tục liên quan đến việc phân chia chức năng của chương trình thành các đơn vị nhỏ hơn (hàm hoặc thủ tục), mỗi đơn vị thực hiện một nhiệm vụ cụ thể.

Xác định tham số : Xác định tham số liên quan đến việc xác định các đầu vào cần thiết cho mỗi hàm hoặc thủ tục để thực hiện nhiệm vụ của nó. Tham số giúp tăng tính tái sử dụng và linh hoạt của hàm.

* Ví dụ trong ứng dụng trắc nghiệm: Hàm register\_user trong lớp User:

+ Hàm thực hiện chức năng đăng ký người dùng mới.

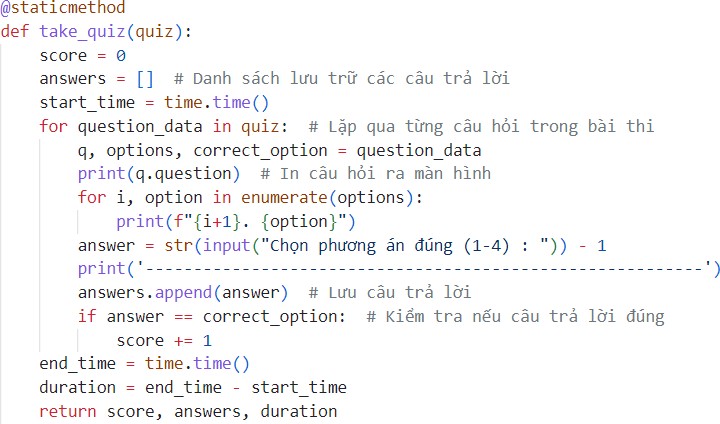
+ Tham số users là danh sách người dùng hiện tại.



Hàm take\_quiz trong lớp Result:

+ Hàm thực hiện chức năng làm bài kiểm tra.

+ Tham số quiz là danh sách các câu hỏi trong bài kiểm tra.

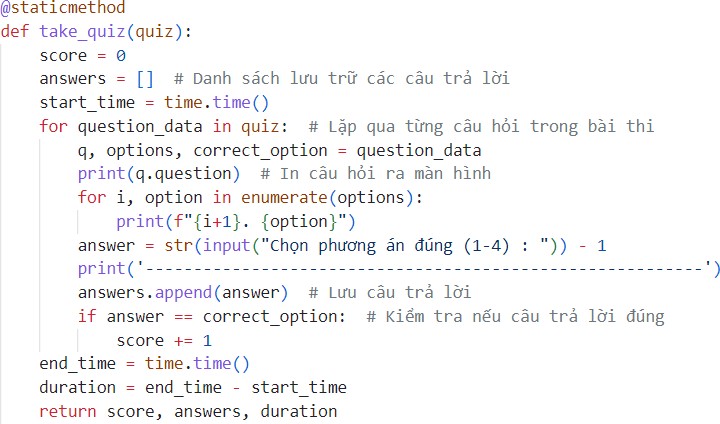


* + 1. *Sử dụng biến*

Có 2 loại biến là biến cục bộ và biến toàn cục: Biến cục bộ chỉ tồn tại trong phạm vi của một hàm hoặc thủ tục, trong khi biến toàn cục có thể được truy cập từ bất kỳ đâu trong chương trình.

Ví dụ trong ứng dụng của em: Biến cục bộ trong hàm take\_quiz:

+ Score, answers, start\_time, end\_time, và duration là các biến cục bộ chỉ tồn tại trong phạm vi của hàm take\_quiz.

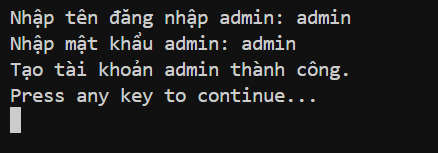


***Tổng quan:*** Việc áp dụng các kỹ thuật thiết kế chương trình như cơ chế top - down, modularity, xác định hàm, thủ tục, tham số và sử dụng biến giúp mã nguồn của ứng dụng trở lên rõ ràng, dễ hiểu, và dễ bảo trì hơn. Các kỹ thuật này cũng giúp tăng tính tái sử dụng và giảm thiểu lỗi trong quá trình phát triển và bảo trì phần mềm.

## Chức năng chương trình

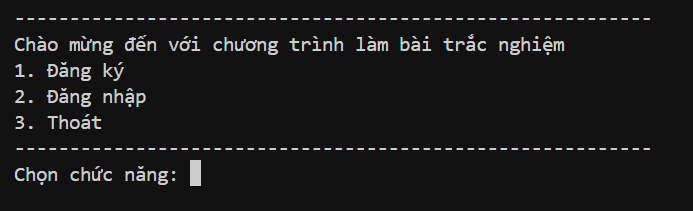
### Chức năng thêm người dùng admin.

Khi mới cài đặt app chúng ta chưa tồn tại người dùng ở cả admin và user lúc này bài toán đặt ra là tạo 1 tài khoản admin mới để thực hiện các chức năng admin, duyệt các thành viên mới.



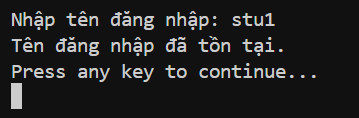
### Chức năng đăng kí đăng nhập (phân quyền admin – user)

Để có phân rõ quyền người giao đề và sinh viên/học sinh làm bài cũng như cho nhiều sinh viên có thể tham gia thi ta cần có chức năng đăng kí, đăng nhập. Đồng thời thêm tiện ích thoát chương trình nếu không sử dụng nữa.



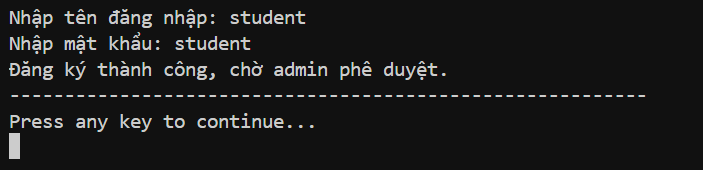
Chức năng đăng kí : Ở giao diện Authentication sau khi ta chọn 1 sẽ đưa vào màn hình giao diện đăng ký

TH1: Nếu người dùng đã tồn tại thì thông báo ra màn hình và quay lại trang đăng kí, đăng nhập.



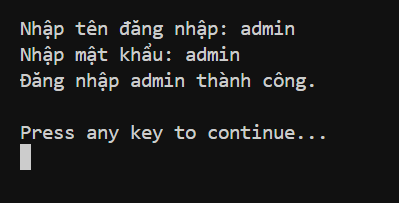
*Thông báo tài khoản đã tồn tại*

TH2: Nếu người dùng chưa tồn tại, gửi thông tin của sinh viên về cho quản trị viên xét duyệt.

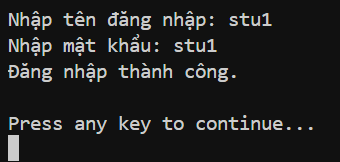


*Đăng ký tài khoản thành công, chờ admin duyệt*

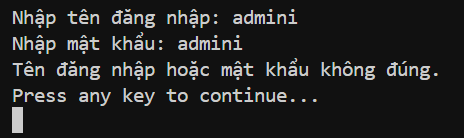
Chức năng đăng nhập: Khi bạn đăng nhập và hệ thống nhận ra bạn là admin hoặc user. Thông báo ra màn hình quyền của bạn và hiện giao diện của admin/user



*Thông báo khi đăng nhập admin thành công .*



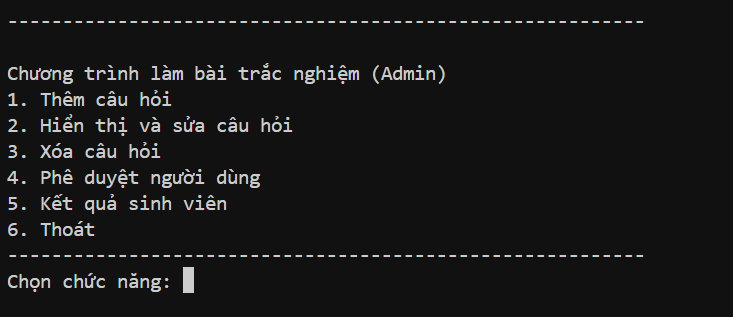
*Thông báo khi nhập user thành công.*



### Chức năng của admin

*Đăng nhập thất bại*

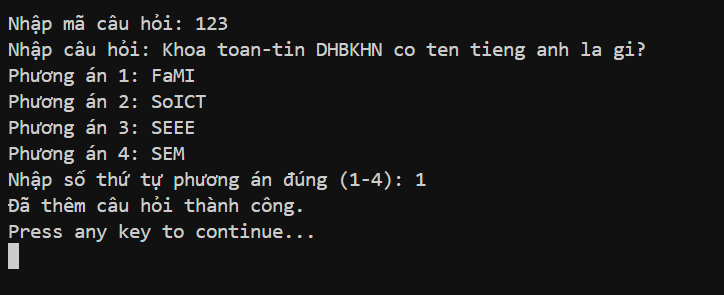
Sau khi đăng nhập admin thành công chúng ta sẽ nhận được giao diện như sau:



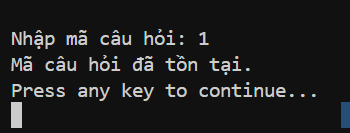
*Màn hình dashbroad của admin*

Từ đây ta có thể thấy được admin có 1 số quyền cơ bản như: Thêm câu hỏi, hiển thị và sửa câu hỏi, xóa câu hỏi, phê duyệt người dùng mới, xem kết quả sinh viên và có chức năng đăng xuất.

