# Foundations and Application of Generative AI

## Final Report

Interactive Laptop Recommendation Chatbot

**Team Members:** Lingwei Lu, Kairui Zhang, Haojun Liang, XiaoYao Wang

**Supervisor:** Chunyang Chen

**Tutors:** Shen Hu, Yuetian Mao, Ludwig Felder

Technical University of Munich,
School of Management / School of Computation, Information and Technology

Heilbronn, Germany
February 17, 2025

# Contents

1 User Guide ........................................................ 1

1.1 Overview ........................................................ 1

1.2 User Login Interface ........................................................ 1

1.3 User Register Interface ........................................................ 2

1.4 Interactive Interface ........................................................ 2

2 Project Management Report ........................................................ 4

2.1 Project Vision ........................................................ 4

2.2 Project Timeline ........................................................ 5

2.3 System Architecture ........................................................ 5

2.4 Project Methodologies ........................................................ 6

2.5 Team Chart ........................................................ 9

2.6 GitHub Link ........................................................ 9

2.7 Current Progress and Future Plans ........................................................ 9

2.8 Image Recognition Accuracy Improvement Experiment ........................................................ 10

3 User Acceptance Testing ........................................................ 12

3.1 User Case Study ........................................................ 12

3.2 User Acceptance Test Plan ........................................................ 13

3.3 Summary of the execution of the UAT ........................................................ 14

4 Project Safety and Reflection Report ........................................................ 15

4.1 Safety of GenAI ........................................................ 15

4.2 Lessons Learned and Reflections ........................................................ 16

Acknowledgments of GenAI Usage ........................................................ 18

# 1 User Guide

## 1.1 Overview

The Interactive Laptop Recommendation Chatbot is designed to assist you in selecting the best laptop based on their preferences and needs. This chatbot provides a seamless interactive experience through three user interfaces, each offering distinct functionalities.

- **User Login Interface**
- **User Register Interface**
- **Interactive Interface**

The following sections will guide you through how to use each interface effectively.

## 1.2 User Login Interface

The User Login Interface allows user to enter their credentials to access the system securely.

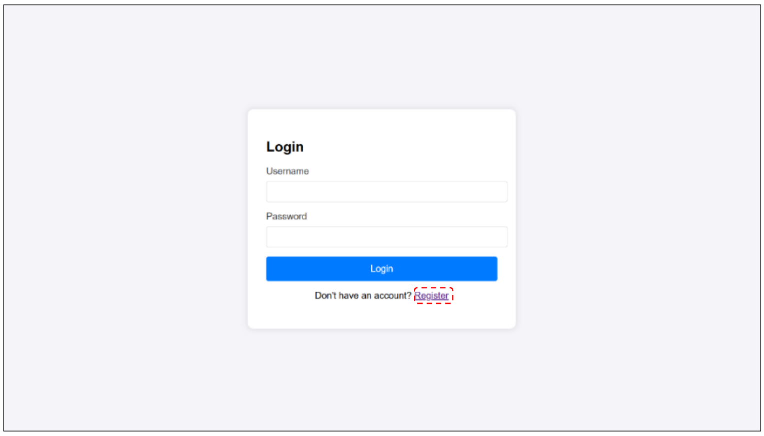
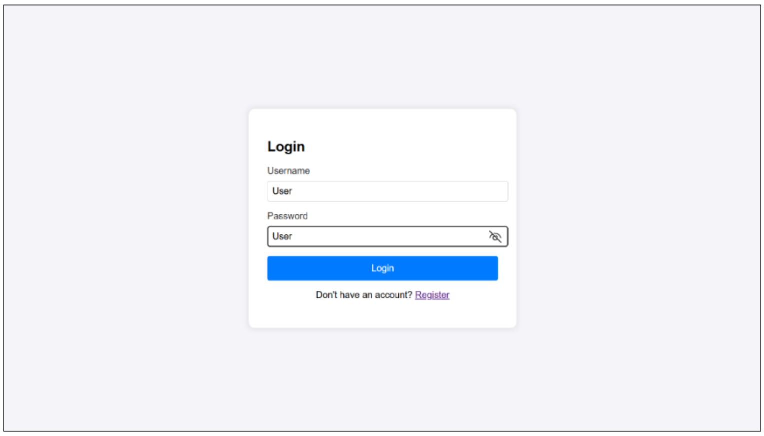


Figure 1: User Login Interface (1/2)

When the you enter the chatbot website, you first go to the login interface. If you don't have an account yet, click the “Register” button in the lower right corner (Figure 1). Then you will be redirected to the register interface to sign up for an account.



Username: User

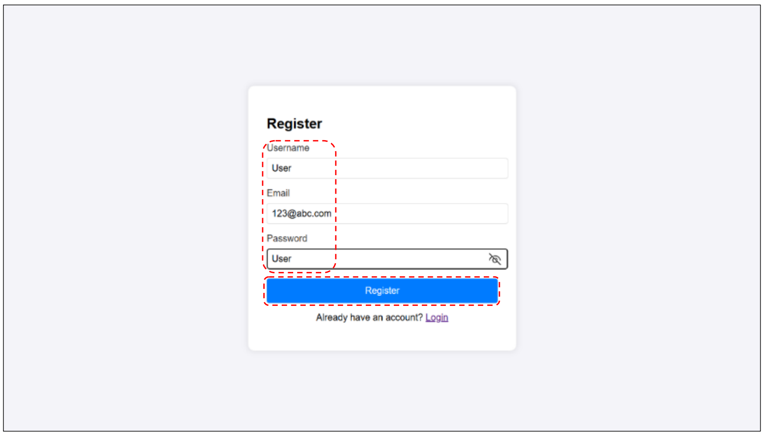
Password: User

Figure 2: User Login Interface (2/2)

If you already have an account, then you enter your username and password in the corresponding positions, and then click “Login” to enter the interactive interface (Figure 2).

