

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



XÂY DỰNG ỨNG DỤNG HỖ TRỢ HỌC
PYTHON ÁP DỤNG TKINTER, PYOBDC VÀ
WEBVIEW

Nhóm 3			
Sinh viên thực hiện:			
STT	Họ tên	MSSV	Ngành
1	Nguyễn Cao Thắng	22521329	CNTT
2	Võ Anh Quân	22521192	CNTT
3	Võ Minh Quyền	22521227	CNTT

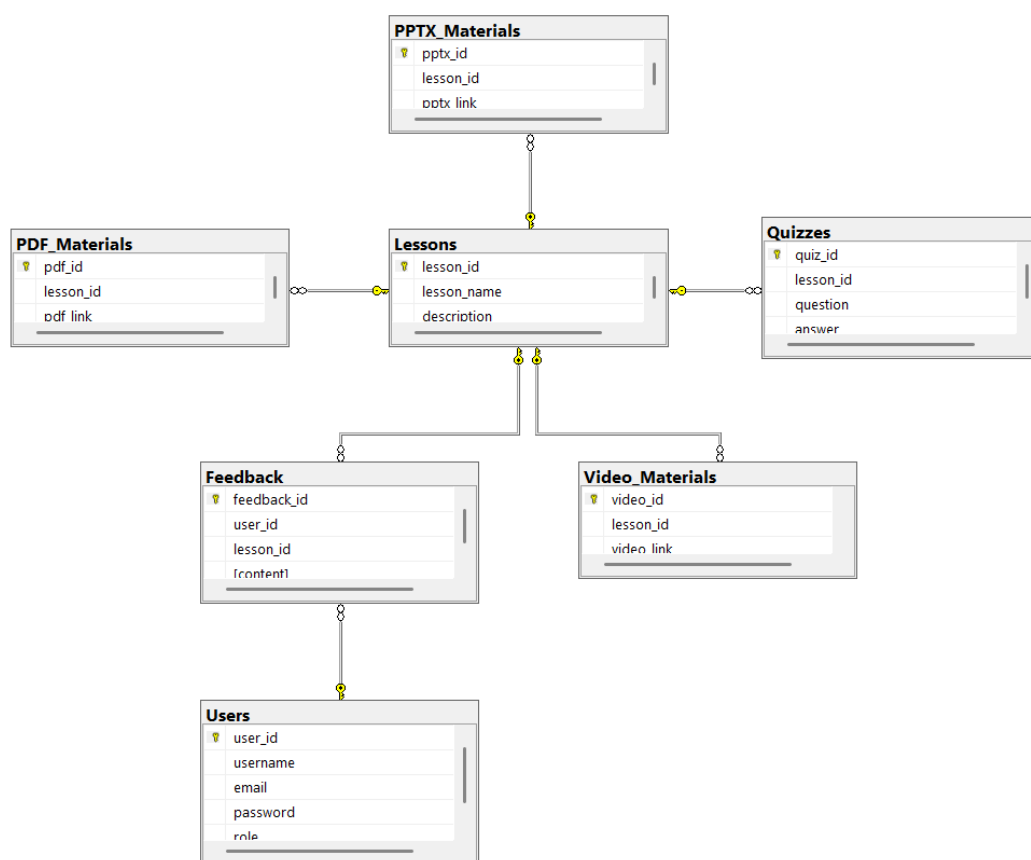
TP. HỒ CHÍ MINH – 12/2024

1. GIỚI THIỆU

Trong đề tài này, chúng tôi đã thực hiện xây dựng một ứng dụng desktop hỗ trợ học Python cơ bản. Để thực hiện đồ án này, chúng tôi đã sử dụng các thư viện pyobdc[1], webview[2] và tkinter[3] để xây dựng giao diện và kết nối cơ sở dữ liệu cho ứng dụng, cũng như áp dụng lập trình hướng đối tượng và mô hình MVC[4] trong quá trình thực hiện. Chúng tôi cũng đã đạt được kết quả là xây dựng thành công cơ sở dữ liệu cho ứng dụng, liên kết với cơ sở dữ liệu và thực hiện các chức năng như cung cấp slide bài giảng, sách tham khảo, video bài giảng và các câu hỏi ứng dụng. Trong đề tài này, chúng tôi tự phân tích thiết kế, không tham khảo các đề tài mẫu trong quá khứ.

2. MÔ TẢ CƠ SỞ DỮ LIỆU CỦA ỨNG DỤNG HỖ TRỢ HỌC PYTHON

Cơ sở dữ liệu này được chúng tôi tự thiết kế và thu thập dữ liệu. Các bảng được tổ chức với sơ đồ minh họa như sau:



Hình 1. Cơ sở dữ liệu hỗ trợ học python.

Các bảng trong cơ sở dữ liệu:

- Bảng Users (Người dùng)
 - + user_id: Khóa chính, tự động tăng, định danh người dùng.
 - + username: Tên người dùng (bắt buộc).
 - + email: Email người dùng (bắt buộc).
 - + password: Mật khẩu người dùng (bắt buộc).

