**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin**.

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC

**LẬP TRÌNH PYTHON**

**ĐỀ TÀI: Xây game Blackjack (Chapter 9) với GUI :chia bài, hit/stand, tính điểm.**

  Sinh viên: Hoàng Đức Hội

  Lớp: K58KTPM

MSV: K225480106085

Giáo viên GIẢNG DẠY: Nguyễn Văn Huy

Link GitHub: https://github.com/duchoi04/Python



**Thái Nguyên – 2025**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC: LẬP TRÌNH PYTHON**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

*Sinh viên:* *Hoàng Đức Hội*

*Lớp*: K58KTPM        *Ngành:* Kỹ thuật máy tính

*Giáo viên hướng dẫn:* Nguyễn Văn Huy

*Ngày giao đề* *Ngày hoàn thành* :

*Tên đề tài :*Xây game Blackjack (Chapter 9) với GUI: chia bài, hit/stand, tính điểm.

*Yêu cầu :*

**Đầu vào – đầu ra:**

Đầu vào: Nút “Hit”, “Stand”.

Đầu ra: Hiển thị lá bài (Label/Icon), điểm, kết quả.

**Tính năng yêu cầu:**

Sử dụng class BJ\_Card, BJ\_Deck, BJ\_Hand, BJ\_Game.

GUI cập nhật khi rút bài.

Xử lý bust (>21) và so sánh với dealer.

**Kiểm tra & kết quả mẫu:**

Rút A,10 → tài – Hiển thị “Blackjack!”

**Các bước triển khai:**

Import module cards, games.

Class GUI: Frame người chơi và dealer.

Button “Hit” và “Stand” gọi phương thức game.

Cập nhật kết quả trên Label/Messagebox.

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**MỤC LỤC**

**[LỜI NÓI ĐẦU](#_Toc24158)** [4](#_Toc24158)

**[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN](#_Toc14993)** [5](#_Toc14993)

**[CHƯƠNG 1: GIỚI THIỆU ĐẦU BÀI](#_Toc28372)** [6](#_Toc28372)

[1.1. Đặt vấn đề 6](#_Toc24486)

[1.2. Mục tiêu chương trình 6](#_Toc27599)

[1.3. Tính năng nổi bật 6](#_Toc21800)

[1.4. Thách thức khi thực hiện 7](#_Toc12945)

[1.5. Kiến thức vận dụng 7](#_Toc17705)

**[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT](#_Toc26455)** [8](#_Toc26455)

[2.1. Cấu trúc dữ liệu trong Python 8](#_Toc9118)

[2.2. Lập trình hướng đối tượng (OOP) 8](#_Toc13378)

[2.3. Tkinter và Pillow 8](#_Toc26654)

[2.4. Xử lý sự kiện GUI 8](#_Toc14497)

[2.5. Quản lý trạng thái và luồng game 9](#_Toc27270)

**[CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH](#_Toc29565)** [10](#_Toc29565)

[3.1. Sơ đồ khối hệ thống 10](#_Toc8658)

[3.2. Sơ đồ khối các thuật toán chính 11](#_Toc2968)

[3.3. Cấu trúc dữ liệu 12](#_Toc16042)

[3.4. Chương trình 13](#_Toc10294)

**[CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT LUẬN](#_Toc10104)** [14](#_Toc10104)

[4.1. Thực nghiệm 14](#_Toc26993)

[4.2. Kết luận 17](#_Toc18941)

**[KẾT LUẬN](#_Toc17379)** [18](#_Toc17379)

[Xin chân thành cảm ơn! 19](#_Toc5824)

**LỜI NÓI ĐẦU**

Trong thời đại công nghệ phát triển mạnh mẽ, việc xây dựng các giải pháp ứng dụng tương tác như trò chơi điện tử không chỉ giúp người học rèn luyện kỹ năng lập trình mà còn phát triển huy tư duy logic, khả năng thiết kế phần mềm theo đối tượng hướng dẫn và trải nghiệm người dùng (GUI). Đề tài “Xây dựng trò chơi Blackjack với giao diện GUI” là một ví dụ công cụ có thể giúp sinh viên tiếp cận các khái niệm về thiết kế chương trình, quản lý trò chơi logic và phát triển giao diện trực quan trong Python.

Blackjack là trò chơi bài nổi tiếng về luật chơi đơn giản nhưng hấp dẫn, Đòi hỏi sự kết hợp giữa may mắn và chiến lược. Qua cài đặt chương trình trò chơi này, người thực thi không chỉ sử dụng kiến trúc nền tảng như danh sách, từ điển, chức năng và lớp trong Python mà còn tiếp cận và ứng dụng cài đặt hướng đối tượng (OOP), đồ họa giao diện thư viện (tkinter), đồng thời học cách xử lý các vấn đề thực tế như so sánh số lượng, xử lý thẻ đặc biệt (A), cập nhật giao diện và tạo trải nghiệm người dùng.

Báo cáo này sẽ trình bày chi tiết từ cơ sở lý thuyết, hệ thống thiết kế, xây dựng chương trình cho các bước thực hiện, đánh giá kết quả và rút gọn phần thảo luận. Với tinh thần học hỏi và rèn luyện kỹ năng lập trình thực tế, sản phẩm hoàn thiện không chỉ giúp xây dựng cố định kiến trúc chuyên ngành mà còn mở rộng khả năng sáng tạo và tư vấn giải quyết vấn đề trong phát triển phần mềm.

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

.......................................................................................................................................................................

.......................................................................................................................................................................

.......................................................................................................................................................................