## 1.3 User Register Interface

The User Register Interface enables new user to create an account by providing necessary details.



Register

Username

User

Email

123@abc.com

Password

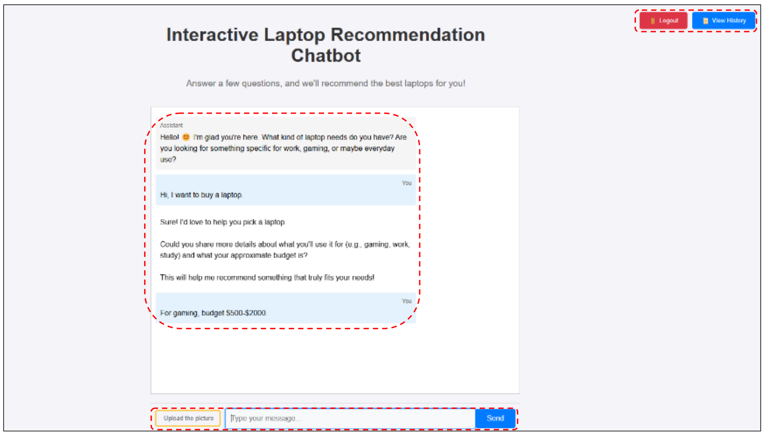
User

Figure 3: User Register Interface

On the register interface, you can fill in your desired username, your email, and a password. After completing the above, click “Register” to complete the process (Figure 3). The uniqueness of the account is determined by the username.

## 1.4 Interactive Interface

The Interactive Interface provides a chatbot-driven experience to assist user in selecting the best laptop based on their preferences.



Interactive Laptop Recommendation Chatbot

Answer a few questions, and we'll recommend the best laptops for you!

Assistant

Hello! 😊 I'm glad you're here. What kind of laptop needs do you have? Are you looking for something specific for work, gaming, or maybe everyday use?

You

Hi, I want to buy a laptop.

Sure! I'd love to help you pick a laptop.

Could you share more details about what you'll use it for (e.g., gaming, work, study) and what your approximate budget is?

This will help me recommend something that truly fits your needs!

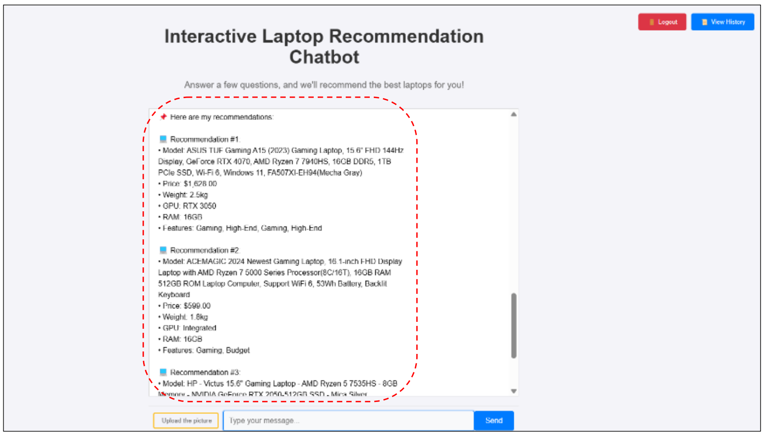
You

For gaming, budget $500-$2000.

Figure 4: Interactive Interface (1/4)

In this interface, the content of your interactions with the Chatbot is displayed in a chat box positioned at the center of the screen, allowing you to view both current and past conversations at any time. To communicate with the Chatbot, simply type your message in the text input box located at the bottom of the interface and click the “Send” button to submit your query. In the upper right corner of the interface, you will find two shortcut buttons, “Logout” and “View History”. Clicking “Logout” will securely log you out of your account, while selecting “View History” enables you to access previous interactions with the Chatbot for reference (Figure 4).

Once you have indicated your intention to seek advice and provided information about your needs, the Chatbot provides you with personalized purchasing advice by analyzing your preferences and matching them with relevant information from the database (Figure 5).



Interactive Laptop Recommendation Chatbot

Answer a few questions, and we'll recommend the best laptops for you!

📌 Here are my recommendations:

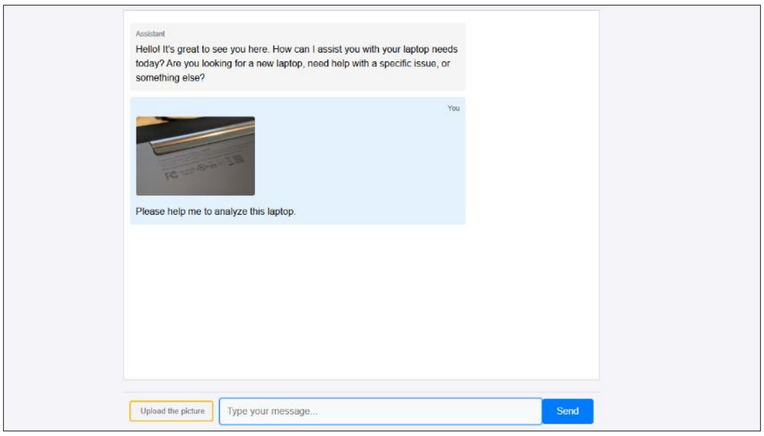
🟦 Recommendation #1:
• Model: ASUS TUF Gaming A15 (2023) Gaming Laptop, 15.6" FHD 144Hz Display, GeForce RTX 4070, AMD Ryzen 7 7940HS, 16GB DDR5, 1TB PCIe SSD, Wi-Fi 6, Windows 11, FA507XI-EH94(Mecha Gray)
• Price: $1,628.00
• Weight: 2.5kg
• GPU: RTX 3050
• RAM: 16GB
• Features: Gaming, High-End, Gaming, High-End

🟦 Recommendation #2:
• Model: ACEMAGIC 2024 Newest Gaming Laptop, 16.1-inch FHD Display Laptop with AMD Ryzen 7 5000 Series Processor(8C/16T), 16GB RAM 512GB ROM Laptop Computer, Support WiFi 6, 53Wh Battery, Backlit Keyboard
• Price: $599.00
• Weight: 1.8kg
• GPU: Integrated
• RAM: 16GB
• Features: Gaming, Budget

🟦 Recommendation #3:
• Model: HP - Victus 15.6" Gaming Laptop - AMD Ryzen 5 7535HS - 8GB Memory - NVIDIA GeForce RTX 2050-512GB SSD - Mica Silver

Figure 5: Interactive Interface (2/4)

In addition to sending text messages, you can also upload images of their laptops for analysis (Figure 6). The Chatbot processes the uploaded images along with your queries to provide relevant insights and recommendations (Figure 7). By combining visual and textual inputs, the Chatbot enhances its analysis, ensuring more accurate and tailored responses.