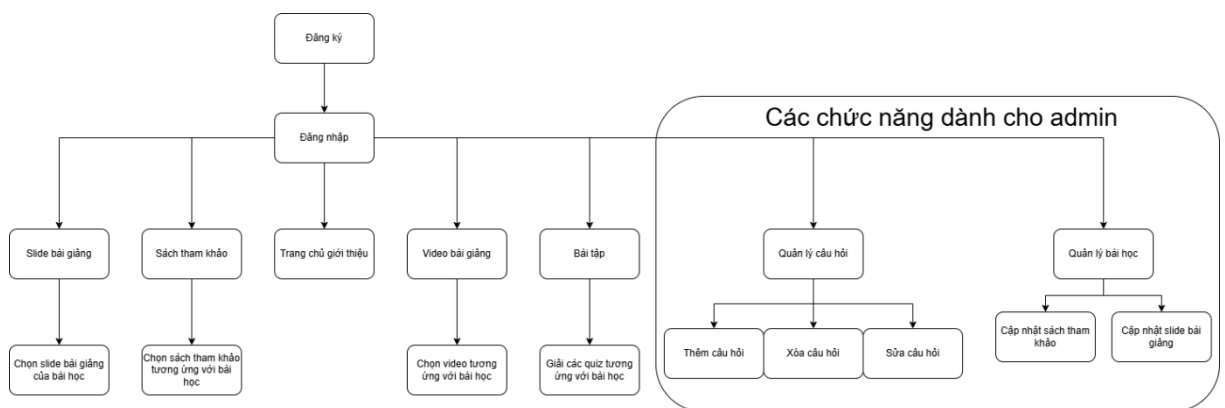
- + role: Vai trò người dùng, có thể là 'admin' hoặc 'user' (bắt buộc). Điều kiện kiểm tra đã được thiết lập thông qua ràng buộc (CHECK).
- Bảng Lessons (Bài học)
 - + lesson_id: Khóa chính, định danh bài học.
 - + lesson_name: Tên bài học (bắt buộc).
 - + description: Mô tả bài học (không bắt buộc).
- Bảng PDF_Materials (Tài liệu PDF)
 - + pdf_id: Khóa chính, định danh tài liệu PDF.
 - + lesson_id: Khóa ngoại liên kết đến bảng Lessons (bài học).
 - + pdf_link: Liên kết tới tài liệu PDF.
- Bảng PPTX_Materials (Bài giảng PPTX)
 - + pptx_id: Khóa chính, định danh bài giảng PPTX.
 - + lesson_id: Khóa ngoại liên kết đến bảng Lessons.
 - + pptx_link: Liên kết tới bài giảng PPTX.
- Bảng Video_Materials (Bài giảng Video)
 - + video_id: Khóa chính, định danh bài giảng video.
 - + lesson_id: Khóa ngoại liên kết đến bảng Lessons.
 - + video_link: Liên kết tới bài giảng video.
- Bảng Quizzes (Quiz Ôn tập)
 - + quiz_id: Khóa chính, định danh câu hỏi quiz.
 - + lesson_id: Khóa ngoại liên kết đến bảng Lessons.
 - + question: Câu hỏi quiz (bắt buộc).
 - + answer: Đáp án đúng của câu hỏi quiz (bắt buộc).
- Bảng Feedback (Phản hồi)
 - + feedback_id: Khóa chính, định danh phản hồi.
 - + user_id: Khóa ngoại liên kết đến bảng Users (người dùng).
 - + lesson_id: Khóa ngoại liên kết đến bảng Lessons (bài học).
 - + content: Nội dung phản hồi.

Ràng buộc và khóa ngoại:

- Bảng Feedback có khóa ngoại tham chiếu đến Lessons (qua lesson_id) và Users (qua user_id).
- Bảng PDF_Materials, PPTX_Materials, Video_Materials, và Quizzes đều có khóa ngoại tham chiếu đến bảng Lessons.
- Bảng Users có ràng buộc (CHECK) đảm bảo rằng giá trị của trường role chỉ có thể là 'user' hoặc 'admin'.

3. TRIỂN KHAI ỨNG DỤNG

Chúng tôi đã lên ý tưởng những chức năng tôi thiết kế một ứng dụng hỗ trợ học tập cần có, đồng thời phải kết hợp với phân quyền giữa người dùng với quản trị viên. Những chức năng chúng tôi đã đề ra bao gồm:



Hình 2. Sơ đồ chức năng hệ thống

Chương trình thực hiện 8 chức năng chính, bao gồm:

- Đăng ký: Người dùng mới có thể đăng ký bằng các thông tin username, email, password. Sau khi đăng ký, các thông tin của người dùng sẽ được lưu vào trong cơ sở dữ liệu, với ID tiếp theo tính từ ID cuối cùng, quyền hạn mặc định là user.
- Đăng nhập: Người dùng có thể đăng nhập với tên người dùng đã đăng ký từ trước được lưu trong cơ sở dữ liệu. Sau khi người dùng đăng nhập thì mới có thể sử dụng những chức năng khác.
- Slide bài giảng: Người dùng có thể chọn slide bài giảng theo thông tin tiêu đề bài học tương ứng.
- Sách tham khảo: Người dùng có thể đọc sách tham khảo có nội dung liên quan đến bài học. Mỗi bài học có thể được đề cập trong nhiều đầu sách tham khảo.

- Video bài giảng: Người dùng có thể xem các video bài giảng có nội dung liên quan đến bài học. Mỗi bài học có thể có nhiều video bài giảng.
- Bài tập: Người dùng có thể trả lời những câu hỏi trắc nghiệm với nội dung liên quan đến bài học. Mỗi câu hỏi có thể kiểm tra đúng sai của các lựa chọn, và chuyển đến câu tiếp theo.
- Chức năng quản lý cho admin:
 - + Quản lý câu hỏi: Admin có quyền thêm, xóa, sửa câu hỏi bằng cách chọn ID bài học tương ứng. Nếu thêm và sửa admin phải nhập nội dung câu hỏi và các đáp án, cùng với đáp án đúng cho câu hỏi, đồng thời tạo ID tương ứng cho câu hỏi nếu được thêm vào, và xóa toàn bộ những nội dung liên quan đến câu hỏi nếu admin muốn xóa. Những thao tác sẽ thay đổi dữ liệu trong bảng Quizzes.
 - + Quản lý bài học: Admin có quyền mở folder lưu trữ các tài liệu sách tham khảo và bài giảng được lưu trữ, cũng như có quyền thêm, xóa, sửa các tài liệu trong folder này.
- Trong tất cả các chức năng, chức năng quản lý câu hỏi là chức năng mà chúng tôi hài lòng nhất, vì chức năng này vừa thể hiện khả năng kết nối đến database, vừa thể hiện được khả năng thực hiện được các thao tác xử lý trực tiếp lên bảng Quizzes của database như thêm, xóa, sửa dữ liệu.