+ *Chức năng thêm câu hỏi*: admin sẽ thêm mã câu hỏi, câu hỏi, các câu trả lời ( 4 câu trả lời) và đánh dấu câu trả lời đúng.

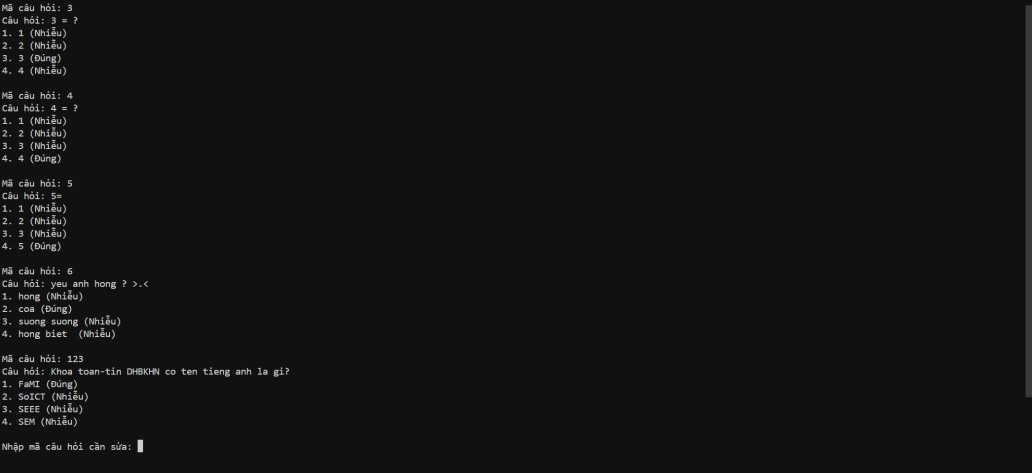


*Ví dụ về thêm 1 câu hỏi*

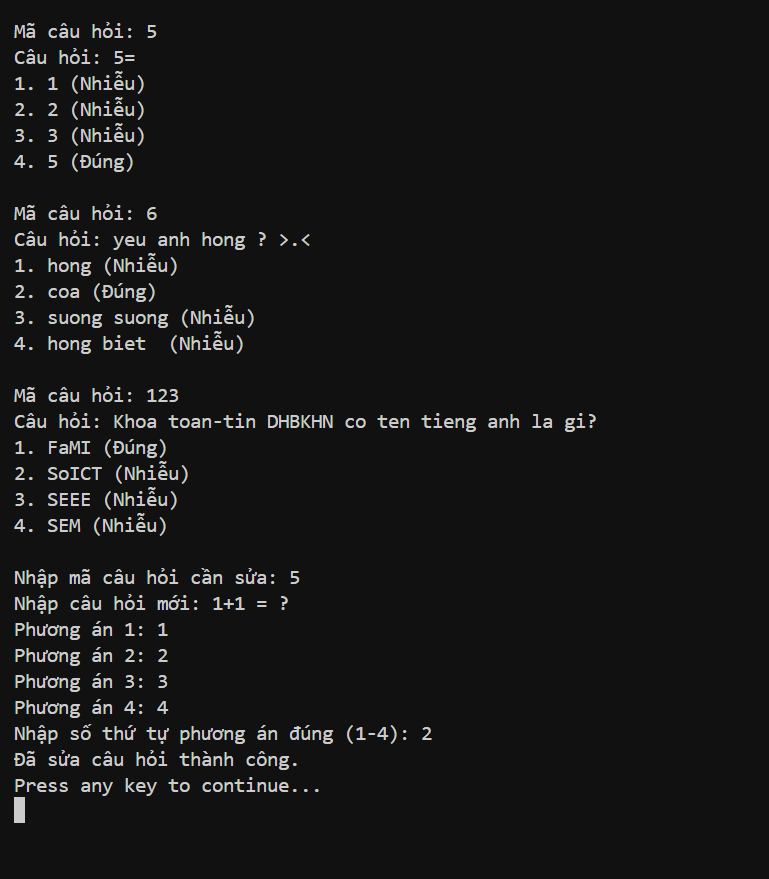


*Nếu mã câu hỏi đã tồn tại -> thông báo cho người dùng*

+ *Chức năng sửa và hiển thị câu hỏi*: sau khi chọn nó sẽ hiện ra toàn bộ các câu hỏi đang có trong ngân hàng đề lúc này ta chỉ cần chọn và sửa câu hỏi theo ý muốn.

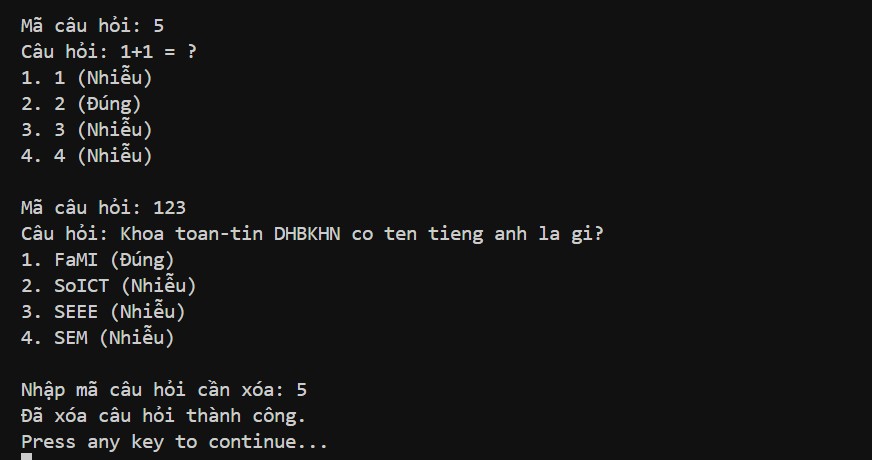


*Ví dụ về hiển thị câu hỏi*



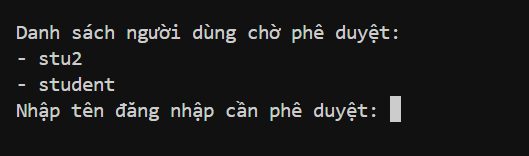
*Ví dụ về việc sửa câu hỏi*

+ *Chức năng xóa câu hỏi*: Khi chọn chức năng này chương trình cũng sẽ hiện toàn bộ câu hỏi trong ngân hàng đề ra để thuận tiện cho việc chọn câu hỏi để xóa. Người dùng sẽ nhập mã đề câu hỏi cần xóa vào để xóa.



*Ví dụ về chức năng xóa câu hỏi.*

+ *Chức năng phê duyệt người dùng*: Chương trình sẽ hiện ra 1 danh sách các user đã đăng kí và đang chờ xét duyệt. Lúc này admin có thể thêm người dùng.



*Duyệt người dùng mới.*

+ *Chức năng xem kết quả sinh viên*: Admin sẽ có thể xem kết quả thi của học viên theo ý muốn. Nếu chưa ai thi => hiện ra thông báo. Không tồn tại bài thi, ngược lại hiển thị ra tất cả các bài thi mà sinh viên đã làm và các dữ liệu liên quan.



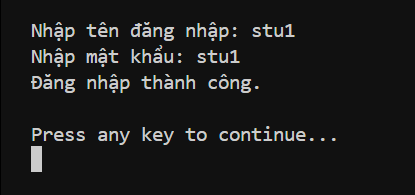
*Nhập id sinh viên muốn xem kết quả*



*Ví dụ về bài thi của 1 sinh viên*

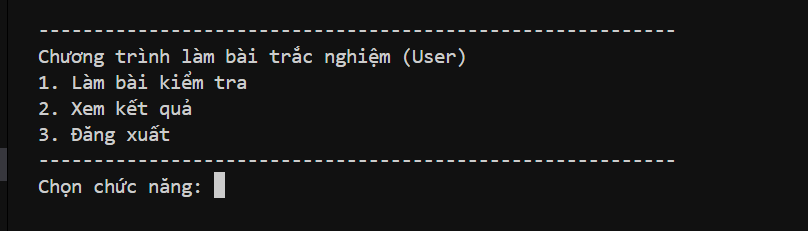
### Chức năng của user

Sau khi đăng nhập thành công với tài khoản học viên ta nhận được màn hình hiển thị thông báo đã đăng nhập với quyền user như sau:



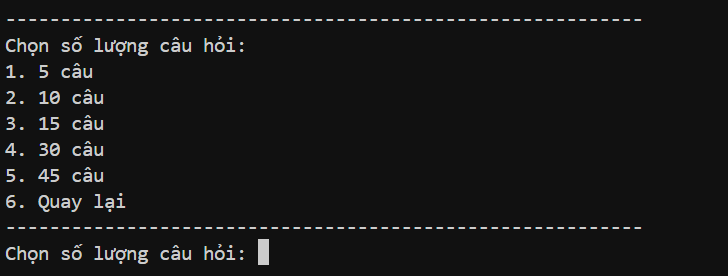
*Đăng nhập user*

Từ đây ta có thể nhìn thấy các chức năng của phần mềm dưới quyền của 1 user như: Làm bài thi, xem kết quả của bản thân mình và đăng xuất.