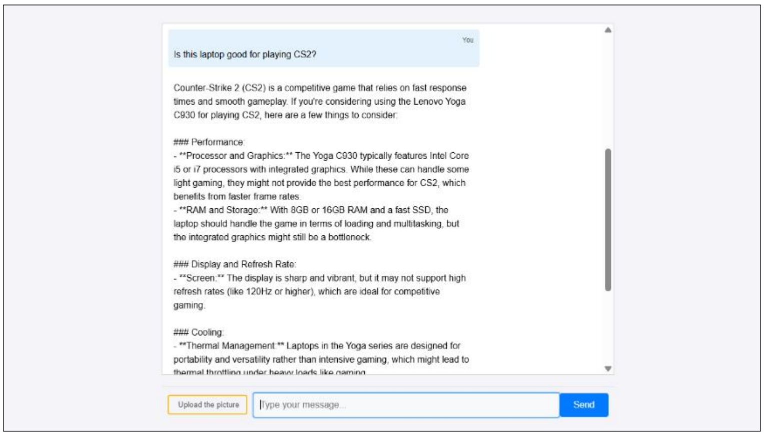
Assistant

Hello! It's great to see you here. How can I assist you with your laptop needs today? Are you looking for a new laptop, need help with a specific issue, or something else?

You

Please help me to analyze this laptop.

Figure 6: Interactive Interface (3/4)



You

Is this laptop good for playing CS2?

```
Counter-Strike 2 (CS2) is a competitive game that relies on fast response times and smooth gameplay. If you're considering using the Lenovo Yoga C930 for playing CS2, here are a few things to consider:

### Performance:
- **Processor and Graphics:** The Yoga C930 typically features Intel Core i5 or i7 processors with integrated graphics. While these can handle some light gaming, they might not provide the best performance for CS2, which benefits from faster frame rates.
- **RAM and Storage:** With 8GB or 16GB RAM and a fast SSD, the laptop should handle the game in terms of loading and multitasking, but the integrated graphics might still be a bottleneck.

### Display and Refresh Rate:
- **Screen:** The display is sharp and vibrant, but it may not support high refresh rates (like 120Hz or higher), which are ideal for competitive gaming.

### Cooling:
- **Thermal Management:** Laptops in the Yoga series are designed for portability and versatility rather than intensive gaming, which might lead to thermal throttling under heavy loads like gaming
```

Figure 7: Interactive Interface (4/4)

# 2 Project Management Report

This report outlines the planning, management, and execution of the Interactive Laptop Recommendation Chatbot project. It provides a comprehensive overview of how our team structured and developed the chatbot, ensuring an efficient and user-friendly experience.

The report also details the project vision, timeline, system architecture, methodologies, and team contributions, offering insights into our strategic approach. Additionally, we discuss our prompt engineering techniques to optimize chatbot performance, along with the current progress and future plans for the project.

## 2.1 Project Vision

Selecting a laptop can be a complex and overwhelming task due to:

- **Numerous available choices** - Making comparison difficult.
- **Complex specifications** - Challenging for non-experts to understand.
- **Time-consuming research** - Requiring users to browse multiple sources for information.

Therefore, our project aims to simplify this process by offering an interactive AI-driven chatbot that guides users toward informed purchasing decisions.

### 2.1.1 Project Boundaries

To ensure feasibility, the project scope is defined with the following constraints:

- The chatbot will focus exclusively on laptop recommendations, excluding other electronic devices such as smartphones or desktop PCs.
- The chatbot will provide recommendations based on existing databases from the market and GPT networking searches, but will not directly sell or manufacture laptops.

### 2.1.2 Project Solution

Our chatbot will enhance the laptop shopping experience through:

- **User interaction** - Engaging with users to understand their specific needs.
- **Needs assessment** - Analyzing user preferences and budget constraints.
- **Tailored recommendations** - Providing data-driven laptop suggestions based on objective criteria.

### 2.1.3 Expected Impact

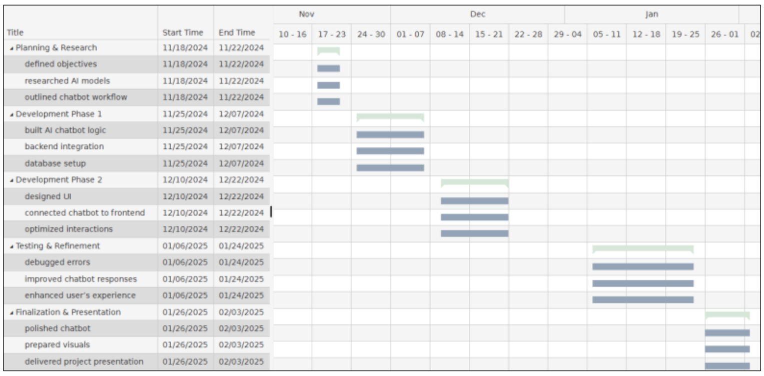
By integrating AI-driven decision-making and personalization, the chatbot will:

- **Increase Efficiency** - Streamlining the research and selection process, saving users time.
- **Enhance Clarity** - Simplify complex laptop specifications and reduce confusion.
- **Ensure Objectivity** - Providing unbiased, data-driven recommendations for better decision-making.