Áp dụng mô hình MVC trong ứng dụng:

- Các lớp Model trong ứng dụng đóng vai trò liên kết giữa ứng dụng và cơ sở dữ liệu. Chúng chịu trách nhiệm thực hiện các thao tác lưu trữ, truy xuất và xử lý dữ liệu. Trong ứng dụng này, bao gồm:
 - + UserModel đảm nhiệm quản lý thông tin người dùng, bao gồm xác thực tài khoản khi người dùng đăng nhập và đăng ký tài khoản mới. Lớp này kiểm tra tính hợp lệ của tên người dùng và mật khẩu hoặc thêm người dùng mới vào cơ sở dữ liệu.
 - + LessonModel quản lý thông tin các bài học, truy vấn cơ sở dữ liệu để lấy danh sách bài học, bao gồm mã số, tên bài học và mô tả. Lớp này đóng vai trò cung cấp dữ liệu bài học để hiển thị trên giao diện.

- + MaterialModel chịu trách nhiệm quản lý tài liệu học tập như file PDF, video hoặc slide liên quan đến từng bài học. Dựa trên mã bài học (lesson_id) và loại tài liệu cần lấy, lớp này sẽ truy xuất các liên kết hoặc tài liệu từ cơ sở dữ liệu.
- + QuizModel đảm nhiệm quản lý các câu hỏi trắc nghiệm. Nó cho phép thêm, xóa, sửa hoặc lấy danh sách câu hỏi cho từng bài học. Khi thêm câu hỏi mới, quiz_id sẽ tự động tăng lên dựa trên giá trị lớn nhất hiện có trong cơ sở dữ liệu.
- Các lớp View trong ứng dụng chịu trách nhiệm hiển thị giao diện người dùng và nhận các đầu vào từ người dùng.
 - + LoginView: Lớp này hiển thị cửa sổ đăng nhập, nơi người dùng nhập tên và mật khẩu để truy cập vào ứng dụng. Nếu người dùng nhập thông tin sai, LoginView sẽ hiển thị thông báo lỗi. Khi thông tin đăng nhập hợp lệ, nó sẽ chuyển người dùng đến màn hình chính.
 - + RegisterView: Tương tự như LoginView, nhưng dành cho việc đăng ký tài khoản mới. Lớp này nhận thông tin từ người dùng, như tên đăng nhập, email, mật khẩu, và vai trò. Sau khi xác thực thông tin, nếu thành công, người dùng sẽ được chuyển đến màn hình đăng nhập.
 - + MainApp: Đây là cửa sổ chính của ứng dụng, nơi người dùng có thể truy cập các chức năng chính như xem bài học, tài liệu, và câu hỏi trắc nghiệm. MainApp cũng hiển thị tên người dùng và quyền hạn của họ, giúp xác định các chức năng mà họ có thể sử dụng.
- Trong ứng dụng này, ApplicationController là lớp điều khiển chính của ứng dụng. Lớp này xử lý các tương tác của người dùng trong suốt quá trình sử dụng ứng dụng, như đăng nhập, đăng ký, và chuyển đổi giữa các cửa sổ giao diện khác nhau.
 - + Khởi tạo: ApplicationController khởi tạo các mô hình (Model) như UserModel, LessonModel, MaterialModel, QuizModel để quản lý dữ liệu từ cơ sở dữ liệu. Đồng thời, nó khởi tạo các cửa sổ giao diện (View) như cửa sổ đăng nhập, đăng ký, và ứng dụng chính.

- + Xử lý đăng nhập: Khi người dùng nhập thông tin đăng nhập, AppController gọi phương thức `handle_login` để xác thực người dùng qua lớp `UserModel`. Nếu thông tin đúng, nó lấy vai trò người dùng và chuyển đến màn hình chính của ứng dụng.
- + Xử lý đăng ký: Tương tự như đăng nhập, nhưng ở đây AppController sẽ gọi phương thức `handle_register` để đăng ký tài khoản mới cho người dùng. Khi thông tin hợp lệ, người dùng sẽ được chuyển đến màn hình đăng nhập.
- + Quản lý bài học và tài liệu: Lớp này cũng chịu trách nhiệm lấy dữ liệu từ các mô hình như bài học, tài liệu, và câu hỏi bài tập. Ví dụ, phương thức `get_lessons` lấy danh sách bài học từ `LessonModel`, và `get_material_links` lấy các tài liệu cho một bài học từ `MaterialModel`.
- + Quản lý câu hỏi trắc nghiệm: Các phương thức như `add_quiz`, `delete_quiz`, và `update_quiz` trong lớp AppController sẽ tương tác với `QuizModel` để thêm, xóa hoặc cập nhật câu hỏi trắc nghiệm cho các bài học. Lớp này cũng kiểm tra dữ liệu đầu vào và đưa ra thông báo lỗi nếu cần.

