**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----o0o----

**TIỂU LUẬN HỌC PHẦN: LẬP TRÌNH PYTHON**

**TÊN ĐỀ TÀI: ỨNG DỤNG QUẢN LÝ ĐỀ THI VÀ BÀI TẬP TRẮC NGHIỆM**

**ĐỀ TÀI: 09**

**Thành phố Hồ Chí Minh,ngày tháng 06 năm 2024**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----o0o----

**TIỂU LUẬN HỌC PHẦN: LẬP TRÌNH PYTHON**

|  |  |
| --- | --- |
| **Trưởng nhóm:**  **Trần Huỳnh Phụng Quyên -2001224008**  **Thành viên:**  **Bùi Tuấn Kiệt – 2001221885**  **Trần Thanh Huy – 2033221631** | **Giảng viên hướng dẫn:**  **Phan Thị Ngọc Mai** |

**Thành phố Hồ Chí Minh, ngày tháng 06 năm 2024**

**MỤC LỤC**

[1. Giới thiệu 5](#_Toc169256379)

[1.1. Bối cảnh 5](#_Toc169256380)

[1.2. Mục tiêu 5](#_Toc169256381)

[2. Phân tích đề tài 6](#_Toc169256382)

[2.1. Cấu trúc dữ liệu và khai báo thư viện 6](#_Toc169256383)

[2.2. Xác định lớp và đối tượng 7](#_Toc169256384)

[2.3. Chức năng đăng nhập và đăng ký 8](#_Toc169256385)

[2.4. Ý tưởng phân quyền 14](#_Toc169256386)

[2.5. Xử lí chức năng chính – CRUD 17](#_Toc169256387)

[**2.5.1. Chức năng Thêm** 20](#_Toc169256388)

[**2.5.2. Chức năng Sửa** 22](#_Toc169256389)

[**2.5.3. Chức năng xóa** 23](#_Toc169256390)

[2.6. Xử lí đọc và ghi JSON 25](#_Toc169256391)

[**2.6.1. Hàm đọc** 25](#_Toc169256392)

[**2.6.2. Hàm ghi** 26](#_Toc169256393)

[2.7. Cấu hình hoạt động 26](#_Toc169256394)

[3. Kết luận 27](#_Toc169256395)

[4. Tài liệu tham khảo 28](#_Toc169256396)

**LỜI CAM ĐOAN**

Em/ chúng em xin cam đoan đề tài **Ứng dụng Quản lí Đề Thi và Bài Tập Trắc Nghiệm** do cá nhân/nhóm nghiên cứu và thực hiện.

Em/ chúng em đã kiểm tra dữ liệu theo quy định hiện hành.

Kết quả bài làm của đề tài **Ứng dụng Quản lí Đề Thi và Bài Tập Trắc Nghiệm** là trung thực và không sao chép từ bất kỳ bài tập của nhóm khác.

Các tài liệu được sử dụng trong tiểu luận có nguồn gốc, xuất xứ rõ ràng.

**(Ký và ghi rõ họ tên)**

**Trần Huỳnh Phụng Quyên**

# 1. MỞ ĐẦU

## 1.1. Bối cảnh

Trong suốt quá trình học tập và rèn luyện tại trường lớp, học sinh/sinh viên thường xuyên tiếp xúc và thực hiện những bài kiểm tra, bài thi. Từ đó, giáo viên có thể đưa ra những đánh giá về năng lực của mỗi cá nhân và xác định hướng phát triển cho học sinh/sinh viên. Do đó, đây là một phần hết sức cần thiết và quan trọng trong quá trình giáo dục & đào tạo con người. Để có thể tạo ra những bài kiểm tra và đề thi có nội dung phù hợp và hiệu quả, giáo viên cần có một công cụ hỗ trợ tạo, quản lý và dễ dàng đánh giá cấu trúc đề.