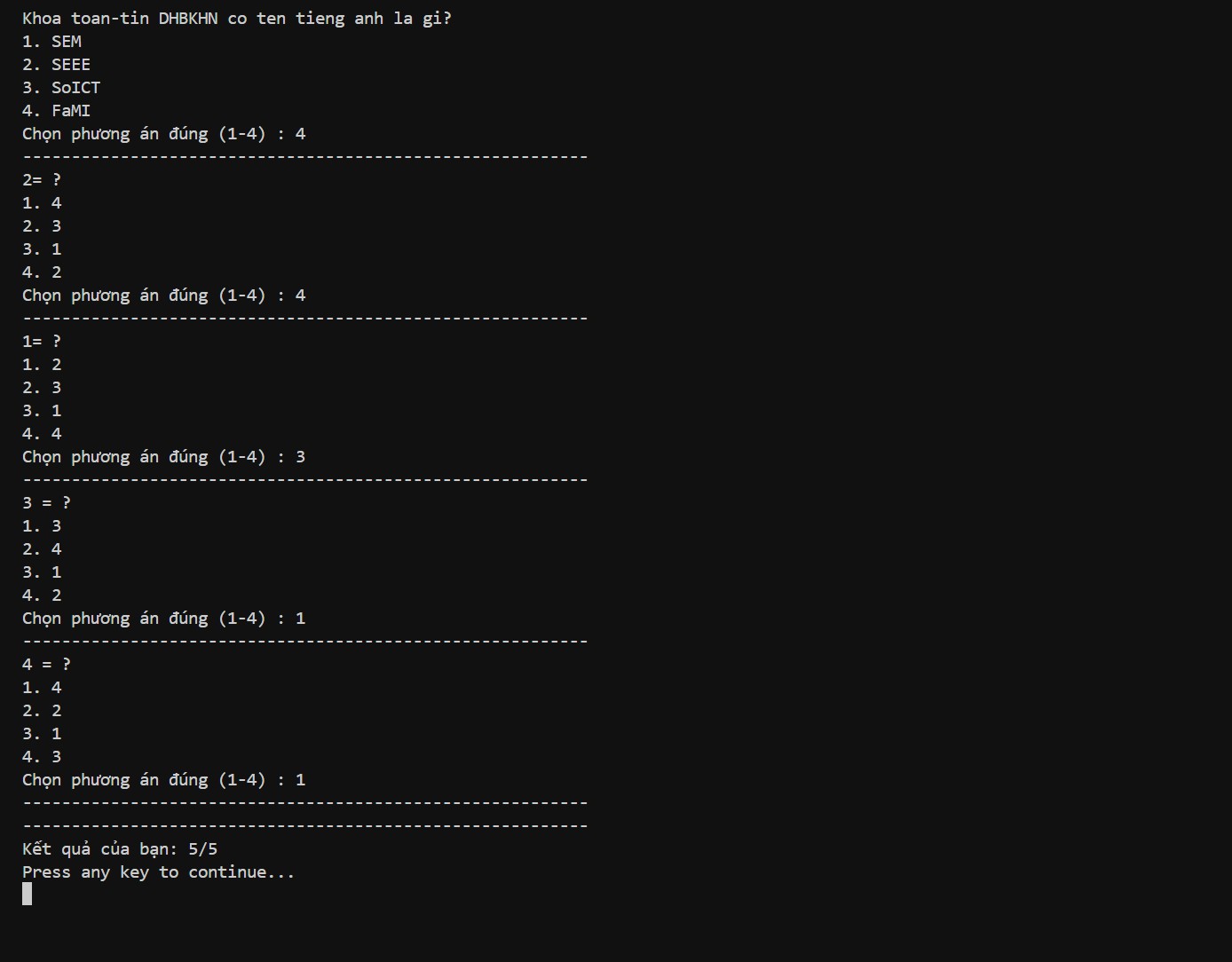


*Giao diện người dùng*

+ *Chức năng làm bài thi:* Ở chức năng này, sinh viên sẽ chọn bài thi tương ứng số câu hỏi. Hệ thống sẽ lấy câu hỏi từ ngân hàng đề để sinh ra bài thi đồng thời gán mã đề sinh viên đã làm + tính toán thời gian sinh viên làm bài.

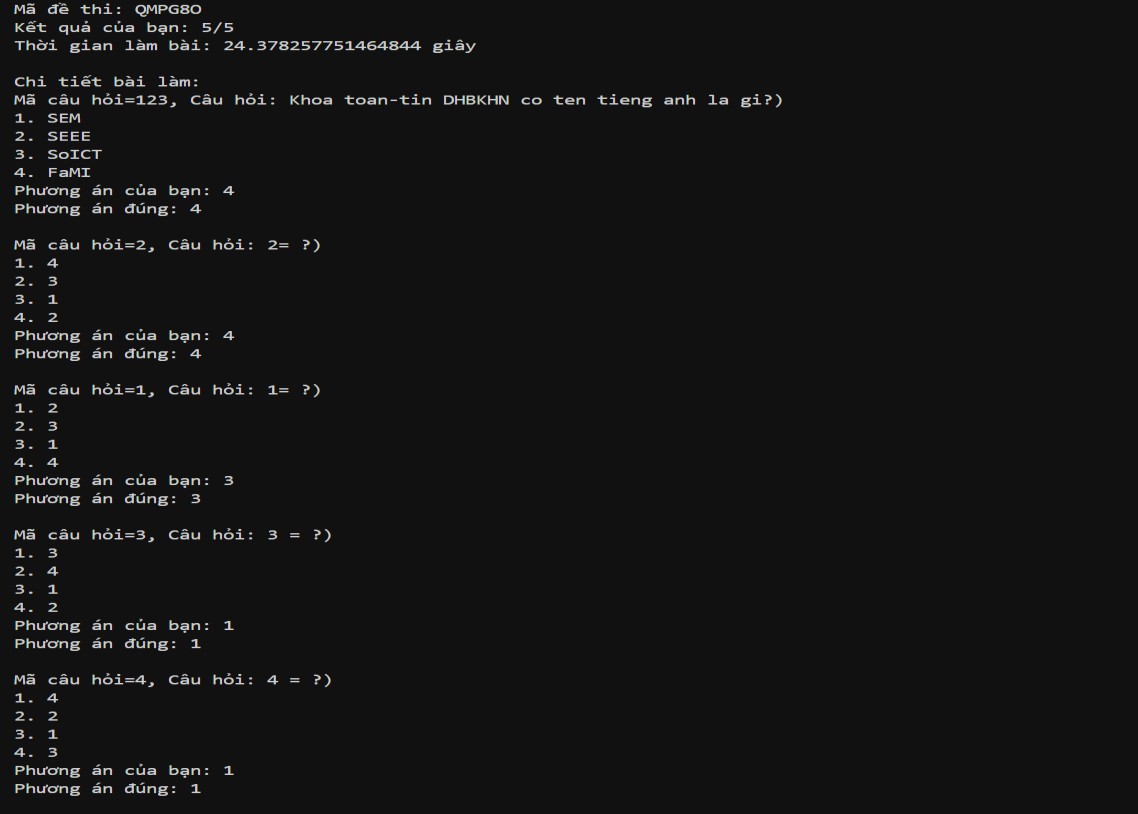


*Chọn số lượng câu hỏi cho bài thi*



*Làm bài thi*

+ *Chức năng xem kết quả cá nhân :* Sinh viên có thể tự xem lại bài kiểm tra mình đã làm và các dữ liệu liên quan



*Hiển thị ra tất cả các bài kiểm tra mà sinh viên đã làm*

## Kiểm thử chương trình và kết quả chạy.

### Kiểm thử, gỡ lỗi.

Các kĩ thuật đã sử dụng để gỡ lỗi:

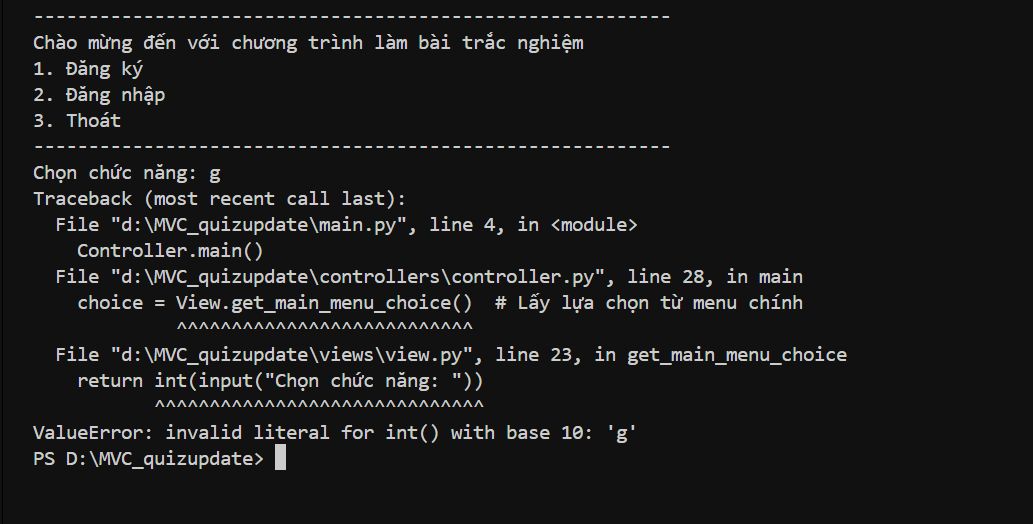
+ Trong quá trình lập trình em đã sử dụng sẵn trình phát hiện lỗi(debuger) của IDE Visual Studio Code và gỡ lỗi (debug) của vscode nhằm tìm ra các lỗi trong chương trình. Nó trả về kết quả biết là em đang bị lỗi ở đâu. Từ đó em có thể xác định được rằng mình cần làm gì tiếp theo.