Kiểm thử ứng dụng:

- Kiểm thử chức năng đăng ký [Phụ lục hình 1]
- Kiểm thử chức năng đăng nhập [Phụ lục hình 2]
- Kiểm thử trang chủ [Phụ lục hình 3]
- Kiểm thử chức năng slide bài giảng [Phụ lục hình 4]
- Kiểm thử chức năng sách tham khảo [Phụ lục hình 5]
- Kiểm thử chức năng video bài giảng [Phụ lục hình 6]
- Kiểm thử chức năng bài tập [Phụ lục hình 7]
- Kiểm thử chức năng cho admin: Quản lý câu hỏi [Phụ lục hình 8]
- Kiểm thử chức năng cho admin: Quản lý bài học [Phụ lục hình 9]

4. KẾT LUẬN

Trong đồ án này, chúng tôi đã sử dụng những thư viện cần thiết và phương pháp lập trình hướng đối tượng kết hợp mô hình MVC, với điểm trọng yếu là liên kết và thao tác trên cơ sở dữ liệu để thực hiện các chức năng. Chúng tôi đã thành công thực hiện một ứng dụng hỗ trợ học Python cơ bản, đáp ứng đầy đủ chức năng về việc cung cấp tài liệu, câu hỏi vận dụng cho người dùng, cũng như là các chức năng hỗ trợ chỉnh sửa dữ

liệu trên database SQL được liên kết đến phần mềm. Chúng tôi cũng đã tạo cho ứng dụng GUI đơn giản và dùng thư viện webview để có thể hiển thị các tài liệu phục vụ cho việc học Python. Trong đó, chức năng mà chúng tôi hài lòng nhất là chức năng quản lý câu hỏi dành cho admin. Trong đồ án này, những ưu điểm chúng tôi đã phát huy được bao gồm áp dụng được mô hình MVC với cấu trúc chương trình rõ ràng, các class phù hợp với chức năng xử lý tương ứng; giao diện linh hoạt, không bị lộn xộn; kết nối đến cơ sở dữ liệu hiệu quả, không có những yêu cầu kết nối không có tác dụng.

Tuy nhiên, chúng tôi cũng mắc phải những khuyết điểm trong quá trình thực hiện, như cách thức lưu trữ dữ liệu của các tài liệu là online, và trong cơ sở dữ liệu chỉ có các đường link dẫn tới việc khó chỉnh sửa trực tiếp các tài liệu; không hỗ trợ để đăng xuất và quay ngược lại; thiếu tính đóng gói;...

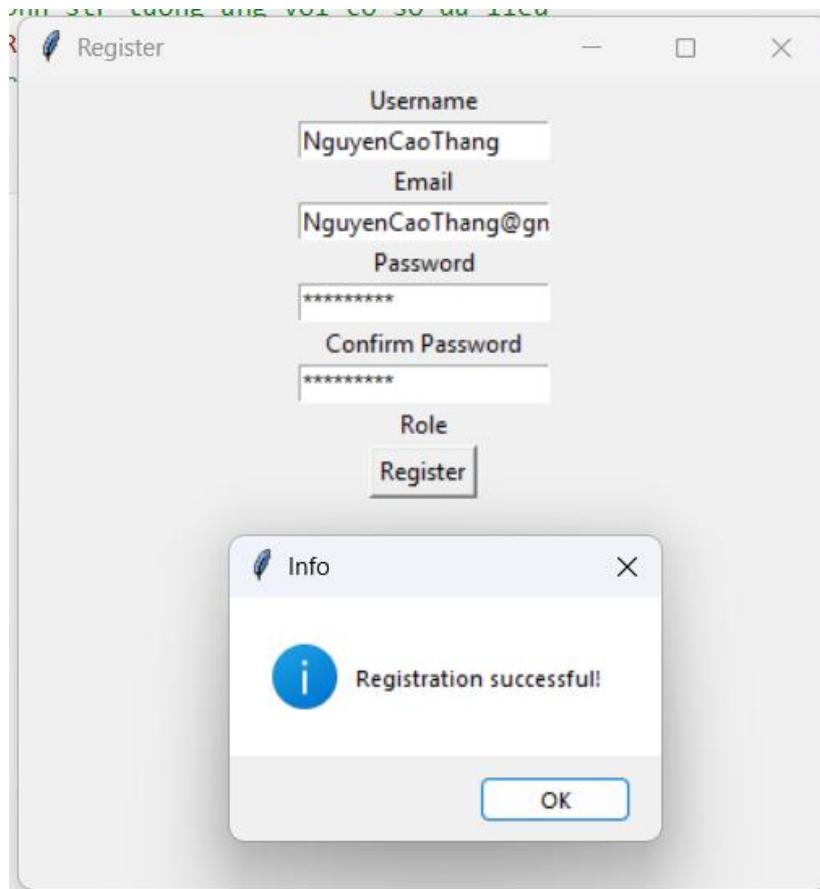
TÀI LIỆU THAM KHẢO

- [1] Home · mkleehammer/pyodbc Wiki. Link: <https://github.com/mkleehammer/pyodbc/wiki> (Ngày truy cập: 16/12/2024)
- [2] API | pywebview. Link: <https://pywebview.flowrl.com/guide/api.html> (Ngày truy cập: 16/12/2024)
- [3] tkinter — Python interface to Tcl/Tk — Python 3.13.1 documentation. Link: <https://docs.python.org/3/library/tkinter.html> (Ngày truy cập: 16/12/2024)
- [4] MVC Design Pattern – GeeksforGeeks. Link: <https://www.geeksforgeeks.org/mvc-design-pattern/> (Ngày truy cập: 16/12/2024)