Nhận thấy được sự cần thiết của vấn đề nêu trên, phần mềm **quản lý đề thi và bài tập trắc nghiệm** đã được nghiên cứu và thực hiện nhằm giúp giáo viên có thể dễ dàng thao tác: tạo, đọc, xóa, sửa và quản lý những đề thi và bài tập để phục vụ cho công tác giáo dục tại trường.

## 1.2. Mục tiêu

Mục tiêu của đề tài là phát triển một phần mềm **quản lí đề thi và bài tập trắc nghiệm**, phục vụ các thao tác cơ bản để quản lí và theo dõi nội dung đề thi, hỗ trợ tạo tài khoản cá nhân thông qua thao tác đăng ký và đăng nhập, phân quyền dựa trên loại người dùng là quản trị viên hay người dùng thường.

# 2. PHÂN TÍCH ĐỀ TÀI

## 2.1. Cấu trúc dữ liệu và khai báo thư viện

**Câu trúc dữ liệu (folder ‘ExamApp\_Source’):**

* Folder ‘assets’: chứa thông tin các hình ảnh.
* Folder ‘JSON’: chứa thông tin các file json sử dụng trong phần mềm.
  + - * cauhoi.json: chứa nội dung thư viện đề.
      * practice.json: chứa nội dung về đề thi và bài tập trắc nghiệm.
      * role.json: chứa nội dung xác định vai trò người dùng.
      * users.json: chứa nội dung tài khoản cá nhân của người dùng.
* File ExamApp.py: file phân tích và thực hiện yêu cầu đề tài với thư viện tkinter – ngôn ngữ python.
* logo.ico: logo của ứng dụng, phục vụ cho việc đóng gói phầm mềm thành một ứng dụng hoàn chỉnh.

**Các folder đi kèm:**

* Exam Application: Chứa ứng dụng ExamApp.exe
* ExamApp - Hướng dẫn sử dụng.ppt: Hướng dẫn sử dụng ứng dụng.
* ExamApp\_Group9.docx

**Khai báo những thư viện cần thiết phục vụ cho quá trình lập trình**:

A computer screen shot of text

Description automatically generated

- **json**: Thư viện được sử dụng để làm việc với định dạng dữ liệu JSON.

- **tkinter**: Thư viện chuẩn của Python để tạo giao diện đồ họa người dùng (GUI).

- **ttk**: Một module trong tkinter cung cấp các widget được cải tiến (thường gọi là "themed widgets") so với các widget cơ bản của tkinter.

- **filedialog**: Một phần của tkinter, cung cấp các hộp thoại để có thể mở và lưu các tệp.

- **messagebox**: Một phần của tkinter, dùng để hiển thị các hộp thoại thông báo.

- **customtkinter**: CustomTkinter là một phần mở rộng của tkinter cung cấp các phần tử giao diện người dùng bổ sung. VD: CtkButton.

- **PIL** (Pillow): Thư viện để làm việc với hình ảnh.

- **requests**: Thư viện để gửi các yêu cầu HTTP trong Python.

- **BeautifulSoup**: Thư viện để phân tích cú pháp HTML và XML.

- **os**: Được sử dụng để thao tác với các tệp và thư mục.

## 2.2. Xác định lớp và đối tượng

**-** Lớp **UserManager** chứa hai phương thức tĩnh là check\_input\_login(username, password) và save\_user(username, password) dùng để xác thực thông tin đăng nhập của người dùng và lưu thông tin đăng ký.

**-** Lớp **LoginApp** được khởi tạo với các thuộc tính phục vụ việc thiết kế giao diện đăng nhập và đăng ký. Các hàm xử líthông tin đăng nhập và đăng ký được gọi trực tiếp từ lớp **UserManager**. Sau khi xác thực thành công, gọi lớp **ExamApp** để bắt đầu vào giao diện chính xử lí chức năng.

- Lớp **CustomButton** được khởi tạo để cấu hình hiển thị chung của các button được sử dụng ở lớp **ExamApp**:

* button\_xoa
* button\_xem
* button\_fix
* button\_them
* button\_crawl

- Lớp **ExamApp** được khởi tạo mở đầu với khai báo cấu hình giao diện với các hàm tạo listbox, sidebar, menubar, button,… Sau đó khởi tạo các hàm xử lí chức năng chính: các hàm CRUD (create, read, update, delete), hàm xử lí crawl dữ liệu, hàm xử lí đăng xuất.