Through this project, we aim to create a smarter, more efficient, and user-friendly approach to laptop selection, ensuring a seamless experience for consumers.

## 2.2 Project Timeline

The following Gantt chart illustrates the project timeline, highlighting key milestones and progress.



| Title | Start Time | End Time |
|---|---|---|
| Planning & Research | 11/18/2024 | 11/22/2024 |
| defined objectives | 11/18/2024 | 11/22/2024 |
| researched AI models | 11/18/2024 | 11/22/2024 |
| outlined chatbot workflow | 11/18/2024 | 11/22/2024 |
| Development Phase 1 | 11/25/2024 | 12/07/2024 |
| built AI chatbot logic | 11/25/2024 | 12/07/2024 |
| backend integration | 11/25/2024 | 12/07/2024 |
| database setup | 11/25/2024 | 12/07/2024 |
| Development Phase 2 | 12/10/2024 | 12/22/2024 |
| designed UI | 12/10/2024 | 12/22/2024 |
| connected chatbot to frontend | 12/10/2024 | 12/22/2024 |
| optimized interactions | 12/10/2024 | 12/22/2024 |
| Testing & Refinement | 01/06/2025 | 01/24/2025 |
| debugged errors | 01/06/2025 | 01/24/2025 |
| improved chatbot responses | 01/06/2025 | 01/24/2025 |
| enhanced user's experience | 01/06/2025 | 01/24/2025 |
| Finalization & Presentation | 01/26/2025 | 02/03/2025 |
| polished chatbot | 01/26/2025 | 02/03/2025 |
| prepared visuals | 01/26/2025 | 02/03/2025 |
| delivered project presentation | 01/26/2025 | 02/03/2025 |

Figure 8: Project Gantt Chart

## 2.3 System Architecture

### 2.3.1 Overview

The Interactive Laptop Recommendation Chatbot is a Flask-based AI system that integrates OpenAI's GPT-4 API with a structured SQL-driven recommendation engine. It provides users with personalized laptop recommendations, supports image-based model analysis, and ensures secure, guided interactions through advanced filtering mechanisms. Recent enhancements include purchase link retrieval, an improved fallback recommendation system, and stricter security compliance.

### 2.3.2 User Interaction Layer

- **Frontend**

  The system serves an HTML-based UI via Flask routes. The primary user interface supports text and image input, allowing users to request laptop recommendations.

- **API Endpoints**
  - ➢ `/chat`: Handles user chat interactions.
  - ➢ `/start'`: Initializes a new conversation session.
  - ➢ `/register`, `/login`: Manages user authentication.
  - ➢ `/health`: Monitors system status and database connectivity.

### 2.3.3 Processing Layer

- **Intent Recognition & Input Processing**
  - ➢ The `parse_user_input()` function analyzes user queries to determine whether they are requesting a laptop recommendation or an analysis.
  - ➢ The `analyze_user_intent()` function categorizes user inputs into different intents, such as "recommendation" or "analysis".
- **Laptop Recommendation Mechanism**
  - ➢ The system attempts database-driven recommendations first.
  - ➢ If no database matches are found, GPT-4 generates alternative recommendations.

- **Visual Recognition**

The system supports image-based laptop analysis using OpenAI's GPT-4 Vision model. It extracts visible brand, model, and hardware details and matches specifications against the existing database for improved accuracy.

- **GPT-4 Integration**
  - The `call_gpt()` function interfaces with OpenAI's GPT-4 API to generate responses.
  - The system filters AI-generated responses to ensure compliance with its security constraints.

### 2.3.4 Database Layer

The system uses an SQLite database (`laptop_recommendations_3.db`) to store and retrieve laptop recommendations.

Tables include:

- `laptops`: Contains model specifications, pricing, GPU, RAM, and product tags for recommendation filtering.
- `chat_history`: Stores user conversation logs for improving chatbot interactions.

### 2.3.5 Session Management & User Authentication

Flask sessions store user states, including conversation history and past recommendations. As for User Authentication, the system allows users to register, log in, and log out, ensuring persistent recommendation tracking.

### 2.3.6 System Security

Enhanced `SecurityFilter` blocks queries related to "`bypass`", "`exploit`", "`malware`".

Context-Based Security Checker (`ContextSecurityChecker`) monitors multi-step security bypass attempts and rejects responses that could lead to adversarial AI manipulation.

Post-Processing Response Validation (`validate_user_input()`) scans GPT-generated responses to prevent accidental security breaches.

## 2.4 Project Methodologies

The Interactive Laptop Recommendation Chatbot applies a combination of Natural Language Processing (NLP), structured rule-based filtering, AI-driven feature matching, and deep learning-based image recognition to provide accurate, personalized, and secure laptop recommendations. The methodologies focus on ensuring efficient data retrieval, dynamic response generation, and AI safety compliance.

### 2.4.1 Natural Language Processing (NLP) Techniques

To effectively interpret user queries, the system leverages NLP techniques for intent recognition, keyword extraction, and security filtering:

- **Intent Recognition**

The function `analyse_user_input` categorizes user goals, distinguishing between:

- ➢ Laptop recommendations
- ➢ Technical analysis
- ➢ General inquiries

This enables the chatbot to prioritize structured data retrieval before invoking AI-based recommendations.

- **Keyword Extraction**

The chatbot detects user preferences such as:

- ➢ “`gaming`” → High-performance recommendations
- ➢ “`budget under $1000`” → Cost-conscious selection
- ➢ “`lightweight`” → Portable ultrabooks

Extracted keywords are mapped to structured filters to refine search results efficiently.

- **Context-Aware Filtering**

`ContextSecurityChecker` monitors recent user interactions to prevent:

- ➢ Repeated attempts to bypass security (e.g., step-by-step jailbreak tactics).
- ➢ Role confusion exploits (e.g., asking the AI to behave as both a “salesperson” and “hacker”).