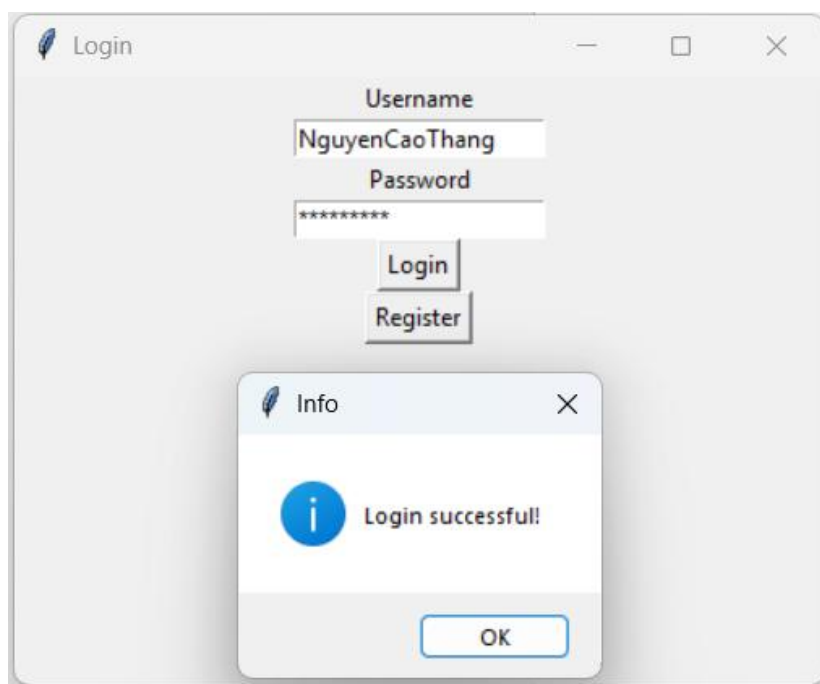
PHỤ LỤC PHÂN CÔNG NHIỆM VỤ

STT	Thành viên	Nhiệm vụ
1	Nguyễn Cao Thắng	Lên ý tưởng, thiết kế và xây dựng database, làm việc với các lớp View, thực hiện phân quyền, viết báo cáo.
2	Võ Minh Quyền	Xây dựng database, làm việc với các lớp Model và Controler, làm slide thuyết trình.
3	Võ Anh Quân	Xây dựng GUI, làm việc với các lớp View và Controller, thực hiện phân quyền.