.......................................................................................................................................................................

.......................................................................................................................................................................

.......................................................................................................................................................................

### *Thái Nguyên, ngày….tháng…..năm 20....*

## **GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

**CHƯƠNG 1: GIỚI THIỆU ĐẦU BÀI**

**1.1. Đặt vấn đề**

Blackjack là một trò chơi bài phổ biến trên toàn thế giới, nổi bật với luật chơi đơn giản nhưng đòi hỏi sự tính toán, chiến lược và may mắn. Việc xây dựng một ứng dụng Blackjack với giao diện đồ họa (GUI) không chỉ giúp người dùng giải trí mà còn là cơ hội để rèn luyện kỹ năng lập trình, tư duy logic và khả năng thiết kế phần mềm.

* Trong bối cảnh học tập, việc hiện thực hóa game Blackjack bằng Python và thư viện Tkinter giúp sinh viên:

- Vận dụng kiến thức lập trình hướng đối tượng (OOP).

- Làm quen với thiết kế giao diện người dùng (UI).

- Hiểu cách tổ chức, quản lý dữ liệu và xử lý sự kiện trong ứng dụng thực tế.

**1.2. Mục tiêu chương trình**

- Xây dựng một ứng dụng Blackjack hoàn chỉnh với giao diện trực quan, thân thiện.

- Hỗ trợ các thao tác cơ bản: đặt cược, rút bài (Hit), dừng (Stand), chơi lại (Restart), xem lịch sử (History).

- Hiển thị lá bài bằng hình ảnh, cập nhật điểm số và trạng thái trò chơi theo thời gian thực.

- Quản lý số dư, tiền cược, thống kê thắng/thua/hòa.

- Lưu và hiển thị lịch sử các ván chơi.

- Cung cấp hướng dẫn luật chơi cho người mới.

**1.3. Tính năng nổi bật**

- Giao diện đẹp, dễ sử dụng, phù hợp cho cả người mới bắt đầu.

- Hiển thị lá bài bằng hình ảnh PNG, tăng tính trực quan.

- Tự động tính điểm, xử lý các trường hợp đặc biệt như Blackjack, Bust.

- Chức năng đặt cược, cập nhật số dư, thống kê kết quả.

- Lưu và hiển thị lịch sử các ván chơi gần nhất.

- Hộp thoại thông báo kết quả, hướng dẫn luật chơi.

**1.4. Thách thức khi thực hiện**

- Quản lý logic game Blackjack (xử lý điểm Aces, so sánh điểm, xác định kết quả).

- Hiển thị động các lá bài bằng ảnh, cập nhật giao diện mượt mà.

- Quản lý trạng thái các nút điều khiển, lịch sử, số dư.

- Đảm bảo chương trình hoạt động ổn định, dễ mở rộng và bảo trì.

**1.5. Kiến thức vận dụng**

- Lập trình hướng đối tượng (OOP) với Python.

- Xử lý giao diện với Tkinter.

- Xử lý ảnh với Pillow.

- Quản lý dữ liệu với list, dict.

- Xử lý sự kiện GUI, cập nhật trạng thái động.

- Tổ chức module, phân chia chức năng hợp lý.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1. Cấu trúc dữ liệu trong Python**

- List: Dùng để lưu trữ các lá bài trong bộ bài, trên tay người chơi và dealer.

Ví dụ: `cards = [Card('A', '♠'), Card('10', '♥')]`

-Dict: Dùng để ánh xạ ký hiệu lá bài sang tên file ảnh, quản lý các trạng thái.

Ví dụ: `rank\_map = {'A': 'ace', 'J': 'jack', ...}`

**2.2. Lập trình hướng đối tượng (OOP)**

- Card: Đại diện cho một lá bài (rank, suit), có phương thức tính điểm.

- Deck: Quản lý bộ bài, trộn bài, chia bài.

- Hand: Quản lý các lá bài trên tay người chơi hoặc dealer, tính điểm tổng.

- Game: Quản lý logic ván chơi, chia bài, rút bài, xác định kết quả.

- BlackjackGUI: Quản lý giao diện, trạng thái game, các thao tác của người dùng.

**2.3. Tkinter và Pillow**

**- Tkinter:** Thư viện chuẩn của Python để xây dựng giao diện người dùng, quản lý các widget, sự kiện, layout.

**- Pillow:** Thư viện xử lý ảnh, dùng để mở, resize và hiển thị hình ảnh lá bài lên giao diện.

**2.4. Xử lý sự kiện GUI**

- Sử dụng các hàm callback cho các nút bấm (Hit, Stand, Bet, Restart, History).

- Cập nhật giao diện động khi trạng thái game thay đổi (ví dụ: khi rút bài, khi kết thúc ván).

**2.5. Quản lý trạng thái và luồng game**

- Sử dụng các biến trạng thái (`bet`, `balance`, `win\_count`, ...) để kiểm soát luồng chơi.

- Cập nhật trạng thái các nút (bật/tắt) dựa trên tiến trình của ván bài.