This ensures chatbot responses remain consistent, ethical, and compliant.

### 2.4.2 Rule-Based Filtering (Structured Query Model)

Before invoking AI, the system applies structured query filtering to retrieve laptops from the database efficiently.

- **Database Query Execution** (`laptop_recommendations_3.db`)

The system queries pre-stored laptop models using structured SQL filtering:

- ➢ Budget Constraint: (`price <= user_budget`)
- ➢ Purpose-Based Filtering: (`tags LIKE '%Gaming%'`)
- ➢ Brand Preference: (`model LIKE '%Dell%'`)
- ➢ Portability Constraints: (`weight <= 2.5kg`)

- **Ranking and Selection**

Results are ranked based on relevance, prioritizing:

- ➢ Performance metrics (GPU, RAM, battery life).
- ➢ Popularity and availability (minimizing outdated recommendations).
- ➢ If multiple laptops meet the same criteria, the system introduces randomization to provide diverse options.

- **Purchase Link Retrieval**
  - ➢ Users can select a laptop (1, 2, or 3), and the chatbot fetches the corresponding purchase link.
  - ➢ If a link is unavailable, the system notifies the user, ensuring clear feedback.
- **AI-Based Fallback System**

- If no laptops meet the structured criteria, the chatbot invokes GPT-4 to suggest alternatives.
- This ensures that users always receive a meaningful recommendation, even for niche requirements.

### 2.4.3 GPT-4 & Feature Matching

When structured database filtering fails, the chatbot dynamically generates recommendations using GPT-4's advanced reasoning capabilities.

- **Extracting Key User Preferences**
  - The system refines budget, purpose, portability, and brand constraints.
  - Extracted data guides AI-generated insights.
- **AI-Driven Contextual Recommendations**

GPT-4:

- Compares models based on user-stated criteria.
- Fills in missing details when database information is incomplete.
- Ensures all suggestions remain safe, preventing security loopholes.

- **AI Safety Compliance**
  - GPT-generated responses are validated before display.
  - Security filtering removes potential exploits (e.g., "bypass restrictions" or "hack into a laptop").

### 2.4.4 Image Recognition & Deep Learning (GPT-4 Vision)

The chatbot integrates deep learning-based image analysis to enable laptop recognition from uploaded images.

- **Image Processing Workflow**

Users upload a photo of a laptop. The system encodes the image into Base64 format for API processing. OpenAI's GPT-4 Vision model extracts:

- Visible brand and model details (e.g., Dell XPS, MacBook Air)
- Screen size, ports, keyboard design
- Additional physical attributes for comparison

- **Matching Against Database Records**
  - Extracted specifications are compared with existing laptop data.
  - If an exact match is found, the chatbot retrieves product information and purchase links.
  - If no match is found, GPT-4 suggests the closest available alternative.
- **Enhancing User Experience**
  - Allows laptop recognition without manual input, making the selection process more intuitive.
  - Helps users verify specifications before purchasing.

### 2.4.5 AI Security & Adversarial Query Protection

To prevent jailbreak attempts and adversarial inputs, the chatbot incorporates multi-layered security enforcement.

- **Keyword-Based Filtering** (`SecurityFilter`)
  - Preemptively blocks queries containing: “`hack`”, “`exploit`”, “`bypass`”, “`malware`”.
  - Ensures GPT-4 does not respond to unauthorized security inquiries.
- **Sequential Context Monitoring** (`ContextSecurityChecker`)

Analyzes the last five messages to:

  - Detect multi-step security evasion attempts.
  - Prevent GPT from being manipulated into revealing restricted information.
- **AI Response Validation** (`validate_user_input()`)
  - Every GPT-generated response is analyzed before display.
  - If an unsafe response is detected, it is blocked or reworded.
- **Contextual Security**
  - Prevents users from incrementally asking questions to bypass initial filtering.
  - Ensures AI maintains consistent ethical standards.

### 2.4.6 User Session Management

- **Authentication & Secure Data Handling**
  - Implements Flask-based session tracking.
  - Uses bcrypt password hashing for secure user authentication.
  - Maintains user privacy by preventing unauthorized data access.

## 2.5 Team Chart

The following chart outlines the team members, their roles, and their specific contributions to the project.

Table 1: Team Members and Contributions

| ***Team Members*** | ***Specific Roles and Contributions*** |
|---|---|
| Lingwei Lu | Team Lead<br>Full-Stack Developer<br>Project Management & Requirements |
| Kairui Zhang | AI Specialist<br>Backend Developer<br>Presentation Strategist |
| Haojun Liang | Data Retriever<br>Backend Developer |
| XiaoYao Wang | Debugging & Testing<br>Frontend Development |

## 2.6 GitHub Link

https://github.com/HAOJUN-LIANG/Interactive-Laptop-Recommendation-Chatbot

## 2.7 Current Progress and Future Plans

### 2.7.1 Current Progress

The development of the Interactive Laptop Recommendation Chatbot has achieved significant milestones. The key components that have been implemented include:

- **User Authentication System** - A secure authentication mechanism to ensure personalized recommendations based on user preferences.

- **Chat-based Interaction System** - An AI-driven chatbot that enables natural language interactions for seamless communication.
- **Laptop Image Recognition Analysis** - Based on the functionality of GPT image recognition, chatbot is able to recognize the images uploaded by users and provide relevant analysis in conjunction with the descriptions.

These foundational components establish the core functionality of the chatbot, enabling user engagement, secure access, and intelligent data analysis.