+ Sử dụng các comment để biết được logic bài code mình đang làm cũng như để thuận tiện cho việc sửa chữa và nâng cấp sau này.

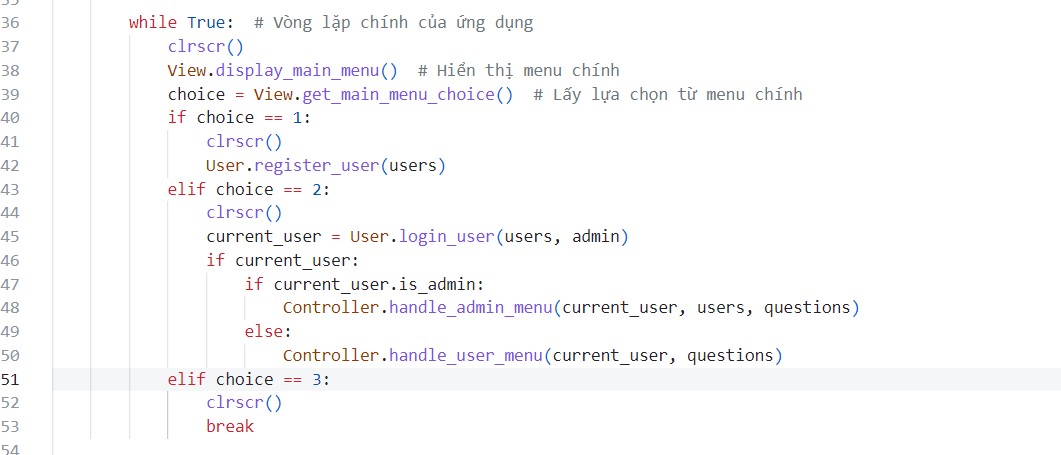
Các kĩ thuật đã được sử dụng để kiểm thử:

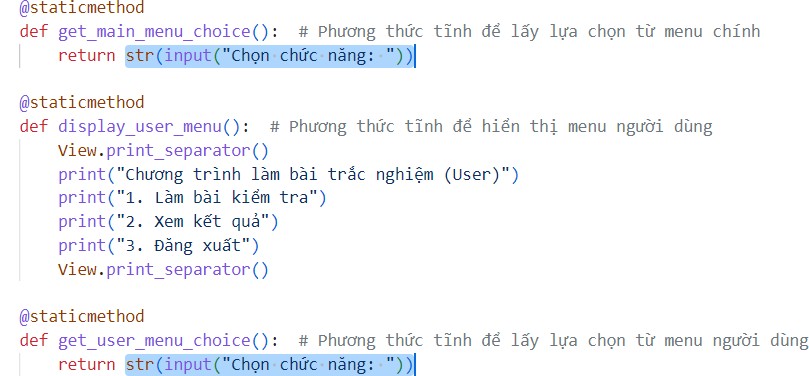
+ ***Kiểm thử chức năng (Sử dụng phương pháp kiểm thử hộp xám)***: Em đã kiểm tra từng chức năng của phần mềm bằng cách chạy và vận hành nó. Với hiểu biết của mình về mã nguồn em đã chạy toàn bộ chức năng của chương trình. Từ đó em đã xác định phần mềm của em có 1 lỗi khi nhập giá trị đầu vào của phần mềm như sau:

Trong khi kiểm thử và vận hành chương trình có 1 lỗi đã xảy ra khi người dùng truyền giá trị đầu vào không đúng ở các ô chọn chức năng:

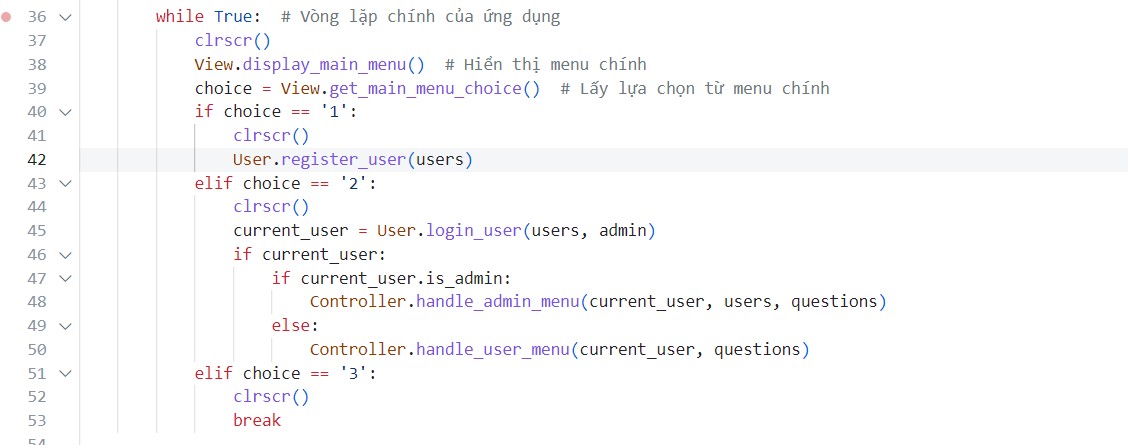


Xác định nguyên nhân : Do em đã khởi tạo gán giá trị đầu vào là int để thực hiện hàm so sánh .

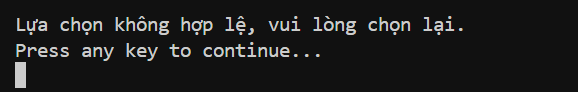


Sửa lỗi: Lỗi này có thể xử lý bằng cách sửa choice = cho kiểu dữ liệu đầu vào là 1 kí tự (char) hoặc chuỗi(string). Và thay đổi cách so sánh.

*Sửa lại kiểu dữ liệu của giá trị truyền vào*



*Sửa lại giá trị so sánh từ 1 số sang thành 1 ký tự*

Thêm tính năng đưa ra thông báo nếu giá trị đầu vào không hợp lệ:

+ ***Kiểm thử hồi quy:*** Sau khi một lỗi được phát hiện và sửa chữa, phần mềm được kiểm thử lại để xác nhận lỗi ban đầu đã được khắc phục.

Điều này giúp đảm bảo rằng không có lỗi mới xuất hiện và tính ổn định của phần mềm được bảo đảm.

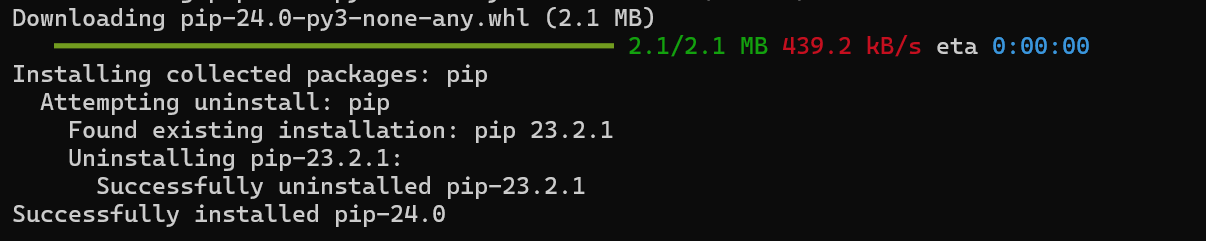
+ Ngoài ra em còn **kiểm thử tương thích** bằng cách chương trình bằng cách mang mã nguồn hoặc phần mềm bản release chuyển qua máy của người thân/ bạn bè để chạy thử. Kết quả là chương trình hoàn toàn chạy được với các máy đã cài python 3.12.1.

### Chạy và phát hành chương trình.

Python có 1 công cụ tuyệt vời để phát hành ra bản cuối cùng của chương trình đó là pyinstaller:

Đầu tiên chúng ta sẽ cài đặt pyinstaller bằng cách vào CMD gõ lệnh:

Chương máy tính sẽ tự động cài phần mềm và cài đặt nó

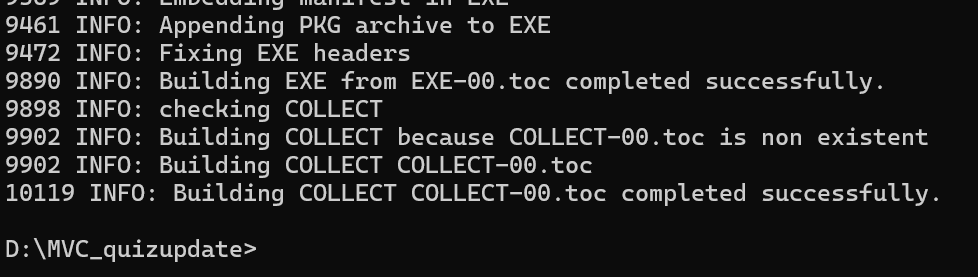


=>Ở đây do máy em đã cài sẵn rồi nên nó có cài đặt thêm bản mới nhất.