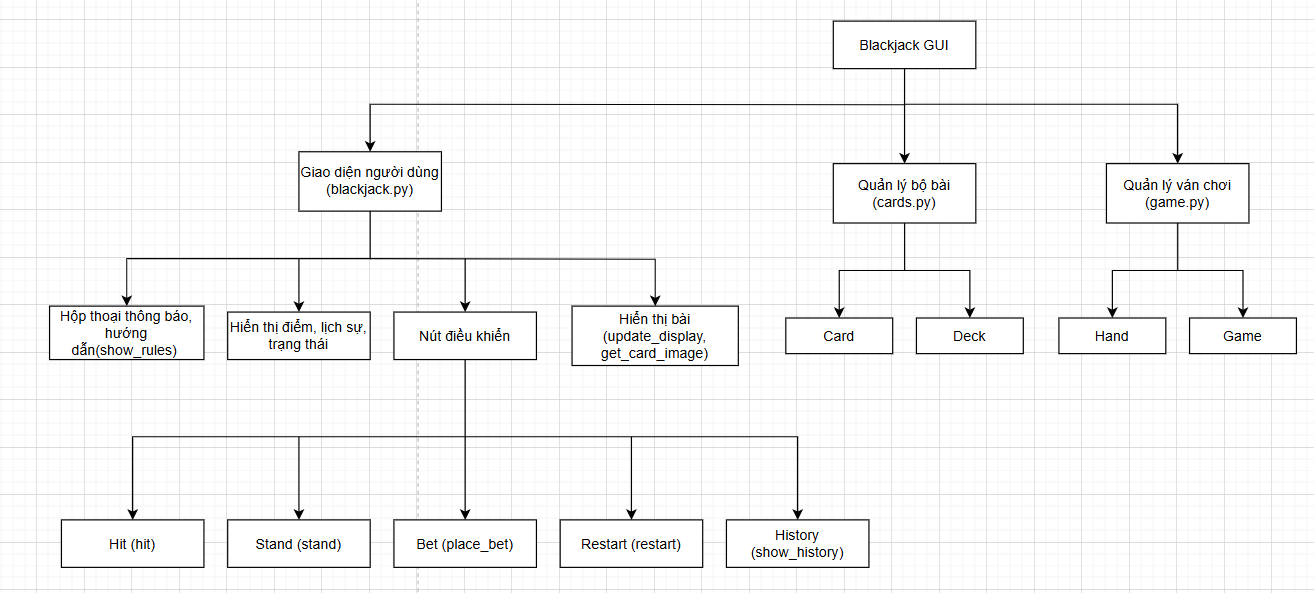
- Lưu lịch sử các ván chơi để người dùng có thể xem lại.

**CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH**

**3.1. Sơ đồ khối hệ thống**

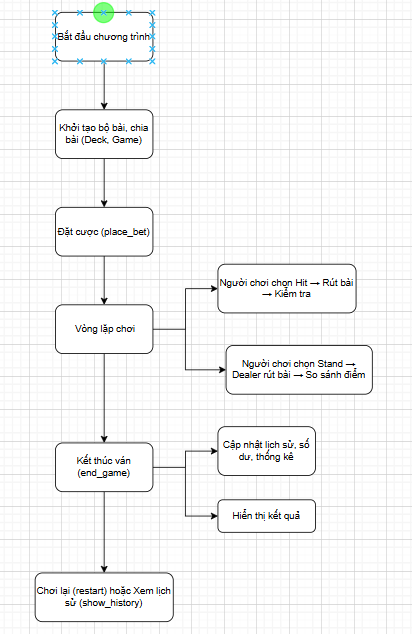
**- Module chính:** blackjack.py (giao diện và điều khiển game)

**- Module phụ:** cards.py (quản lý lá bài, bộ bài), games.py (quản lý logic game)



*Hình 3.1 Biểu đồ phân cấp chức năng*

**3.2. Sơ đồ khối các thuật toán chính**

****

*Hình 3.2 Sơ đồ khối các thuật toán chính*

- Khởi tạo game: Tạo bộ bài, chia bài cho người chơi và dealer.

- Đặt cược: Nhập số tiền cược, trừ vào số dư, kiểm tra hợp lệ.

- Rút bài (Hit): Thêm lá bài cho người chơi, kiểm tra Bust/Blackjack.

- Dừng (Stand): Dealer rút bài theo luật, so sánh điểm.

- Kết thúc ván: Cập nhật lịch sử, số dư, thông báo kết quả.

- Restart: Tạo lại bộ bài, reset trạng thái.

- Hiển thị lịch sử: Hiển thị 10 ván gần nhất.

**3.3. Cấu trúc dữ liệu**

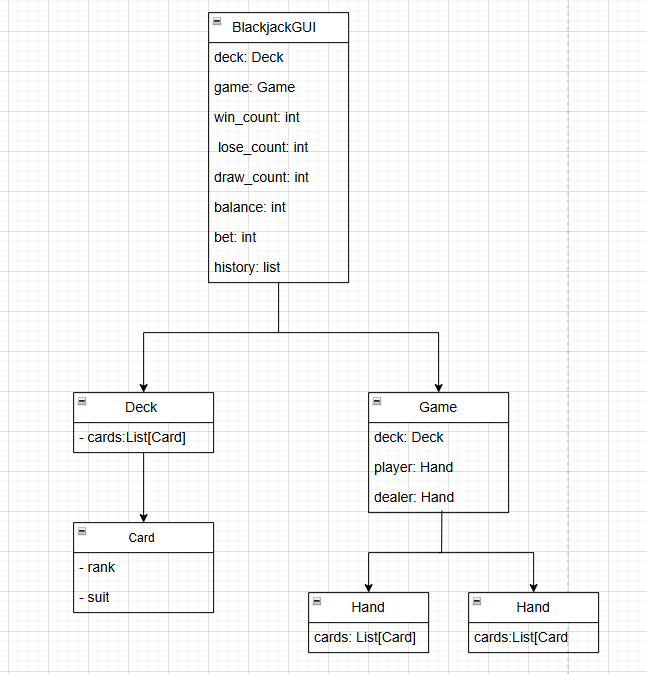
**- Card:** rank, suit.