### 2.7.2 Future Plan

To further enhance the chatbot’s effectiveness and user experience, several improvements are planned:

- **Feature Enhancements**
  - ➢ **Advanced filtering options** - Allow users to provide more preferences to refine search results.
  - ➢ **Price history tracking** - Provide users with historical price trends to support informed purchasing decisions.
  - ➢ **User reviews integration** - Incorporate real user feedback from multiple sources to enhance recommendation reliability.
  - ➢ **Export recommendations** - Enable users to save or share recommended laptops for future reference.
- **User Experience Updates**
  - ➢ **More interactive elements** - Design better UI to make user interaction more intuitive.
  - ➢ **Enhanced mobile experience** - Optimize chatbot for mobile platforms to ensure seamless usability across devices.
  - ➢ **Social sharing features** - Allow users to share laptop recommendations with peers, fostering a more collaborative decision-making process.
- **Technical Improvements**
  - ➢ **Integration of e-commerce platform APIs** - Connect with online selling platforms to provide real-time pricing, availability, and product details.
  - ➢ **Response optimization** - Enhance the chatbot’s processing speed and accuracy for a smoother user experience.
  - ➢ **Additional visualization options** - Introduce multiple visual displays to support intuitive decision making.

By implementing these improvements, the project aims to deliver a highly efficient, user-friendly, and intelligent laptop recommendation system, ensuring a seamless shopping experience for users.

## 2.8 Image Recognition Accuracy Improvement Experiment

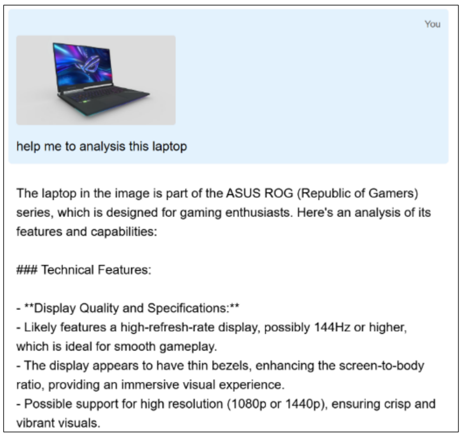
This experiment aims to evaluate the improvement in image recognition accuracy after refining the prompt used in the model. The experiment compares the model's performance before and after the prompt modification to determine its effectiveness in identifying laptop models more accurately and providing more professional analyses.

### Test Data and Environment

- **Test Dataset:** 20 images of different laptop models, including various angles and lighting conditions (Laptop front, back, side and other angles.)
- **Testing Environment:** 1. Model: GPT-4o Vision API. 2. Hardware: 32GB RAM, Intel i5-13500H 3. Software: Python 3.12.7, OpenAI API.
- **Evaluation Metric:** Accuracy in correctly identifying laptop model and detail depth in analysis.

### Experiment Setup (For function: `def analyze_laptop_image`)

- **Baseline Test (Original Prompt):** The model was given a basic prompt to recognize laptop models from the dataset. Results were recorded, focusing on identification accuracy and analysis depth.
- **Improved Test (Refined Prompt):** The optimized prompt was introduced, guiding the model to focus on distinguishing laptop models more accurately and providing technical specifications. Results were compared against the baseline test. Providing professional analysis from the perspectives of Technical Features, User Experience, and Recommendations.



help me to analysis this laptop

```
The laptop in the image is part of the ASUS ROG (Republic of Gamers)
series, which is designed for gaming enthusiasts. Here's an analysis of its
features and capabilities:

### Technical Features:

- **Display Quality and Specifications:**
- Likely features a high-refresh-rate display, possibly 144Hz or higher,
which is ideal for smooth gameplay.
- The display appears to have thin bezels, enhancing the screen-to-body
ratio, providing an immersive visual experience.
- Possible support for high resolution (1080p or 1440p), ensuring crisp and
vibrant visuals.
```

Figure 9: Part of the Analysis Result for Refined Prompt

### Results and Comparison

Table 2: Results and Comparison

| Metrics | Baseline Prompt | Refined Prompt | Improvement |
|---|---|---|---|
| Accuracy | 70% | 85% | +15% |
| Processing Time | 6.8s | 9.4s | +2.6s |
| Misidentifications | 6/20 | 3/20 | -3 |

### Conclusion

The refined prompt significantly improved the model's accuracy in identifying laptop models, increasing accuracy by 15% and reducing misidentifications by 50%. However, the improvement came at the cost of an increased processing time (+2.6s), indicating a trade-off between speed and precision.

# 3 User Acceptance Testing

## 3.1 User Case Study

### 3.1.1 User Background Information

The user A chosen for this UAT is a master student major in management at TUM. This user represents an important demographic for our laptop recommendation system, as they require a high-performance gaming laptop within a budget constraint. The user needs a system capable of running modern AAA games like Assassin's Creed while maintaining portability and staying within a $900 budget. The user has no brand preference and is open to any laptop that meets performance and display criteria.

### 3.1.2 User A Requirements

1. High performance to run modern AAA games like Assassin's Creed smoothly.
2. Dedicated GPU (at least GTX 1650 or higher) to ensure an optimal gaming experience.
3. Weight around 2kg for portability.
4. 16-inch display with 2K-4K resolution for high-quality visuals.
5. Budget ≤ $900, requiring the best balance between price and performance.
6. No brand preference, willing to consider any suitable laptop.

### 3.1.3 Processes, Pros, and Cons

- **User testing process**
  1. Users input requirements (budget, purpose, graphics card, weight, screen).
  2. The system recommends laptops and displays 2-3 models that meet the requirements, three laptops were recommended, all of which are within $900. Two laptops meet the GPU requirements (RTX 3050), and one has a slightly weaker GPU (GTX 1650). One model weighs more than 2kg and does not fully meet the portability requirements. In terms of screen, two models have a 2K resolution, and one model is only 1080p.
- **Pros**
  1. The user login system can well save the chat history between the customer and the chatbot.
  2. Intuitive UI: The chatbot allows users to seamlessly input preferences and receive results.
  3. Highly Relevant Recommendations: Most recommended laptops meet user needs for performance, screen size, and GPU.
- **Cons**
  1. Some Laptops Exceeded Weight Limit: A few recommendations exceeded 2kg, which does not meet portability requirements.
  2. Resolution Was Not Always Prioritized: Some recommendations did not strictly adhere to 2K-4K resolution criteria.
  3. The local database has a small number of products. When users put forward more detailed

requirements, the model cannot fully interpret the requirements put forward by users. As a result, the model cannot return products that are very suitable for users' specific needs, and the chatbot need to search online.