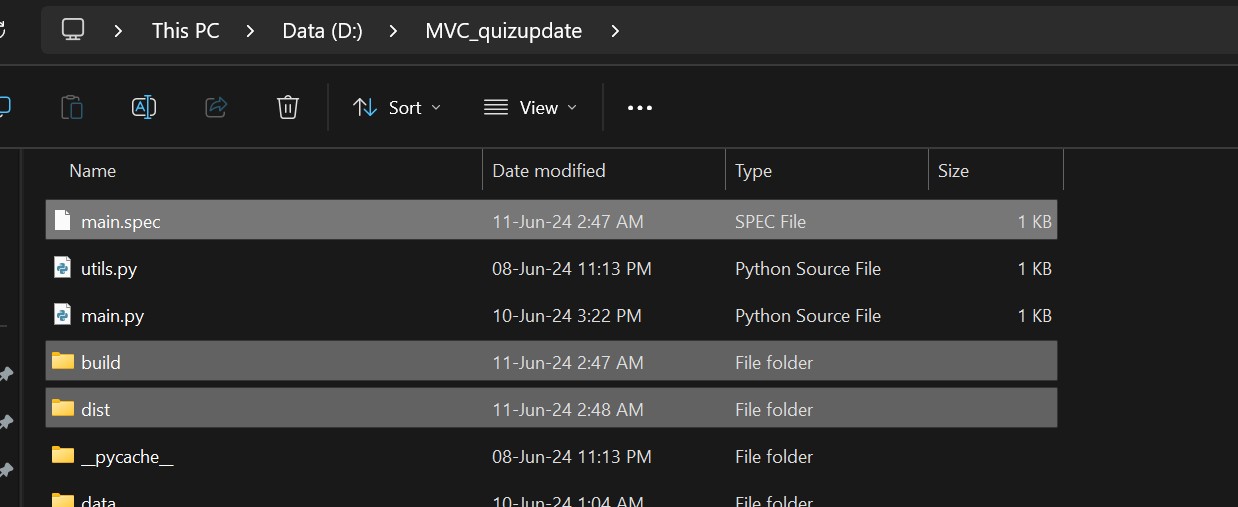
Sau khi cài đặt PyInstaller, em sử dụng nó để tạo file thực thi từ mã nguồn Python của em. Để làm điều này, di chuyển đến thư mục chứa mã nguồn python trong terminal hoặc command prompt và chạy lệnh sau:



Sau khi chương trình chạy thành công sẽ có thông báo dạng như sau



Kết quả sau khi thực hiện chúng ta có thêm 3 thư mục mới dược sinh ra trong thư mục chứa mã nguồn:



Chương trình cuối cùng được chạy ở trong thư mục dist (sau khi mở lên thì data cũng được sinh và lưu trữ trong thư mục dist/data):



Từ file nguồn thực thi như này ta chỉ cần gửi 3 thư mục/file mới được tạo gồm dist, build và main.spec cho người khác là họ hoàn toàn có thể chạy được trên máy của họ.

Ngoài ra ta có thể mã hóa mã nguồn và sau đó viết thêm 1 chương trình để thực hiện cài đặt trên máy họ.

Khi cần cập nhật ta chỉ cần mở cmd và thực hiện ghi đè lên chính thư mục mà ta đã tạo trước đó là được.

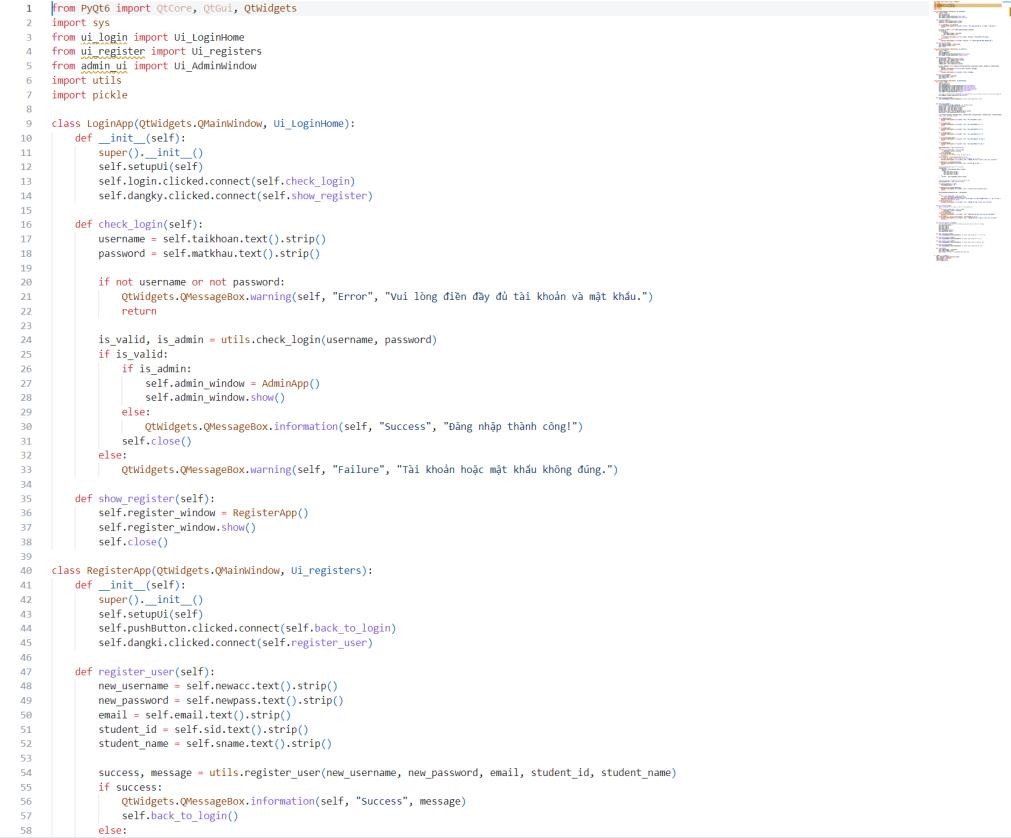
## Đề xuất nâng cấp.

### Xây thêm giao diện bằng framework Qt (tìm hiểu thêm)

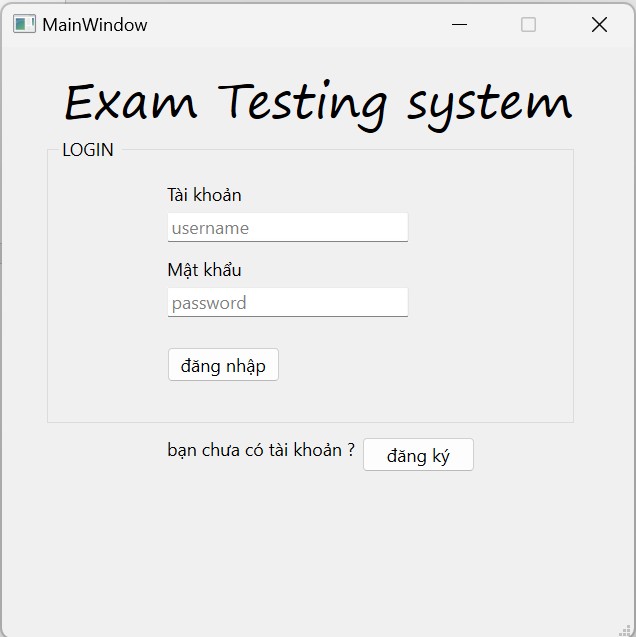
Sau khi hoàn thành chương trình em có tìm hiểu thêm 1 số framework để nâng cấp chương trình ví dụ ở đây em có sử dụng framework PyQt6 để

làm giao diện cho chương trình. Phần này em mở rộng thêm nên mã nguồn của giao diện em sẽ không đưa lên báo cáo tránh việc khi thầy chạy thử thì cần phải cài thêm 1 số gói dữ liệu rắc rối và giảm tải dung lượng chương trình khi phát hành.