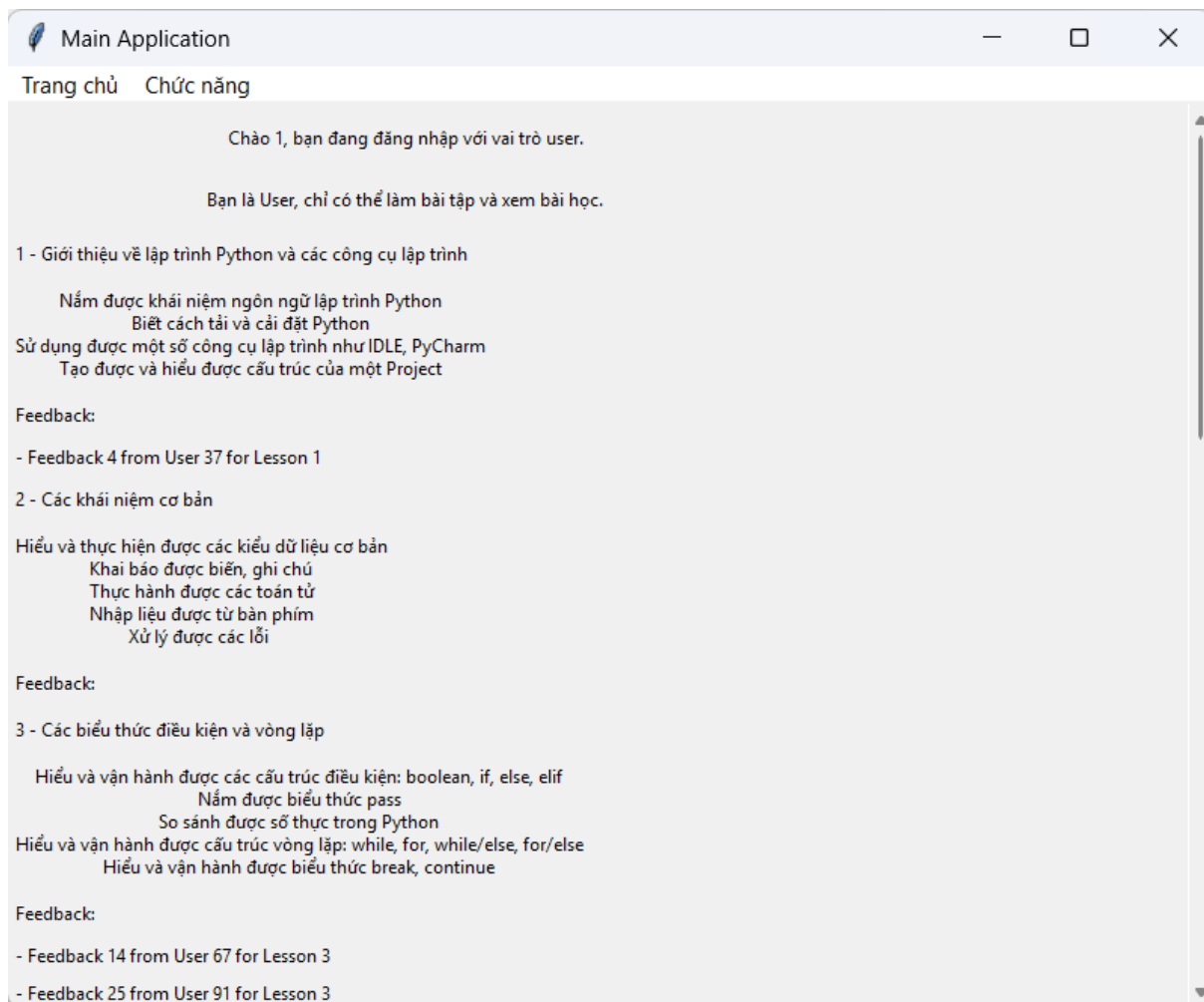
PHỤ LỤC HÌNH



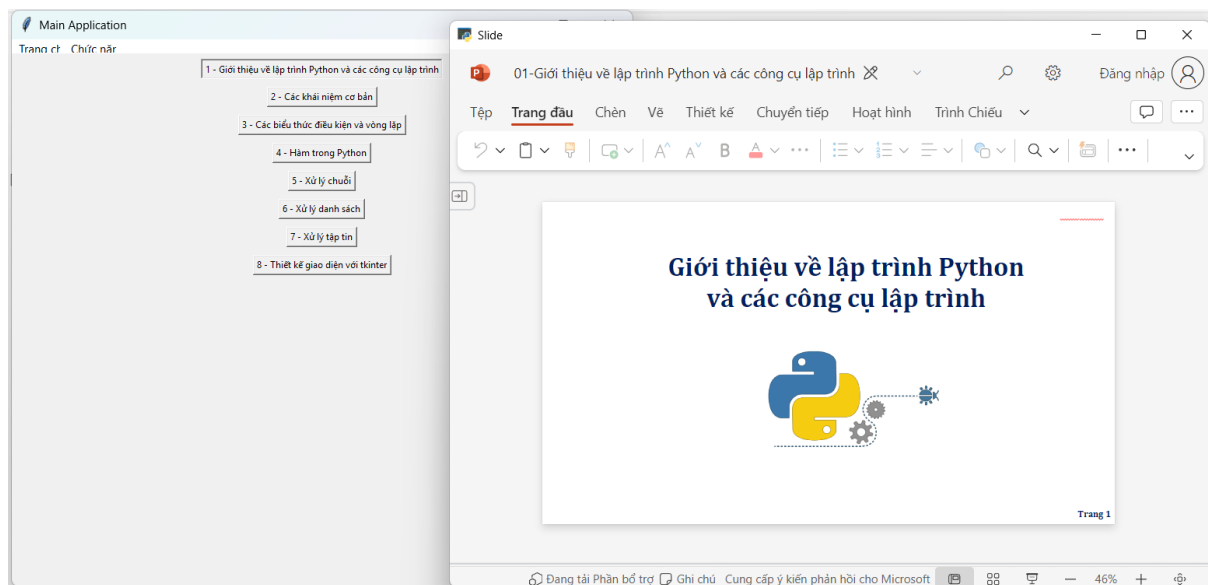
Hình 1. Chức năng đăng ký



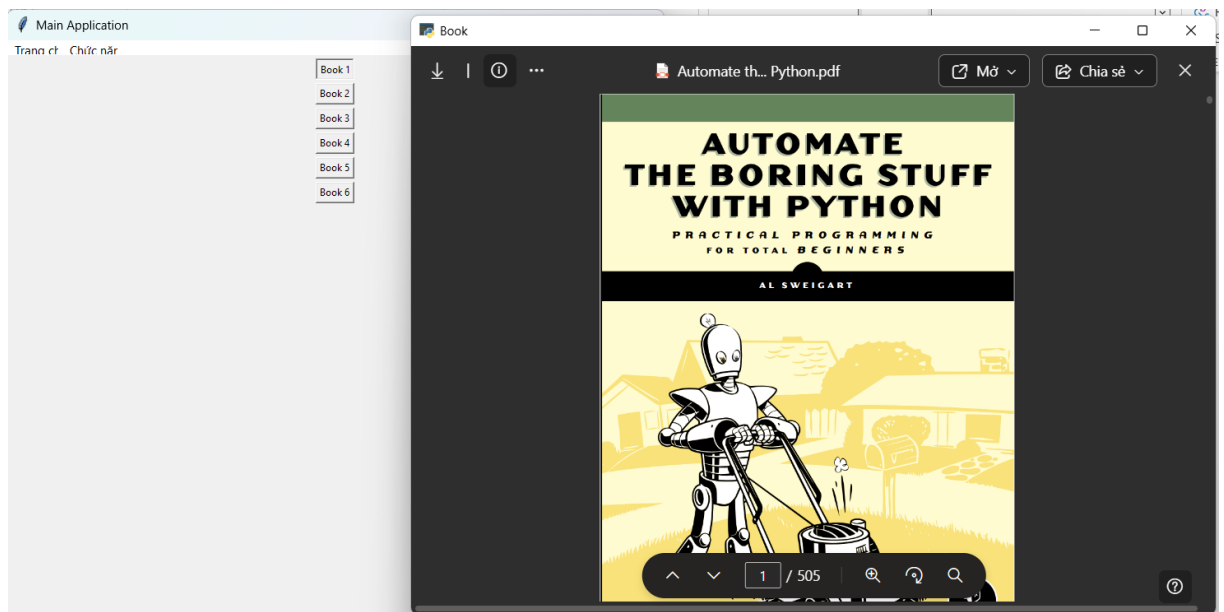