## 3.2 User Acceptance Test Plan

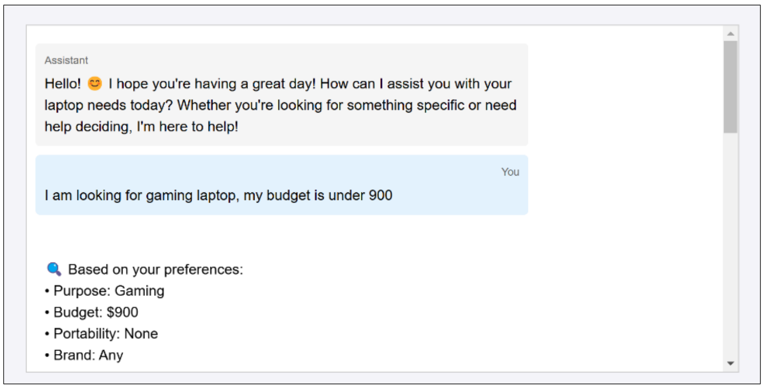
- **Processes for Testing**

➢ **User A Provides Requirements**

The user A interacts with the chatbot, which asks and records the following information:

(1) Purpose (2) Budget (3) Portability (4) Brand

The chatbot can perform preliminary retrieval based on these.



Assistant

Hello! 😊 I hope you're having a great day! How can I assist you with your laptop needs today? Whether you're looking for something specific or need help deciding, I'm here to help!

You

I am looking for gaming laptop, my budget is under 900

🔍 Based on your preferences:
• Purpose: Gaming
• Budget: $900
• Portability: None
• Brand: Any

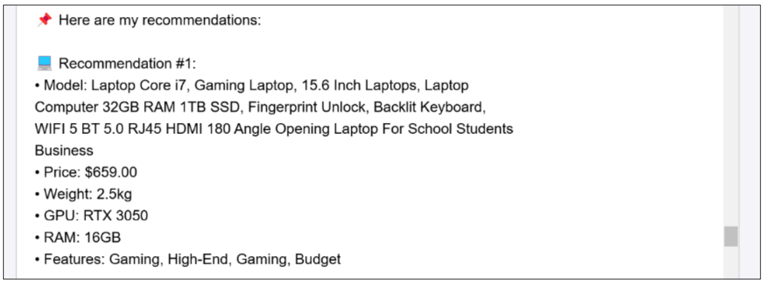
Figure 10: Interaction & Recommendation Generation

➢ **System Filtering Process**

1. Filters laptops under $900
2. Ensures weight does not exceed 2kg
3. Ensures the laptops are for gaming

➢ **Displaying Recommendations**

Shows 2-3 matching laptops with details on price, performance, GPU, RAM, storage, display quality, purchase links are included (if available).



📌 Here are my recommendations:

💻 Recommendation #1:
• Model: Laptop Core i7, Gaming Laptop, 15.6 Inch Laptops, Laptop Computer 32GB RAM 1TB SSD, Fingerprint Unlock, Backlit Keyboard, WIFI 5 BT 5.0 RJ45 HDMI 180 Angle Opening Laptop For School Students Business
• Price: $659.00
• Weight: 2.5kg
• GPU: RTX 3050
• RAM: 16GB
• Features: Gaming, High-End, Gaming, Budget

Figure 11: Displaying Recommendations

➢ **User Interaction**

If satisfied, the user can select a laptop from the recommendations.

If unsatisfied, the user can adjust budget, performance requirements, or request the chatbot search the Internet automatically and search for a computer suitable for the user according to the user's needs.

- **Overall Satisfaction from User A**

Table 3: Feedback from User A

| **Overall rating** | **4/5** |
|---|---|
| Output rating | 4/5 |
| Fluency rating | 5/5 |
| UI rating | 3/5 |
| Does the recommendation meet the budget? | Yes. |
| Does the laptop meet AAA gaming requirements? | Yes. |
| Are the GPU, weight, and display quality satisfactory? | Not all of them meet this. |
| Does the user want more options? | Yes. |

## 3.3 Summary of the execution of the UAT

- **Execution Overview of UAT**
  - **Testing Period**

    2025-02-06 to 2025-02-08
  - **Target User**

    Master student major in Management at TUM
- **Main Testing Objectives**
  1. Verify whether the laptop recommendation system meets the budget and gaming performance needs of the user.
  2. Evaluate user satisfaction with the recommendations.
  3. Collect feedback to optimize recommendation logic and improve user experience.
- **Main Positive Feedback**
  1. Good budget control, all recommended models were within $900.
  2. Accurate GPU selection, most recommended models met AAA gaming requirements (e.g., GTX 1650 / RTX 3050 or higher).
  3. Smooth chatbot interaction, easy-to-use and clear information.
- **Main Improvement Areas**
  1. Some recommended laptops exceeded 2kg, affecting portability.
  2. Some recommended models didn’t provide information about screen size, users needed this information.
  3. Lack of detailed recommendation explanations, users wanted more insights on why specific laptops were suggested.
- **Improvement Plan**
  1. Optimize algorithm to ensure all recommended models are ≤ 2kg.
  2. Extract laptop screen size information when searching the local database.
  3. Add detailed explanations for each recommendation.
  4. Allow the model to retrieve from the Internet instead of the local database when receiving complex user requirements, providing the user with the most suitable computer model.