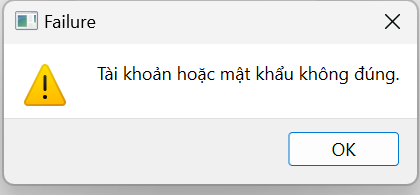
Ví dụ 1 số hình ảnh cơ bản về việc nâng cấp giao diện cho chương trình:



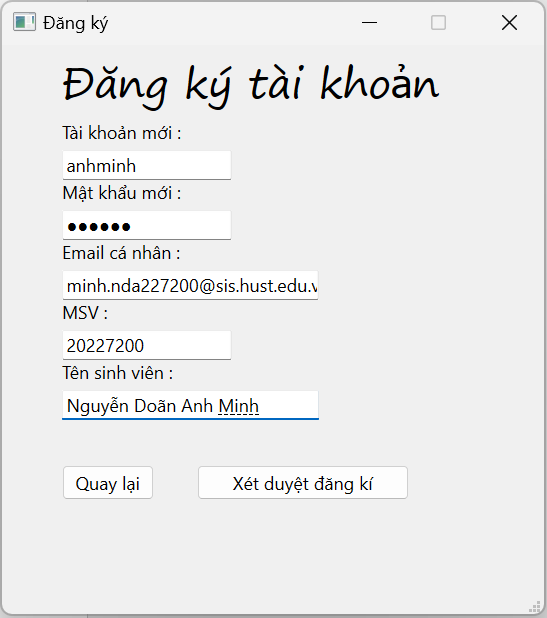
*Tổng quan code của chương trình có giao diện*



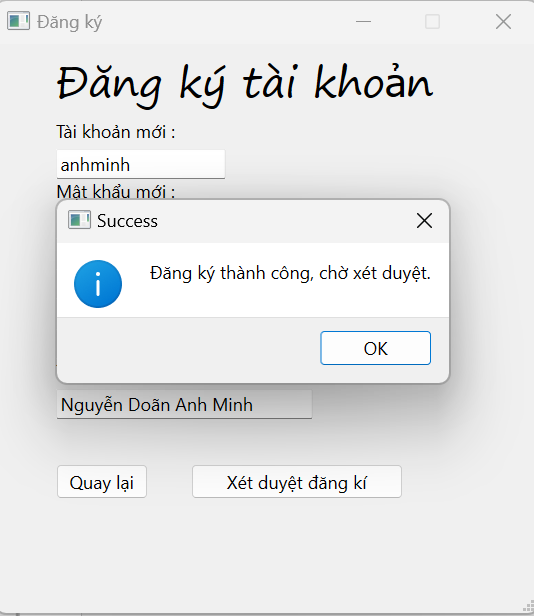
*Giao diện đăng nhập*



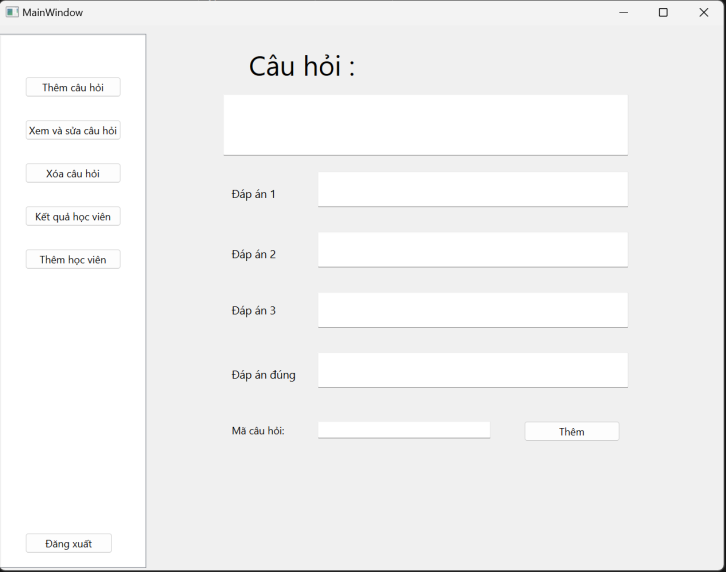
*Thông báo khi đăng nhập thất bại*



*Giao diện đăng kí*



*Đăng kí thành công*



*Giao diện admin*

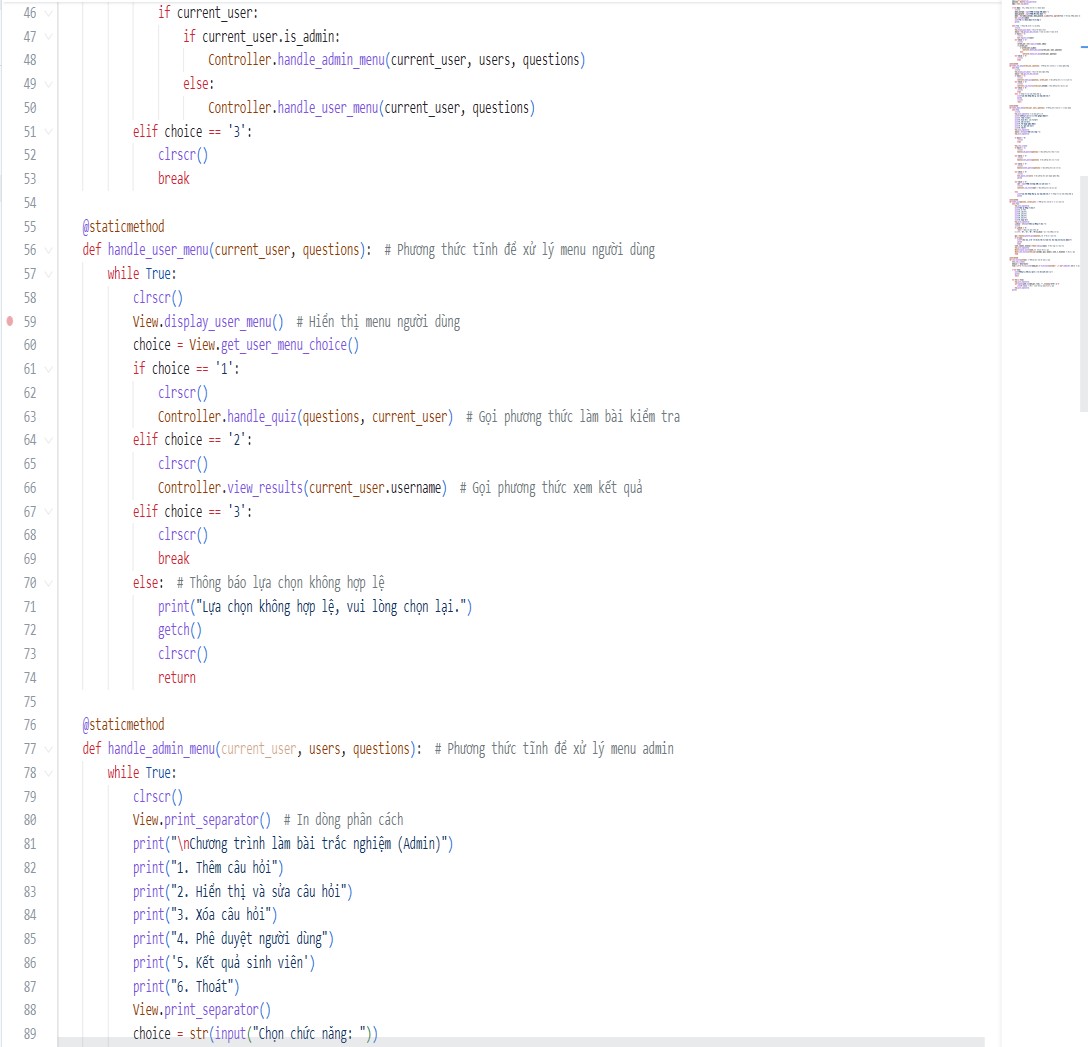
Ngoài ra còn các chức năng của admin và giao diện người dùng…

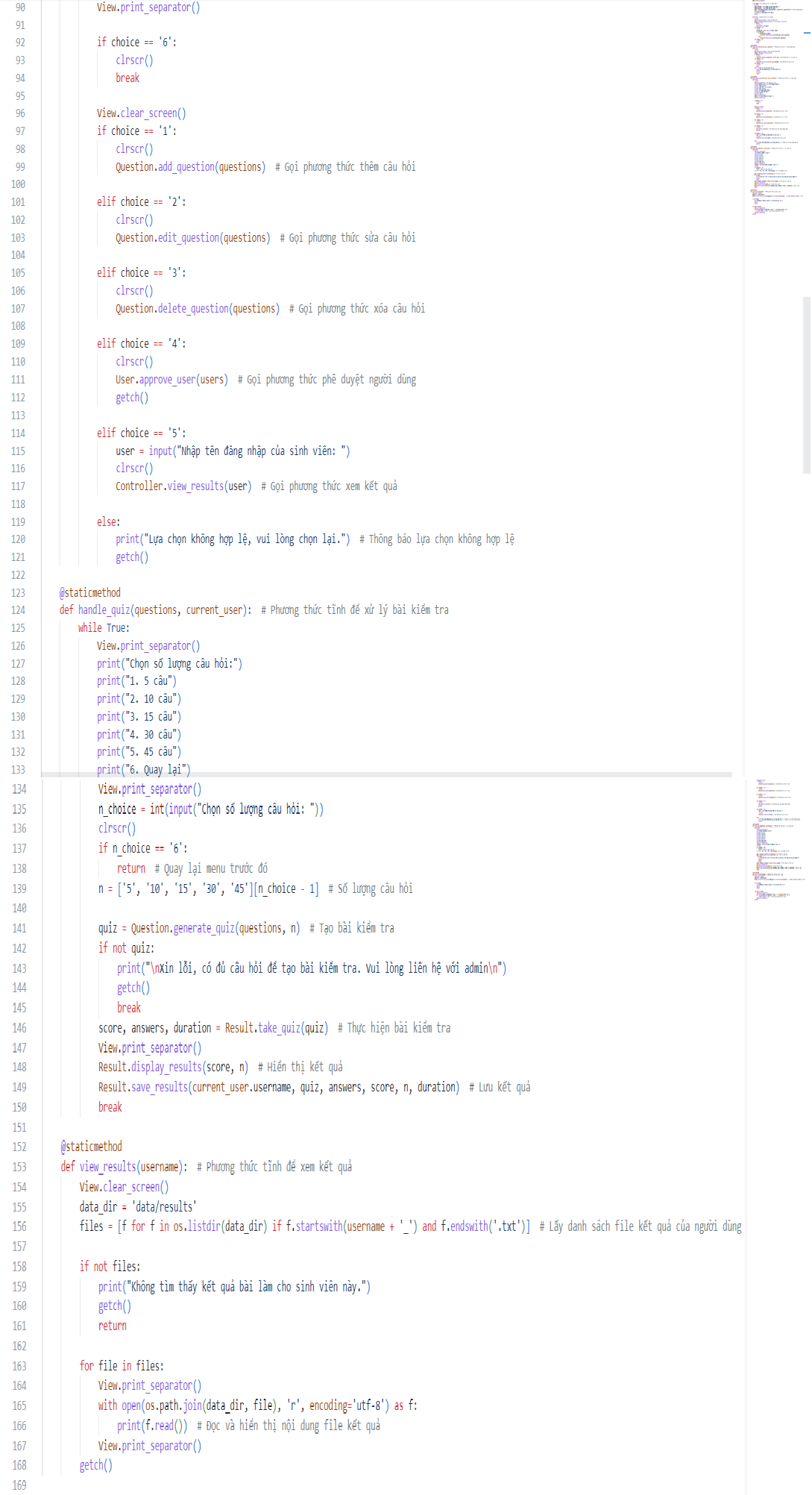
 Việc nâng cấp giao diện UX/UI cho chương trình làm cho chương trình trở lên dễ sử dụng hơn và cũng thẩm mỹ hơn.