Hình 2. Chức năng đăng nhập



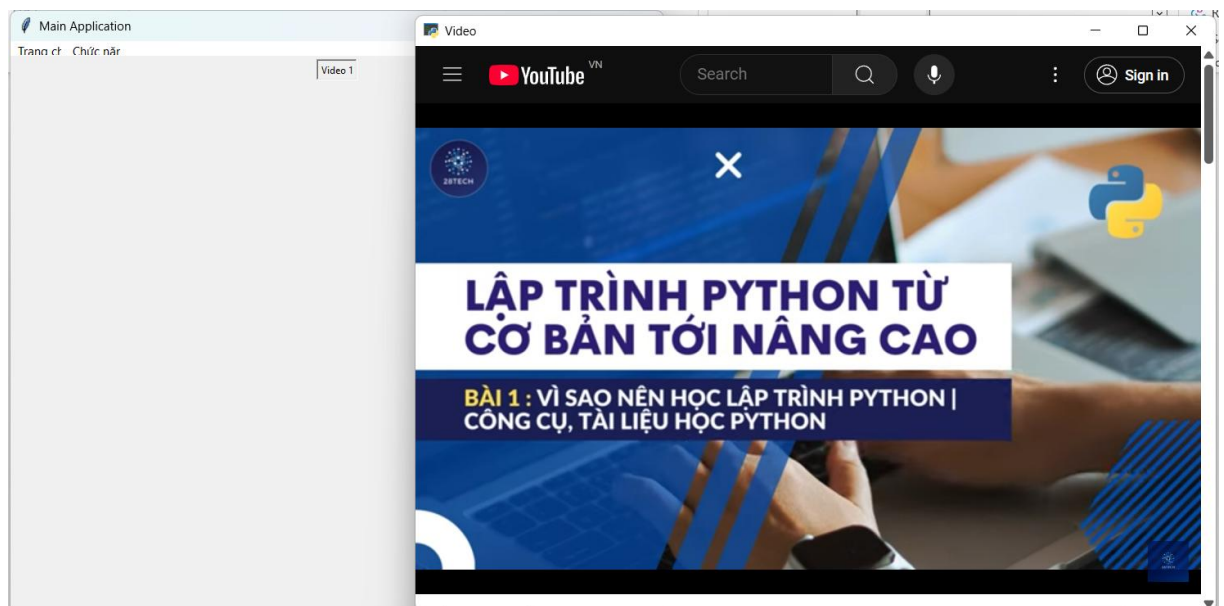
Hình 3. Trang chủ



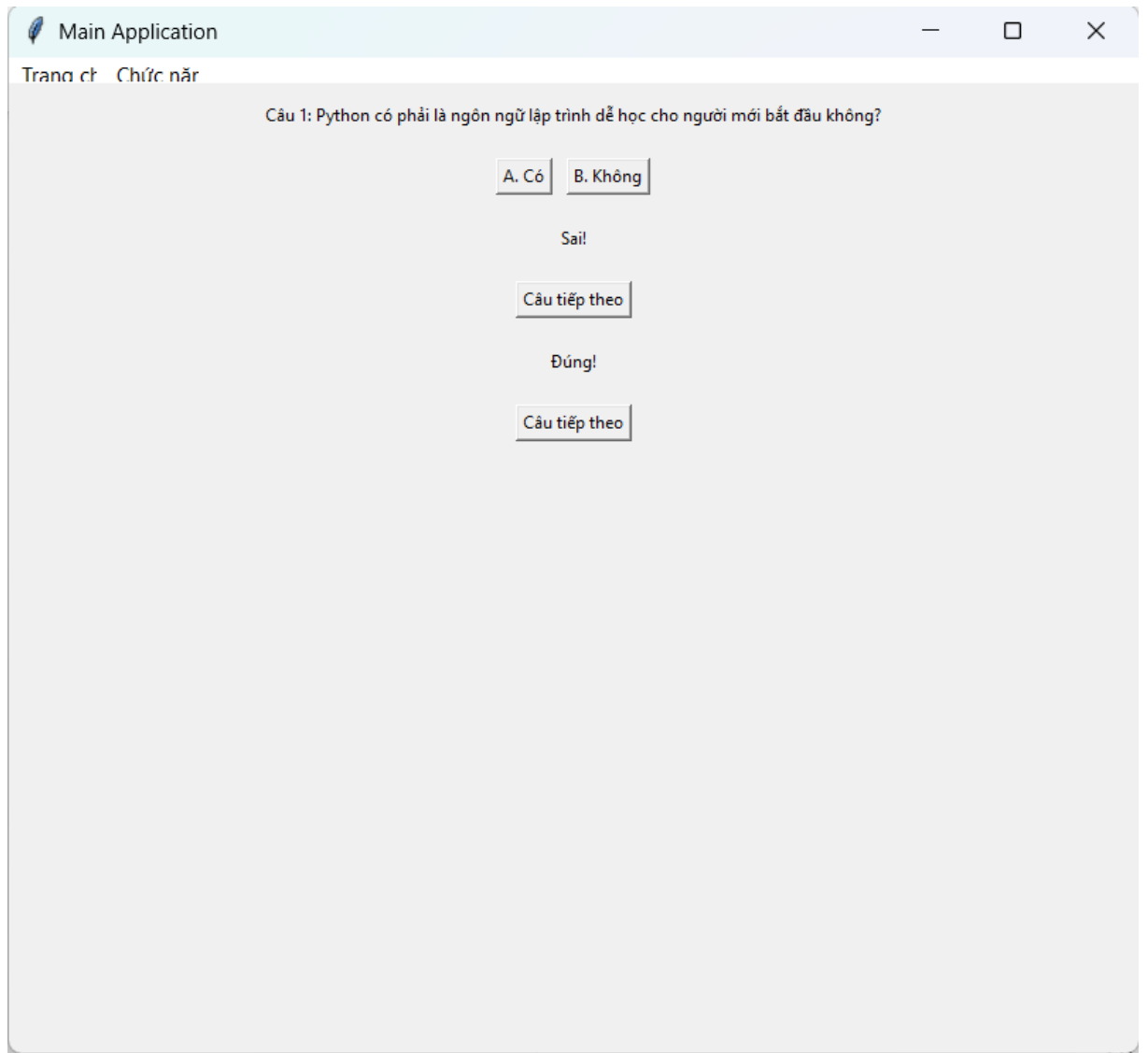
Hình 4. Slide bài giảng



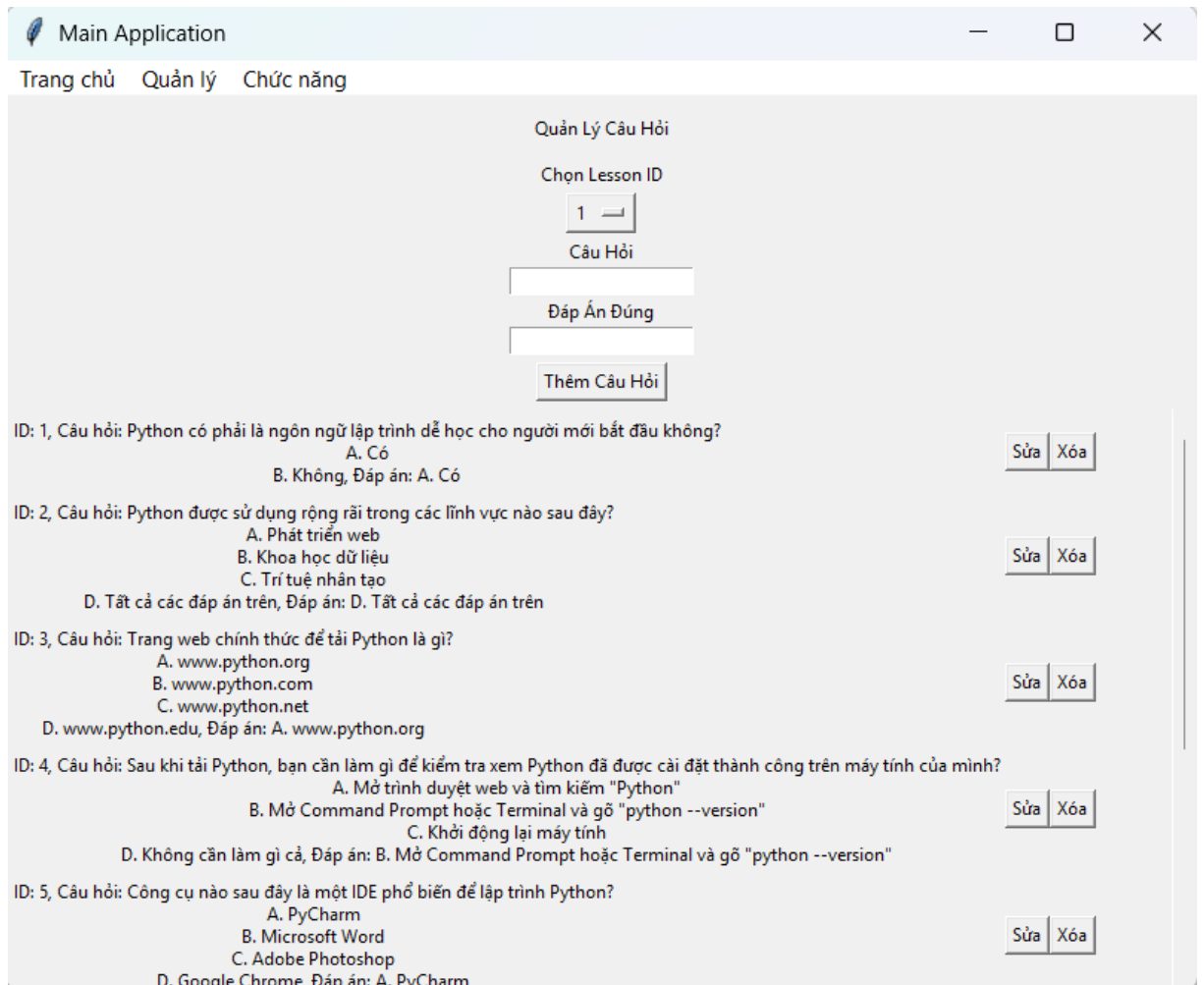