**- Deck:** list các Card.

**- Hand:** list các Card.

**- Game:** quản lý Deck, Hand của player và dealer.

**- BlackjackGUI:** lưu trạng thái điểm, số dư, cược, lịch sử (list các string).

****

*Hình 3.3 Sơ đồ dữ liệu*

**3.4. Chương trình**

Các hàm chính:

**- `get\_card\_image`:** Lấy ảnh lá bài từ thư mục, trả về đối tượng ảnh để hiển thị.

**- `update\_display`:** Cập nhật giao diện khi có thay đổi (bài, điểm, trạng thái nút).

**- `place\_bet`:** Đặt cược, kiểm tra hợp lệ, cập nhật số dư.

**- `hit`:** Rút bài cho người chơi, kiểm tra Bust/Blackjack.

**- `stand`:** Dealer rút bài, so sánh điểm, xác định kết quả.

**- `end\_game`:** Kết thúc ván, cập nhật lịch sử, số dư, khóa các nút.

**- `restart`:** Bắt đầu ván mới, reset trạng thái.

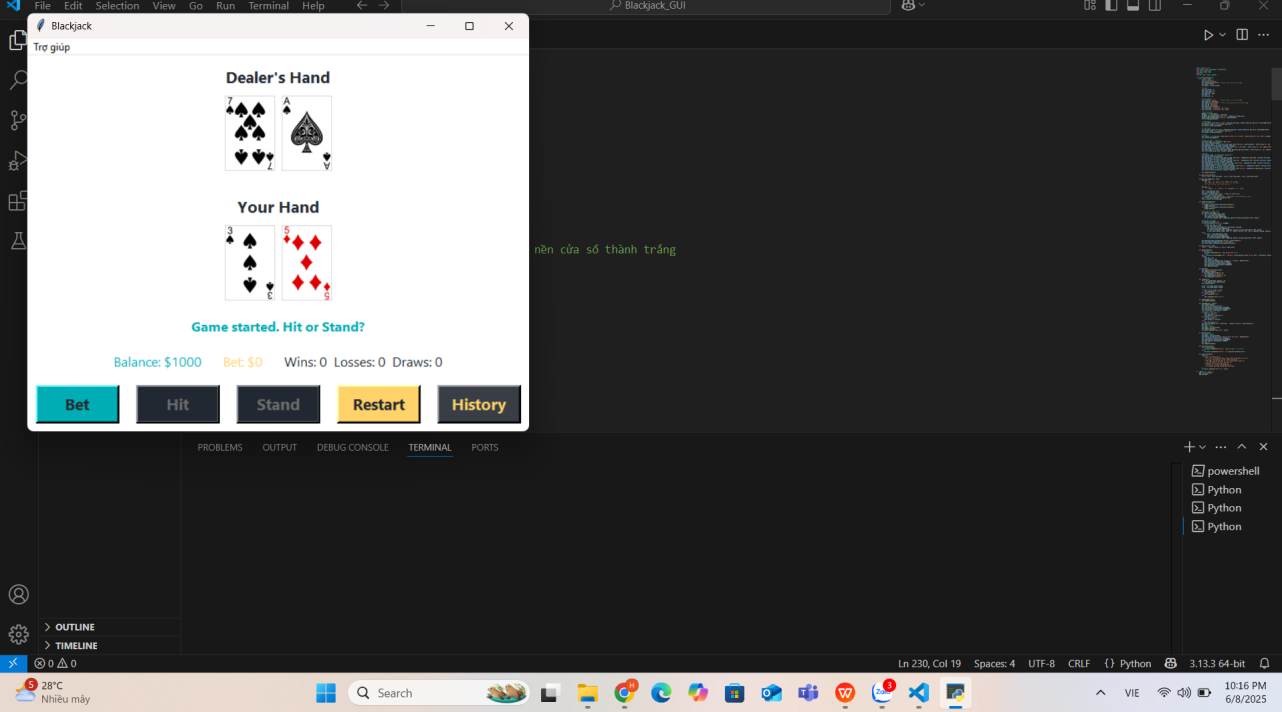
**- `show\_history`:** Hiển thị lịch sử ván chơi.