## Mã nguồn.

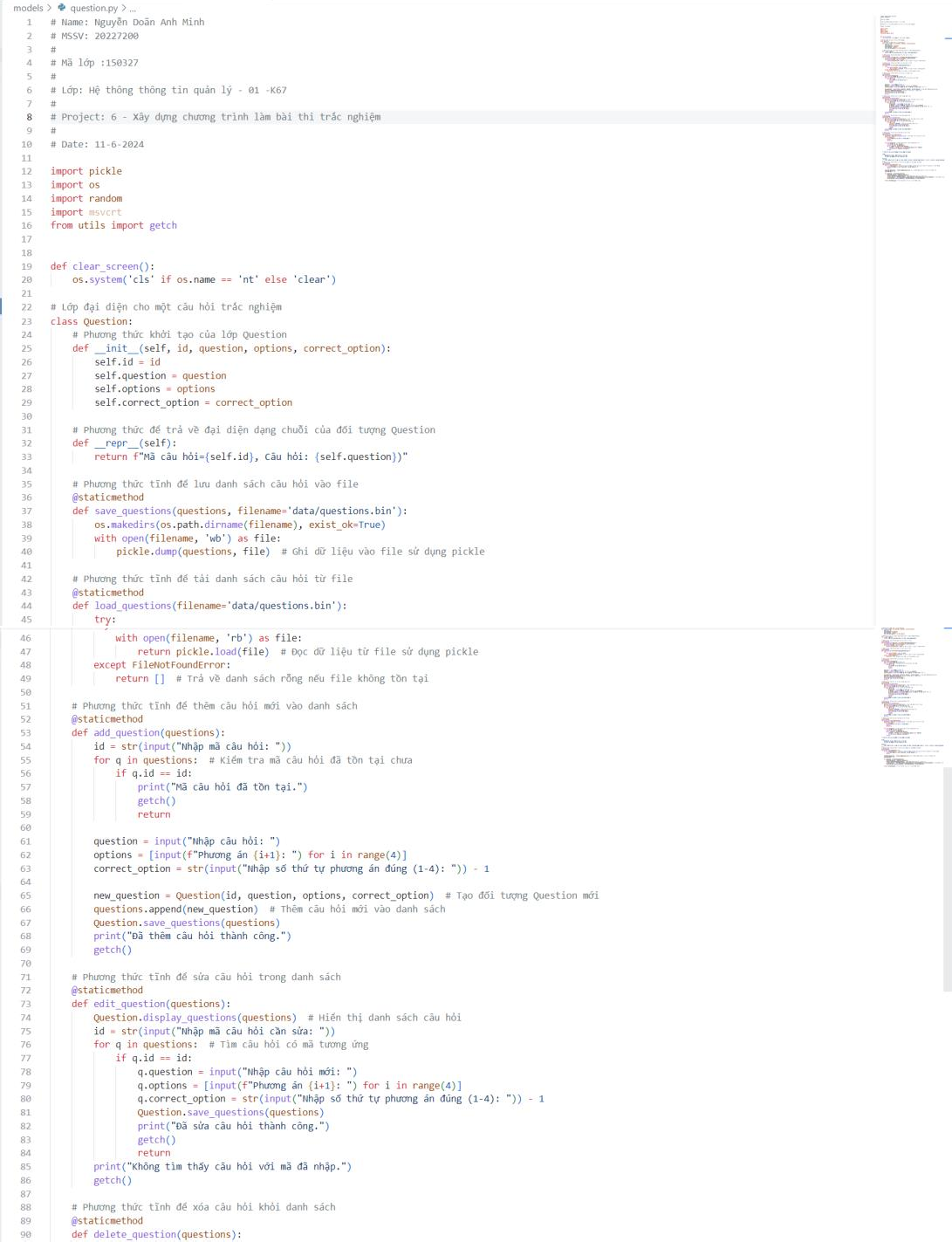
Do em đã chia cấu trúc chương trình thành các file nhỏ độc lập có những chức năng độc lập nên em xin trình bày mục này thành các file lần lượt như sau :