- Hàm **main()** được xử dụng để gọi lớp **LoginApp** để khi chương trình bắt đầu sẽ tiến hành đăng nhập/đăng ký rồi sau đó chạy lớp **ExamApp.**

## 2.3. Chức năng đăng nhập và đăng ký

**Khai báo hàm và thuộc tính: \_\_init\_\_()**

* Cửa sổ chính của ứng dụng: applg.
* Tải và xử lí hình ảnh thông qua: Image.open() và ImageTk.PhoToImage().
* Hàm ẩn hiện giao diện đăng nhập/đăng ký: setup\_login, setup\_reg.
* Khung chứa giao diện đăng nhập/đăng ký: login\_frame, reg\_frame.

**A computer screen shot of a program code

Description automatically generated**

**Thiết kế giao diện đăng nhập: setup\_login()**

* Gồm các label hiển thị chữ trên cửa sổ ứng dụng như: “Welcome Back!”, “Đăng nhập với tài khoản của bạn”, “Username”, “Password”, cảnh báo đỏ khi nhập sai thông tin. Các entry nhập liệu cần thiết như username, password.
* Các button cho sự kiện nhấn ‘Đăng nhập’ và button ‘Đăng ký’ để chuyển sang giao diện đăng ký.

**A computer screen shot of text

Description automatically generated**

**A computer code with many colorful text

Description automatically generated with medium confidence**

**Thiết kế giao diện đăng ký: setup\_reg()**

* Gồm các label hiển thị chữ trên cửa sổ như: “TẠO TÀI KHOẢN!”, “Username”, “Password”. Các entry cần thiết như username, password.
* Các button cho sự kiện nhấn ‘Đăng ký’ và button trở về giao diện đăng nhập.

**A computer screen shot of text

Description automatically generated**

**A computer screen with text

Description automatically generated with medium confidence**

**Xử lí ẩn/hiện giao diện đăng nhập và đăng ký:**

**A computer screen shot of text

Description automatically generated**

**Kiểm tra thông tin đăng ký: validate\_reg()**

Sau khi nhập thông tin vào username và password, tiến hành kiểm tra:

* Nhập thông tin đầy đủ
* Mật khẩu phải đáp ứng tối thiểu 8 ký tự
* Kiểm tra người dùng có tồn tại hay chưa (**validate\_username\_reg()**)

Khi xác thực thành công – thông báo thành công.

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

**Kiểm tra thông tin đăng nhập: validate\_login()**

Sau khi đã có tài khoản, tiến hành so sánh username và password với file **users.json** chứa thông tin tài khoản cá nhân. Nếu điền thông tin chưa đầy đủ sẽ thông báo lỗi “Kiểm tra lại username hoặc password”. A screen shot of a computer code

Description automatically generated 

Ý tưởng phân quyền: Sau khi xác thực thành công, ghi đè vào file **role.json** để thực hiện xác định vai trò của người dùng và từ đó lớp **ExamApp** có thể đọc và thực hiện phân quyền.