**- `show\_rules`:** Hiển thị luật chơi.

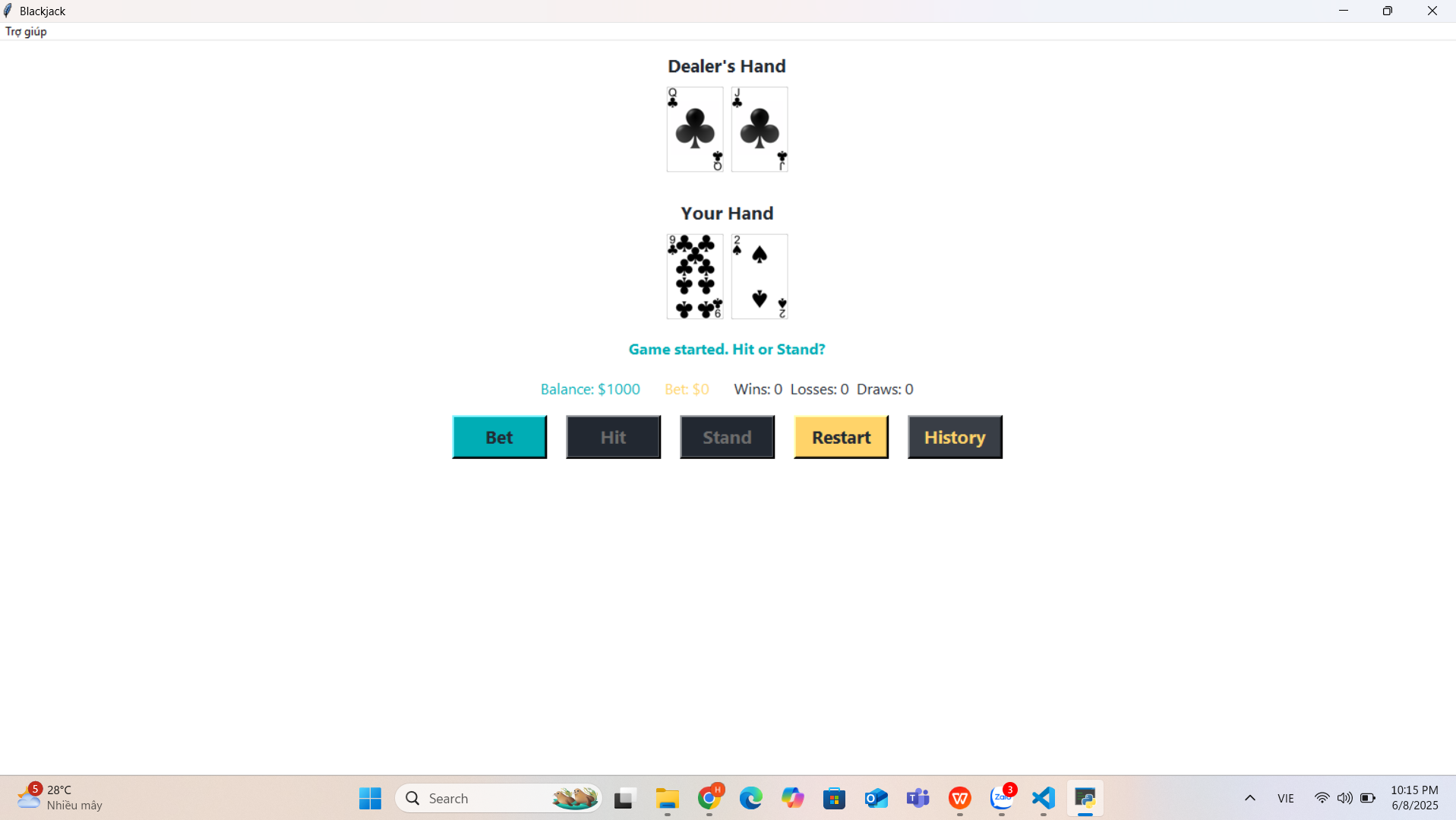
**CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT LUẬN**

**4.1. Thực nghiệm**

- Chạy chương trình:

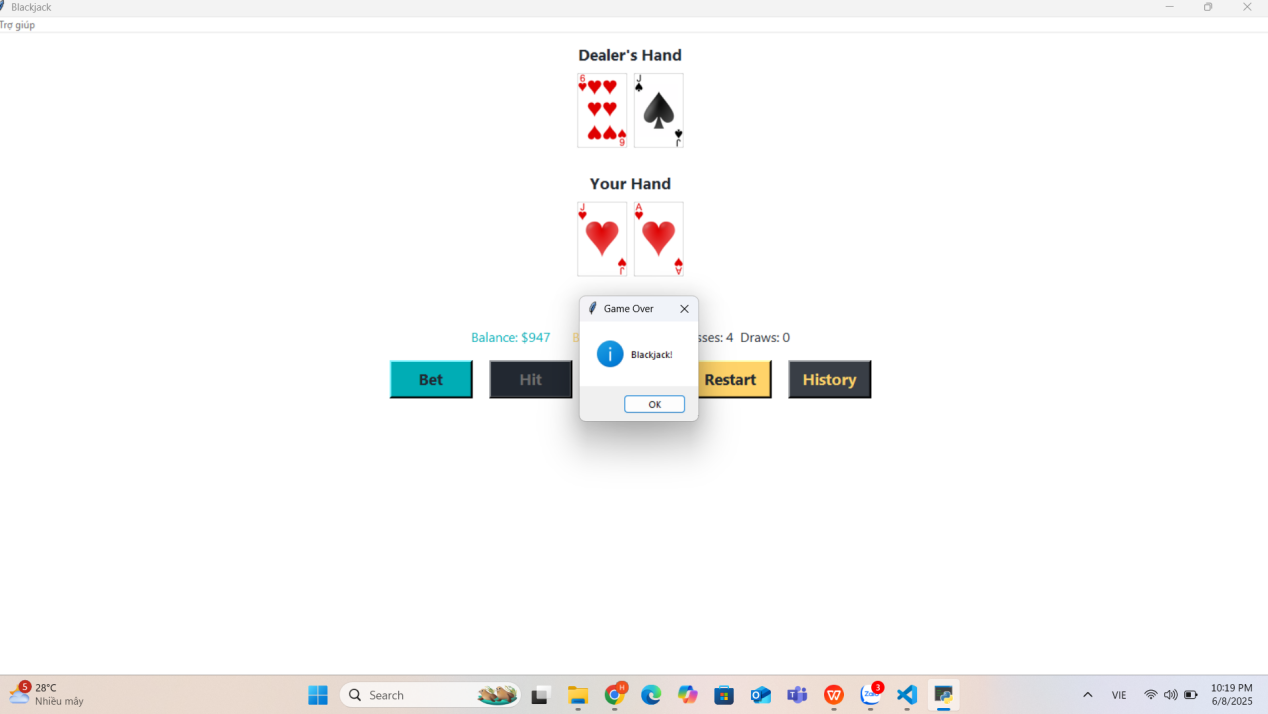


- Đặt cược, rút bài, dừng, xem lịch sử:

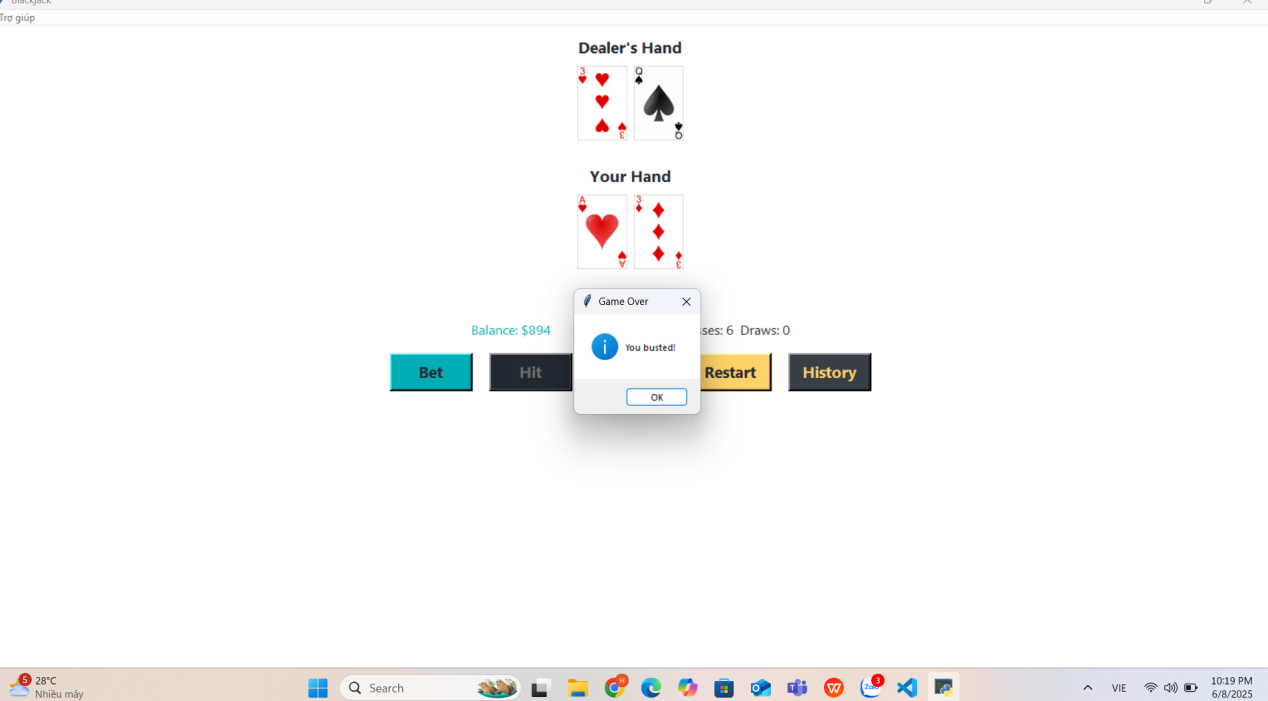


- Kiểm tra các trường hợp đặc biệt:

- Rút A, 10 → hiện “Blackjack!”