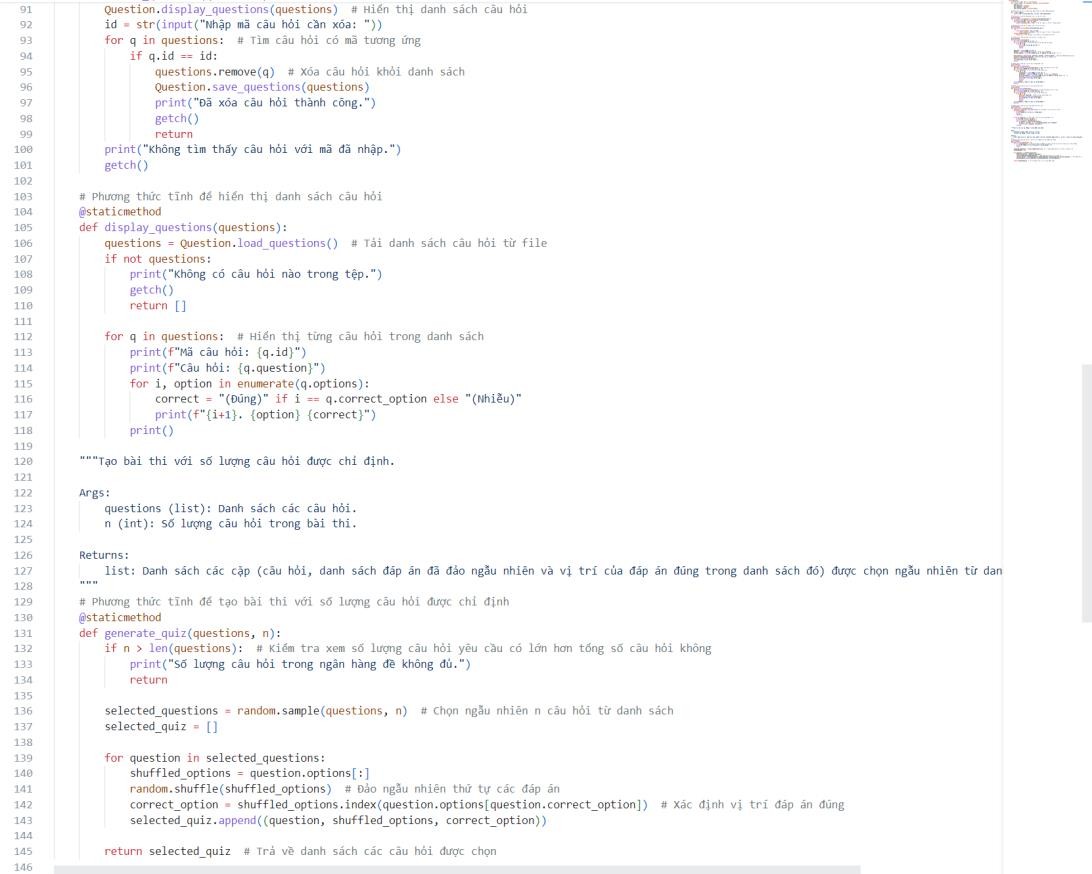
***File: controller.py***



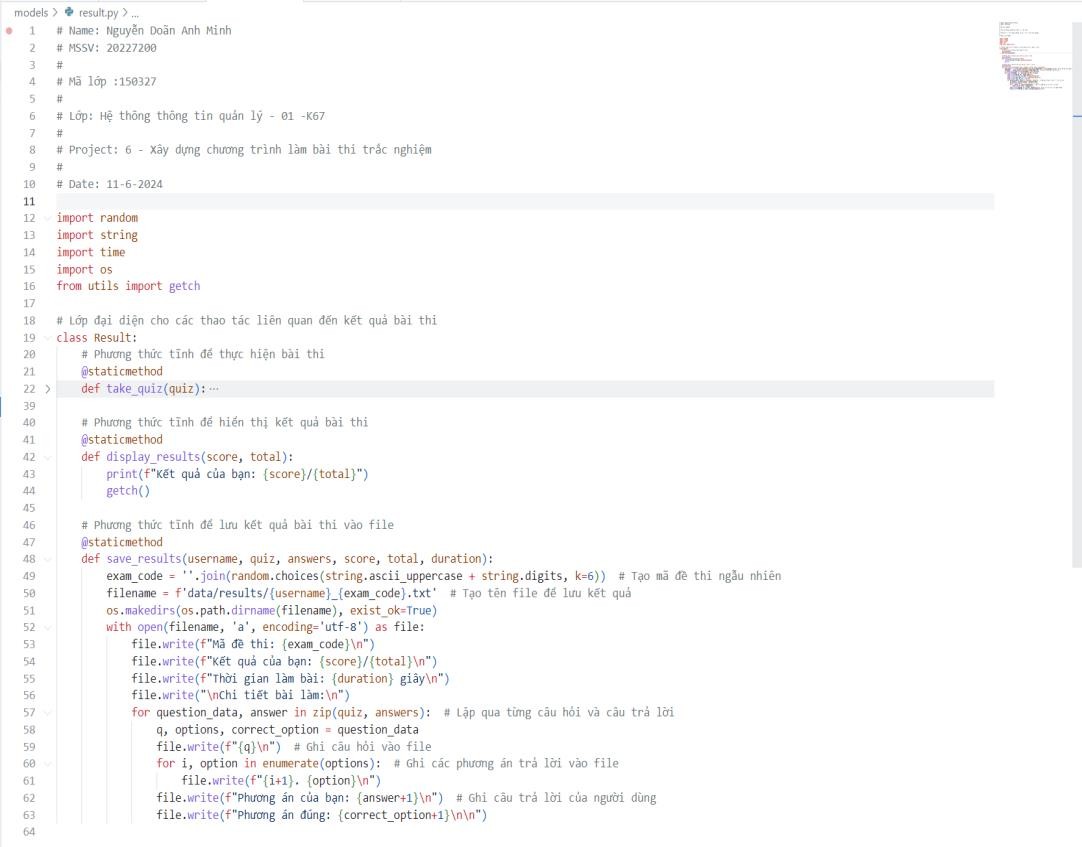


***File: question.py***

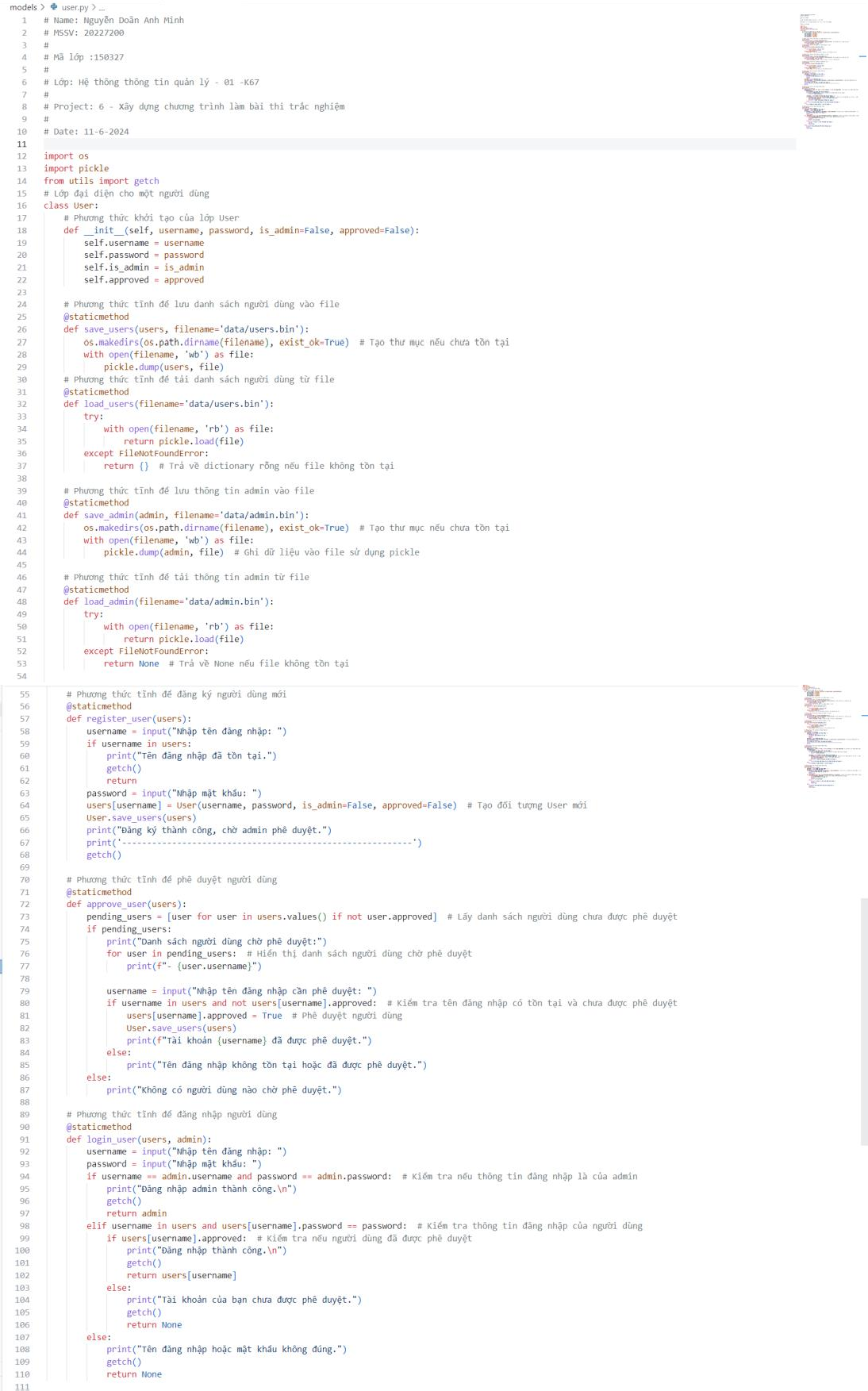




***File: result.py***



***File: user.py***



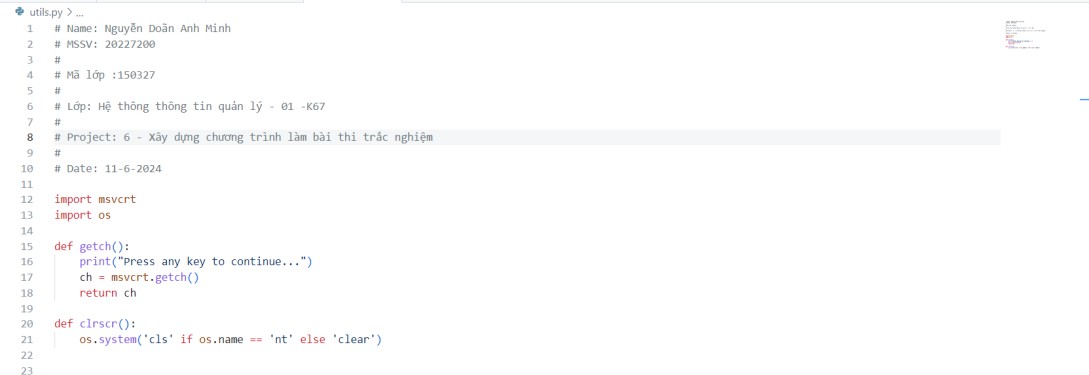
***File: view.py***



***File: main.py***



***File: utils.py***



## Tổng kết

Những gì đã học được trong học phần này:

+ Phong cách lập trình tốt.

+ Kĩ năng viết chương trình máy tính.

+ Các phương pháp lập trình hiệu quả.

+ Kỹ thuật tổ chức chương trình để quản lý 1 cách hiệu quả

+ Cách xử lý, thao tác dữ liệu trên các file .

+ Cách thiết kế 1 chương trình từ đầu..

Những gì đã làm được:

+ Xây dựng được 1 chương trình hoàn chỉnh .

+ Sửa lỗi và phát triển chương trình theo nhu cầu.

+ Khả năng giải quyết vấn đề thực tế gặp phải khi lập trình.

+ Thiết kế 1 chương trình có đầy đủ các chức năng.

Tóm lại, bài tập lớn này đã là cơ hội vời giúp em tiếp cận và hiểu thêm những kiến thức mới, từ đó phát triển và củng cố kỹ năng thực hành quan trọng trong lĩnh vực lập trình. Qua trải nghiệm này, em đã xây dựng được một nền tảng vững chắc là cơ sở cho các môn học và công việc lập trình trong tương lai.

## Tài liệu tham khảo

Nguồn tham khảo từ dự án javaFX của youtuber Feel To Code Free:

1. <https://github.com/virendrapatel62/Quiz_application>
2. <https://youtu.be/DSy-PIHxdn0?si=9e5yWaFqUPDf5xxq>

Ngoài ra còn 1 số nguồn khác:

1. <https://stackoverflow.com/>
2. <https://www.w3schools.com/python/>
3. <https://copilot.microsoft.com/>
4. <https://github.com/edu>