# 4 Project Safety and Reflection Report

## 4.1 Safety of GenAI

- **Safety Considerations**

When integrating GenAI into our chatbot, we have also addressed ethical and technical risks through the following measures:

- **Data Privacy and Security**

➢ **Risk**

User interactions involve sensitive data (such as budget, preferences). Unauthorized access or data leaks could compromise privacy.

➢ **Technical Improvements**

1. End-to-End Encryption: Ensured secure communication between the user and the chatbot using HTTPS, such that cookies are only transferred via HTTPS and prevented client-side scripts accessing cookies.

```
# Configure session cookies to enhance security
app.config['SESSION_COOKIE_SECURE'] = True  # Ensure cookies are transmitted only via HTTPS
app.config['SESSION_COOKIE_HTTPONLY'] = True  # Prevent client-side scripts from accessing cookies
```

2. Secure Data Storage: Avoided storing personal data in plain text. Passwords were hashed using Werkzeug's security functions.

```
# Import functions for password hashing and verification
from werkzeug.security import generate_password_hash, check_password_hash
```

```
# Generate a hashed version of the given password
hashed_password = generate_password_hash(password)
```

3. Regular Security Audits: Implemented database encryption and API key rotation to mitigate vulnerabilities.

```
conn = sqlite3.connect('chat_history.db', uri=True)
conn.execute(f"PRAGMA key = {app.secret_key}")
```

- **Bias in Recommendations**

➢ **Risk**

GPT models may inherit biases from training data, leading to skewed recommendations.

➢ **Technical Improvements**

1. Diverse Data Sources: Curated laptop data from multiple platforms to minimize brand bias.

```
query = '''
SELECT model, price, weight, gpu, ram, tags, link
FROM laptops
WHERE price <= ?
AND (tags LIKE ? OR tags LIKE ?)
'''
```

2. Jailbreak prevention: Added prompts when initially calling GPT so that irrelevant responses are not generated from jailbreaking questions.

```
8. Strictly focus on providing laptop recommendations and related advice
9. Do not respond to requests that are unrelated to laptops or computer technology
10. If the user asks for unrelated content, politely decline and redirect the conversation to laptop recommendations
```

- **Misinformation Risks**

➢ **Risk**

GPT-generated responses may include inaccurate specifications.

➢ **Technical Improvements**

1. Local Database Validation: Cross-checked GPT's real-time search results with a verified local database.

```
query = '''
SELECT model, price, weight, gpu, ram, tags, link
FROM laptops
WHERE price <= ?
AND (tags LIKE ? OR tags LIKE ?)
'''

params = [state['budget'], f"%{state['purpose']}%", "%Gaming%"]

if state['portability'] == "Portable":
    query += " AND weight <= 2.5"

if state['brand']:
    query += " AND model LIKE ?"
    params.append(f"%{state['brand']}%")
```

2. User Feedback Loop: Allowed users to flag errors, which were logged and used to update the database.

```
@app.route('/report_error', methods=['POST'])
def report_error():
    user_id = session['user_id']
    error_details = request.json.get('error')
    log_error_to_db(user_id, error_details)
    return jsonify({"status": "success"})
```

## 4.2 Lessons Learned and Reflections

- **What Worked Well**

Dual Layer Recommendation System: Composed system of local database and fallback, then lastly GPT's real-time search provided more accurate and diverse results than pure GPT.

```
# Determine whether generate_recommendations_based_on_state fails to find results
# If it returns "I couldn't find any laptops" or similar messages, assume no results were found
if "couldn't find any laptops" in db_result.lower():
    # Retry fallback_recommendation_system
```

```python
    fallback_result = fallback_recommendation_system(user_message, conversation_history)

    # If fallback also fails
    if "I couldn't find" in fallback_result.lower() or "trouble accessing" in fallback_result.lower():
        # Finally, call GPT for a response
        return call_gpt(user_message, conversation_history)
    else:
        # If fallback has results, return them
        return fallback_result

else:
    # generate_recommendations_based_on_state succeeded, return the result
    return db_result
```

- **Areas for Improvement**

  1. More edge cases handling to prevent error or jailbreak
  2. Image recognition limitations due to GPT-4's vision API occasionally misidentified hardware components. Training a custom vision model could improve accuracy.

- **Discussion of GenAI Applications**

  ➢ **Strengths**

  During natural language procession, GenAI accomplished dynamic interaction and response generation at an unprecedented pace.

  ➢ **Weaknesses**

  An excessive dependency on GenAI without sufficient human involvement renders the system vulnerable to biases or inaccuracies being overstated.

- **Impact on Future Projects**

  ➢ **Collaborative Tool**

  Dynamic ideas and drafts are made possible through GenAI, but they need to be substantiated with appropriate other tools and lastly hard work and effort.

  ➢ **Ethical Guidelines**

  As with all AI based projects, they give the utmost importance to bias audits and transparency provisions, therefore more considerations will be taken into account.

  ➢ **Technical Validation**

  Use of GenAI together with well-constructed databases and APIs ensures that accuracy is attained, yet absolute accuracy is not possible, therefore validation should always be done.

- **Conclusion**

With the incorporation of technical restrictions and improving the system design, the safety, precision, and user experience of the chatbot has been improved. To achieve sustained success with the project, the next step would be to refine the performance in edge cases and the recognition of images.

# Acknowledgments of GenAI Usage

In the creation of this report, we utilized Generative AI (GenAI) in two key areas:

- **Grammar and Language Refinement**

  GenAI assisted in improving the clarity, grammar, and overall language quality of the document. It provided suggestions for rewording sentences, correcting grammatical errors, and enhancing the fluency of the writing.

- **Framework and Outline Construction**

  GenAI was employed to help create the general structure and outline of the report. It guided the organization of the content, ensuring logical flow and coherence across different sections.

- **Code Modification and Annotation**

  GenAI was used to assist with modifying and annotating certain sections of code, providing suggestions for improvements and clarifying the purpose of various code blocks.

These tools were used to enhance the overall quality of the report, while the substantive content and analysis remain the collective work of the team.