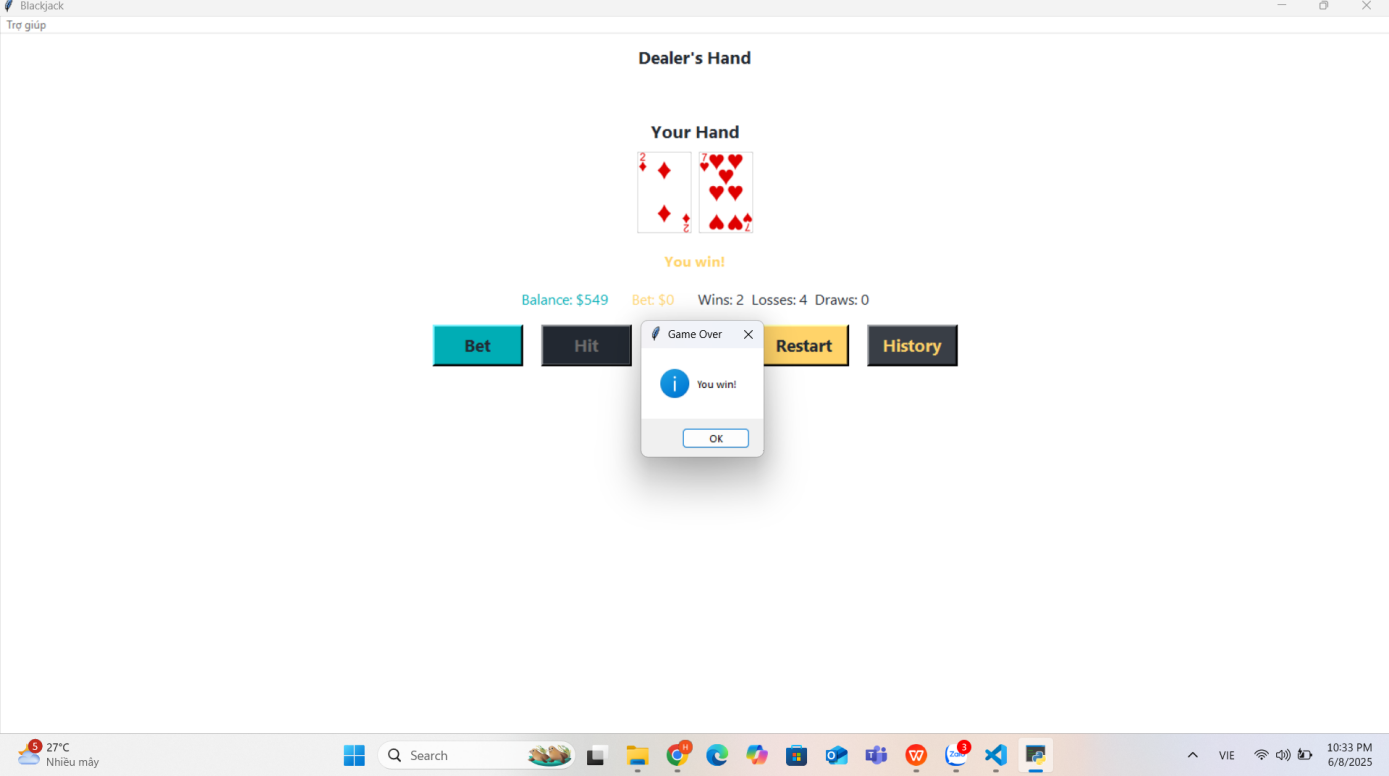


- Rút quá 21 → hiện “You busted!”

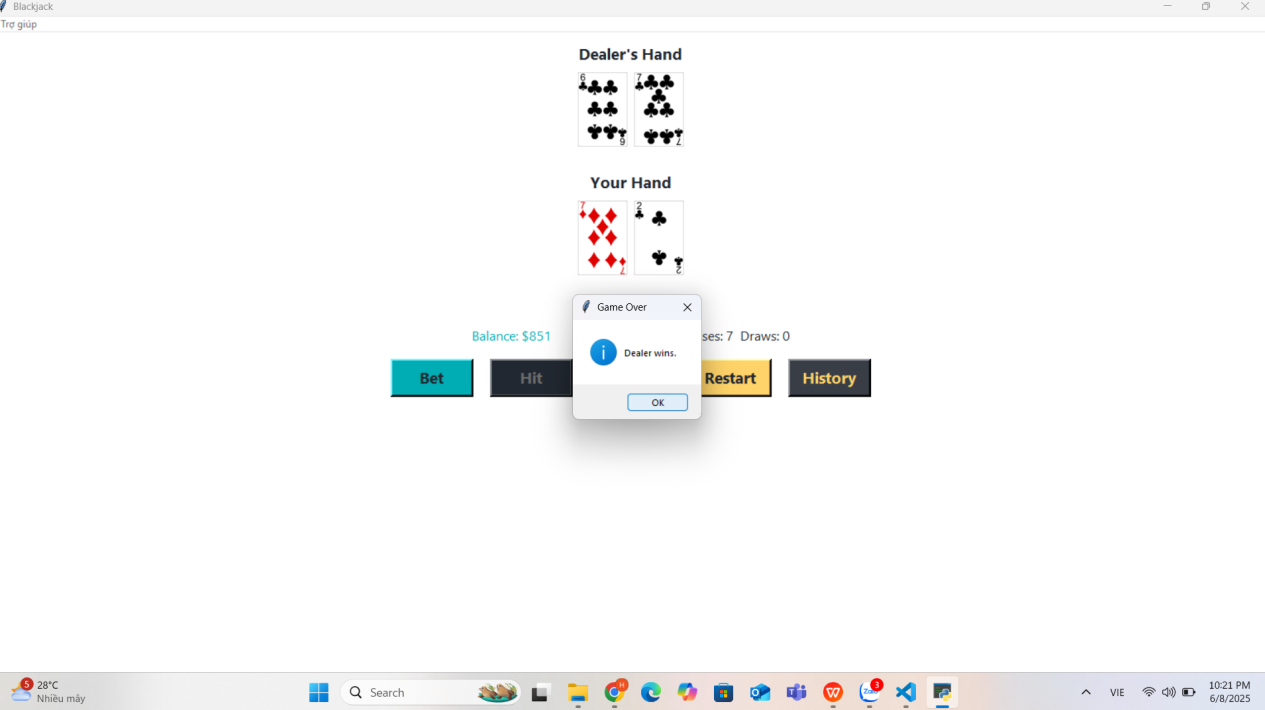


* So điểm với dealer:

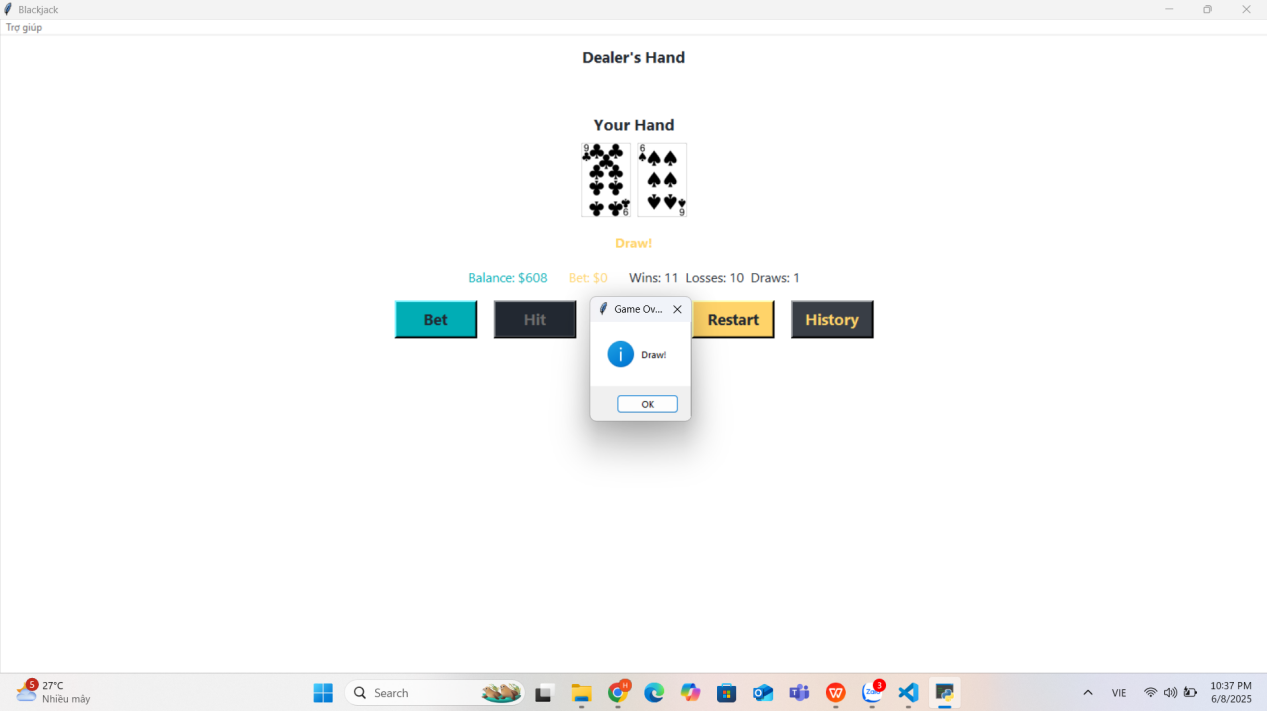
-Thắng:



-Thua:



-Hòa:



Đánh giá:

- Giao diện trực quan, dễ sử dụng, phù hợp với người mới.

- Các chức năng hoạt động đúng yêu cầu, cập nhật trạng thái mượt mà.

- Lịch sử ván chơi hiển thị rõ ràng, dễ theo dõi.

**4.2. Kết luận**

- Sản phẩm đã làm được:

- Xây dựng thành công game Blackjack có GUI, đặt cược, lưu lịch sử, hiển thị ảnh lá bài, hướng dẫn luật chơi.

- Bài học kinh nghiệm:

- Nâng cao kỹ năng lập trình hướng đối tượng, xử lý giao diện động với Tkinter.

- Hiểu sâu hơn về quản lý trạng thái, sự kiện trong ứng dụng GUI.

- Rèn luyện kỹ năng tổ chức mã nguồn, phân chia module hợp lý.

- Hướng phát triển:

- Thêm nhiều người chơi, chế độ chơi với máy hoặc nhiều người.

- Lưu lịch sử ra file, phân tích thống kê kết quả.

- Cải thiện giao diện, bổ sung hiệu ứng động, âm thanh.

**KẾT LUẬN**

Qua quá trình nghiên cứu, thiết kế và xây dựng chương trình Blackjack với giao diện đồ họa (GUI), nhóm/tôi đã hoàn thiện một sản phẩm đáp ứng đầy đủ các yêu cầu của đề bài. Chương trình cho phép người dùng trải nghiệm trò chơi Blackjack một cách trực quan, sinh động với các chức năng như đặt cược, rút bài, dừng, hiển thị lá bài bằng hình ảnh, thống kê kết quả, lưu lịch sử ván chơi và hướng dẫn luật chơi.

Bài học kinh nghiệm:

- Củng cố và nâng cao kiến thức về lập trình hướng đối tượng, quản lý dữ liệu với list, dict trong Python.

- Thành thạo hơn trong việc xây dựng giao diện người dùng với Tkinter, xử lý ảnh với Pillow.

- Hiểu rõ hơn về cách tổ chức, phân chia module, quản lý trạng thái và sự kiện trong ứng dụng GUI.

- Rèn luyện kỹ năng giải quyết vấn đề, kiểm thử và hoàn thiện sản phẩm.

## LỜI CẢM ƠN

Em xin chân thành cảm ơn **thầy Nguyễn Văn Huy** đã tận dụng tình giảng dạy và hướng dẫn trong suốt quá trình học tập và thực hiện đề tài **“Xây dựng trò chơi Blackjack với giao diện GUI”** . Những kiến ​​thức về lập trình hướng dẫn đối tượng, xử lý thuật toán và thiết kế đồ họa giao diện mà thầy/cô truyền đạt đã giúp em hoàn thành bài báo báo cáo này một cách hiệu quả và đầy đủ.

Em cũng xin gửi lời cảm ơn đến **các bạn cùng lớp** đã trao đổi, chia sẻ kinh nghiệm và hỗ trợ em trong suốt quá trình làm bài. Để giúp đỡ và góp ý quý báu, em đã có thêm động lực và định hướng để hoàn thiện đề tài một cách tốt nhất.

Mặc dù đã nỗ lực hết mình, nhưng bài báo cáo không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và lời khuyên từ thầy/cô để em có thể rút kinh nghiệm và hoàn thiện hơn trong những lần thực hiện sau.

**Xin chân thành cảm ơn!**