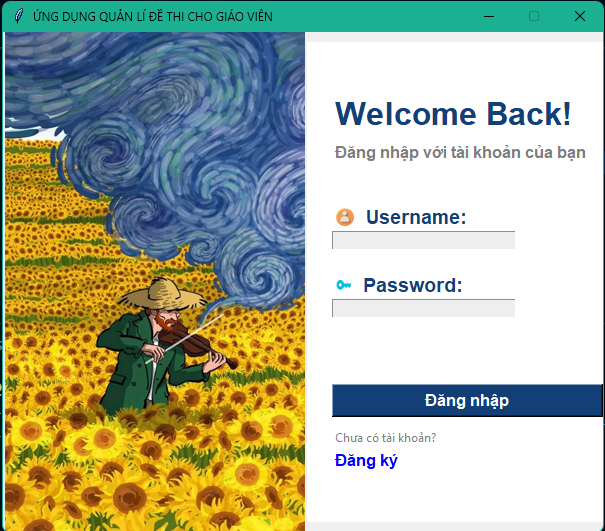


**Di chuyển đến giao diện chính: Start\_app()**

**A computer screen with text

Description automatically generated with medium confidence**

**Chạy giao diện chức năng đăng nhập và đăng ký:**

 Cùng với các yêu cầu về hình thức nhập thông tin đăng nhập, thông qua nút ‘**Đăng nhập**’ sẽ đến giao diện chính.

Nếu chưa có tài khoản, thực hiện nhấn nút ‘**Đăng ký’** để thực hiện đăng ký tài khoản.

A screenshot of a computer

Description automatically generated

Ở giao diện đăng ký, nếu đã có tài khoản, nhấn nút ‘**Đăng nhập**’ ở phía dưới nút ‘**Đăng ký**’ và chuyển về giao diện đăng nhập.

## 2.4. Ý tưởng phân quyền

Để có thể xác định được vai trò của người dùng, trong quá trình đăng nhập được chạy ở hàm **validate\_login()** thuộc lớp **LoginApp**. Ngay sau khi xác thực thông tin đăng nhập của người dùng là chính xác, ghi đè vào file mang tên role.json để lưu dữ liệu về vai trò người dùng đăng nhập hiện tại. Ngoài ra, đối với mỗi user được tạo mới từ đăng ký, mặc định sẽ có role ‘user’ vì chỉ tồn tại một admin trong phầm mềm.

A computer screen shot of a program code

Description automatically generated

Ý tưởng phân quyền dựa trên vai trò:

* **Admin**: có tất cả quyền thao tác hiện có trên ứng dụng.
* **User**: Được thao tác các quyền cơ bản như CRUD và không được phép tác động lên dữ liệu thư viện đề (thêm, xóa, sửa) ngoài quyền được xem. Ngoài ra, khi crawl dữ liệu từ website, user chỉ có quyền xem và không có quyền thêm câu hỏi mới vào thư viện đề.

A screen shot of a computer program

Description automatically generated

Nếu người dùng đăng nhập vào là admin thì role.json có nội dung như sau:

A screenshot of a computer

Description automatically generated

Nếu người dùng đăng nhập vào là user thường thì role.json có nội dung như sau:

A screen shot of a computer

Description automatically generated

Việc tạo file role.json chứa thông tin vai trò giúp hỗ trợ xác định được người dùng hiện tại đang sử dụng là ai. Từ đó, ở lớp phía dưới như ExamApp sẽ thực hiện đọc file role.json và nhận biết được người dùng hiện tại thuộc vai trò admin hay user, từ đó thực hiện phân quyền dựa trên nội dung file role.json.

A screen shot of a computer code

Description automatically generated

Một số ví vụ về cách thực hiện phân quyền vào hàm:

VD: Trong hàm **Read\_KhoDe()**, sau khi nhận biết được người dùng là admin hay user bằng cách gọi hàm **read\_role()**, thực hiện ẩn hoặc hiện những button dựa trên quyền của vai trò. Với admin có tất cả quyền và được thấy tất cả button, user sẽ bị hạn chế khung nhìn.



A screen shot of a computer program

Description automatically generated

Giao diện thực tế so sánh sự khác nhau giữa vai trò admin và user:

A screenshot of a computer

Description automatically generated

Hình 1. Mục Thư Viện Đề của Admin

A screenshot of a chat

Description automatically generated

Hình 2. Mục Thư Viện Đề của User

## 2.5. Xử lí chức năng chính – CRUD

Trong khai báo ở lớp **ExamApp** (lớp xử lí chức năng chính), có phần khai báo:



Mục đích của biến này là vì màn hình hiển thị sử dụng listbox, vì thế current\_list giữ vai trò quan trọng trong việc xác định vị trí thao tác chuột trên listbox, từ đó trỏ chính xác đến file cần thao tác.