Hình 5. Sách tham khảo



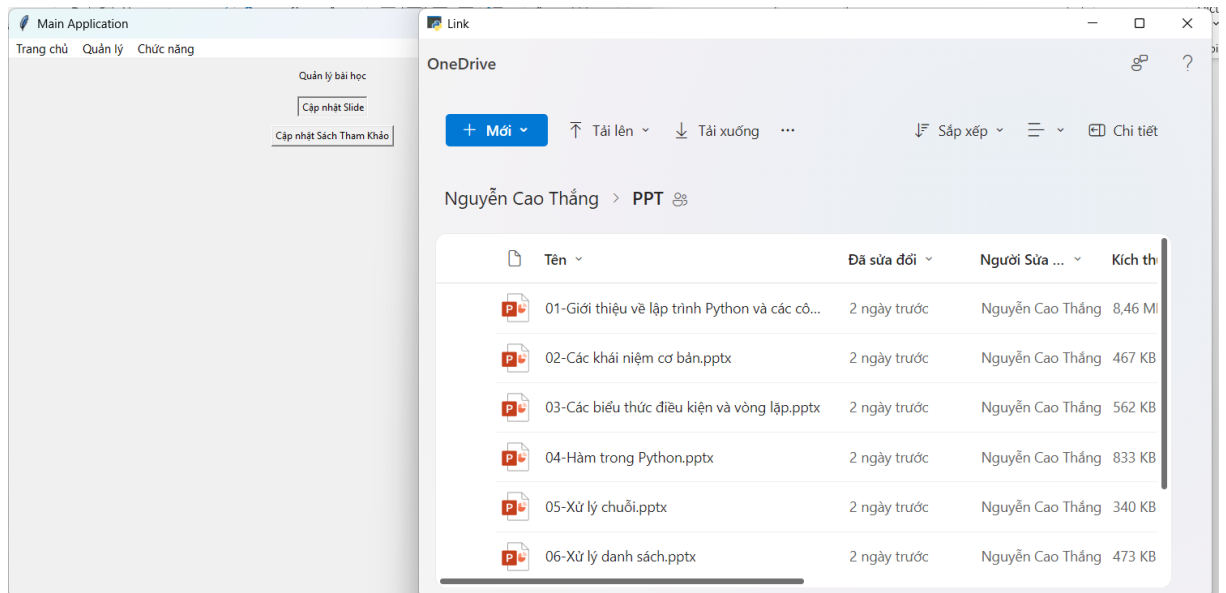
Hình 6. Video bài giảng



Hình 7. Bài tập



Hình 8. Admin – Quản lý câu hỏi.



Hình 9. Admin - Quản lý bài học