A computer screen shot of a code

Description automatically generated

A computer screen with text

Description automatically generated with medium confidence

* **Các hàm xử lí chính trong lớp ExamApp**:
  + Các hàm khai báo và thiết kế giao diện:
    - **\_\_init\_\_()**
    - **setuo\_gui()**
    - **create\_sidebar()**
    - **create\_listbox()**
    - **create\_menubar()**
    - **create\_buttons()**
    - **create\_button()**
    - **Trangchu()**
    - **login()**
  + **read\_role():** Đọc vai trò.
  + **Read\_Name():** Phương thức chung cho việc đọc ‘exam’ hay ‘test’ trong file practice.json.
  + **Read\_NameDT()** và **Read\_NameBT()**: Kế thừa từ Read\_Name().
  + **Read\_KhoDe():** Đọc nội dung file cauhoi.json (thư viện đề).
  + **Display\_Name():** Phương thức cấu hình hiển thị chung cho Read\_NameDT và Read\_NameBT (màu, cỡ chữ,…).
  + **Delete\_Name():** Xóa mã đề.
  + **Delete\_Question():** Xóa câu hỏi trong mã đề.
  + **Delete\_KhoDe():** Xóa câu hỏi trong thư viện đề.
  + **Add\_Question():** Thêm câu hỏi vào đề thi hoặc bài tập dựa trên lựa chọn thêm mới mã đề/tạo câu hỏi mới cho đề.
  + **Add\_KhoDe():** Thêm câu hỏi mới cho thư viện đề.
  + **Add\_FromCrawl():** Lấy câu hỏi được crawl từ website để sử dụng trong thư viện đề.
  + **QuestionKD():** Phương thức xử lí chung cho chức năng Sửa/Thêm câu hỏi trong thư viện đề.
  + **Add\_QuestionKD()** và **Edit\_QuestionKD():** Được kế thừa từ QuestionKD() cho việc thêm hoặc sửa.
  + **Edit\_Question():** Sửa câu hỏi có trong Đề thi hoặc Bài tập.
  + **View\_Question():** Đọc nội dung câu hỏi và đáp án có trong Đề thi hoặc Bài tập.
  + **CrawlToFile():** Hàm gửi yêu cầu lấy dữ liệu từ website.
  + **Crawl\_Display():** từ dữ liệu đã lấy ở CrawlToFile(), in lên listbox.
  + **Logout():** Xử lí đăng xuất.

### **2.5.1. Chức năng Thêm**

* Thêm một mã đề mới (thực hiện bởi hàm **Add\_Question()** thuộc lớp **ExamApp**):
  + Tại **menubar** 🡪 **CRUD** 🡪 Chọn ‘**Thêm**’ 🡪 Hộp thoại xuất hiện

A screenshot of a computer

Description automatically generated

* Thêm câu hỏi mới vào đề (thực hiện bởi hàm **Add\_Question()** thuộc lớp ExamApp):
  + Tại **menubar** 🡪 **CRUD** 🡪 Chọn ‘**Thêm**’ 🡪 Hộp thoại xuất hiện

A screenshot of a computer

Description automatically generated

* Thêm câu hỏi mới từ thư viện đề (thực hiện bởi hàm **Add\_KhoDe()** thuộc lớp ExamApp):
  + Tại **menubar** 🡪 **CRUD** 🡪 Chọn ‘**Thư viện đề**’ 🡪 Hộp thoại xuất hiện

A screenshot of a computer

Description automatically generated

* Thêm câu hỏi mới vào thư viện đề (thực hiện bởi hàm **Add\_QuestionKD()** thuộc lớp ExamApp):
  + Vai trò Admin: **Trang chủ** 🡪 **Thư viện đề** 🡪 Chọn ‘**Thêm**’ 🡪 Hộp thoại xuất hiện

A screenshot of a computer

Description automatically generated

* Thêm câu hỏi crawl được từ website vào thư viện đề (thực hiện bởi hàm **Add\_FromCrawl()**) – Chọn câu hỏi và nhấn nút ‘Lấy’:
  + Vai trò Admin: **menubar** 🡪 **FILE** 🡪 Chọn ‘**From Website**’ 🡪 Chọn câu hỏi 🡪 Chọn ‘**Lấy**’ 🡪 Hộp thoại xuất hiện

A screenshot of a computer

Description automatically generated

Sau khi thực hiện xong, nhấp nút ‘**Lưu**’ – Thao tác hoàn thành.

### **2.5.2. Chức năng Sửa**

* Sau khi chọn một mã đề trên listbox (Đề thi hoặc Bài tập), vào **menubar** 🡪 **CRUD** 🡪 chọn ‘**Sửa**’ 🡪 Hộp thoại chọn câu hỏi xuất hiện (được thực hiện bởi hàm **Edit\_Question()**:

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Hình 3. Giao diện sau khi nhấn '**Chọn**'

* Chức năng Sửa ở thư viện đề (Được thực hiện bởi hàm **Edit\_QuestionKD()**):
  + Tại giao diện **Thư viện đề** 🡪 Chọn ‘**Sửa**’ 🡪 Hộp thoại xuất hiện

A screenshot of a computer

Description automatically generated

Sau khi thực hiện xong, nhấp nút ‘**Lưu**’ – Thao tác hoàn thành.

### **2.5.3. Chức năng xóa**

* Xóa một mã đề: Khi chọn một mã đề trên màn hình và nhấn nút ‘Xóa’ sẽ hiện ra thông báo yes or no – Nếu yes, cập nhật lại màn hình (Thực hiện bởi hàm **Delete\_Name()** trong lớp ExamApp).

A colorful background with white and pink stripes

Description automatically generated

A screenshot of a computer

Description automatically generated

* Xóa câu hỏi trong mã đề, sau khi chọn xem sẽ đọc nội dung đề (**View\_Question()**), hiển thị button ‘Xóa’ thực hiện xóa câu hỏi (**Delete\_Question()**):

A white and pink rectangle

Description automatically generated

A screenshot of a computer

Description automatically generated

* Xóa câu hỏi trong thư viện đề. Trong giao diện Thư Viện Đề 🡪 Chọn câu hỏi 🡪 Chọn ‘**Xóa**’ 🡪 Thông báo xóa thành công:

A screenshot of a chat

Description automatically generated

## 2.6. Xử lí đọc và ghi JSON

### **2.6.1. Hàm đọc**

Sử dụng **with** để mở tệp, đảm bảo rằng tệp sẽ được đóng tự động sau khi khối lệnh with được thực thi, ngay cả khi xảy ra lỗi.

**open("JSON/ten\_file.json", "r")**: Mở các tệp json trong chế độ đọc (**r**).

**json.load(file)**: Đọc và phân tích dữ liệu từ tệp JSON, kết hợp với đọc utf-8 để có thể sử dụng tiếng Việt có dấu mà không lỗi định dạng.

Sau cùng là xử lí ngoại lệ "**except** –": FileNotFoundError, json.JSONDecodeError.

A computer code with colorful text

Description automatically generated with medium confidence



A close-up of a computer screen

Description automatically generated

### **2.6.2. Hàm ghi**

Sử dụng **with** để mở tệp, đảm bảo rằng tệp sẽ được đóng tự động sau khi khối lệnh with được thực thi, ngay cả khi xảy ra lỗi.

**open("JSON/ten\_file.json", "w")**: Mở các tệp json trong chế độ đọc (**w**).

**json.load(file)**: Đọc và phân tích dữ liệu từ tệp JSON, kết hợp với đọc utf-8 để có thể sử dụng tiếng Việt có dấu mà không lỗi định dạng.

Sau cùng là xử lí ngoại lệ "**except** –": IOError.



A computer code with colorful text

Description automatically generated with medium confidence

## 2.7. Cấu hình hoạt động

Sau khi chạy xong lớp **Login\_App**, hàm cuối cùng được thực hiện là **Start\_app()** – có chức năng chạy giao diện chính:

A computer code with text

Description automatically generated with medium confidence

Ở giao diện chính, nhấn button ‘**Đăng xuất**’(được thực hiện bởi hàm **logout()** trong lớp ExamApp, được gọi đến hàm **main()** và nội dung hàm được khởi tạo để chạy lại lớp LoginApp để xử lí đăng nhập:

A screen shot of a computer code

Description automatically generated

A video game screen with a blue rectangle and green lines

Description automatically generated

Lựa chọn này có thể bắt đầu từ: Đăng nhập 🡪 Giao diện chính 🡪 Đăng nhập 🡪 Giao diện chính/Đăng nhập 🡪 Giao diện chính 🡪 Kết thúc (**app.destroy()**)

# 3. KẾT LUẬN

Với đề tài ‘**Ứng dụng Quản lí Đề Thi và Bài Tập Trắc Nghiệm’**, sinh viênđược làm quen với thư viện tkinter – hỗ trợ giao diện, nghiên cứu và thực hiện phần mềm nhằm mục đích hỗ trợ giáo viên trong quá trình giảng dạy tại trường, được triển khai với các thao tác đơn giản như tạo, thêm, xóa, sửa,… từ đó làm bước đệm cho những phát triển sau này, cải thiện kĩ năng cá nhân và nâng cao tư duy lập trình một cách hợp lí. Phần mềm còn hỗ trợ thao tác đăng ký và đăng nhập giúp tạo sự riêng tư và cá nhân trong quá trình sử dụng, từ đó nâng cao trải nghiệm cá nhân khi sử dụng phần mềm.

# 4. TÀI LIỆU THAM KHẢO

[1] tkinter — Python interface to Tcl/Tk. (n.d.). Python Documentation. Retrieved June 14, 2024, from <https://docs.python.org/3/library/tkinter.html>

[2] Fitzpatrick, M. (2022, June 14). Create Python GUI with tkinter. Python GUIs. <https://www.pythonguis.com/tutorials/create-gui-tkinter/>

[3] Python, R. (2019, March 6). Using PyInstaller to easily distribute Python applications. Realpython.com; Real Python. <https://realpython.com/pyinstaller-python/>

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**BIÊN BẢN HỌP NHÓM**

(*V/v Phân công công việc /Đánh giá hoàn thành /Họp nhóm định kỳ 1*)

**1. Thời gian, địa điểm, thành phần tham dự.**

**1.1. Thời gian:**

**1.2. Địa điểm:**

**1.3. Thành phần tham dự:**

+ Chủ trì: Trần Huỳnh Phụng Quyên

+ Tham dự: 3

+ Vắng: 0

**2. Nội dung cuộc họp**

**2.1. Công việc các thành viên như sau\* (Bắt buộc không được để trống)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ** | **Tên** | **Đóng góp tỷ lệ %** | **Phân công** | **Mức độ hoàn thành** |
| 25 | 2001224008 | Trần Huỳnh | Phụng Quyên | 100% | Code, Word | 100% |
| 15 | 2001221885 | Bùi | Tuấn Kiệt | 100% | Code, Word | 100% |
| 13 | 2033221631 | Trần | Thanh Huy | 100% | Code, Word | 100% |

**2.2. Ý kiến của các thành viên:** Đề nghị ghi rõ ý kiến của từng thành viên, đồng ý hay không đồng ý với ý kiến của nhóm trưởng, hoặc phản biện với các ý kiến của các thành viên khác,...

**2.3. Kết luận cuộc họp**

Thống nhất lại nội dung cuộc họp sau khi có ý kiến của từng thành viên

Cuộc họp đi đến thống nhất và kết thúc lúc ....giờ.... phút cùng ngày.

|  |  |
| --- | --- |
| **Thư ký**  **(Ký và ghi rõ họ tên)** | **Chủ trì**  **(Ký và ghi rõ họ tên)** |