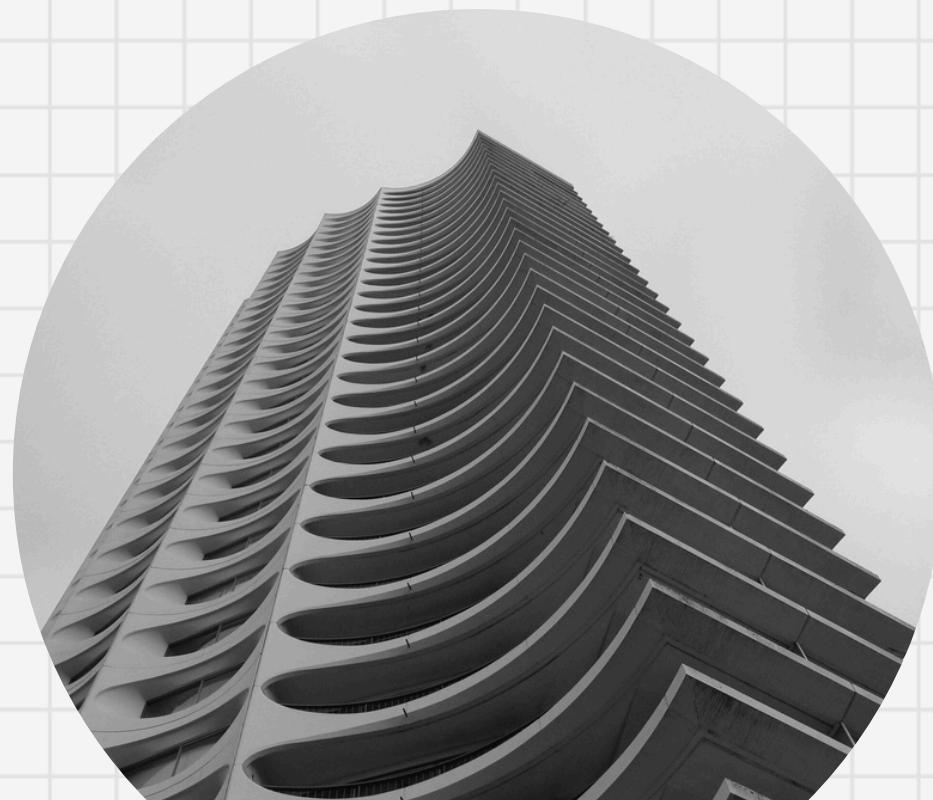


INDIVIDUAL PROJECT

ONLINE SHOPPING SITE WITH AI WEB CRAWLING SYSTEM

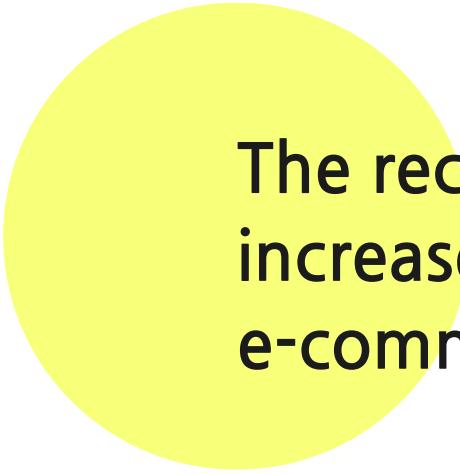
BY JEONGWOO KWAK



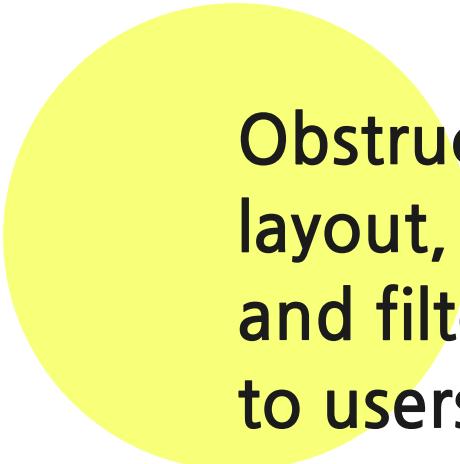
Why this Project?



Many online store managers still rely on manual methods for collecting product information.



The recent surge in hacking incidents has led to an increase in complaints from online shoppers about e-commerce platforms and reduced credibility.



Obstructive advertising, badly designed site layout, and uncomfortable category placement and filtering capabilities cause inconvenience to users.

Project aim

- Develop an AI web crawling system for online shopping site administrators to swiftly gather product information
- Focusing on improving the site's UI/UX for a better shopping experience.

Project objectives

Provide an AI-Based
Web Crawling
System

Simplify Product
Information
Collection

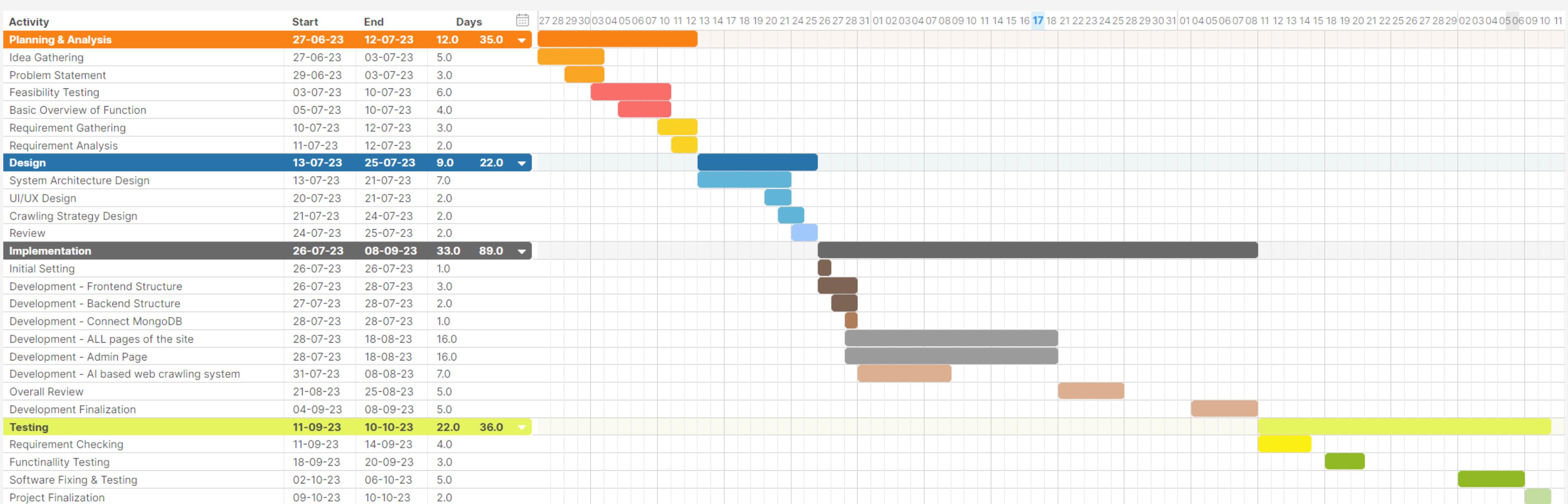
UI/UX Enhancement
for Buyers

Data Security and
Privacy using
MongoDB

Problem statement and Solutions

- Many websites including online shopping malls are hacked for lack of security
 - The manual collection of product information from the official website or various sales sites causes many problems
 - User convenience is reduced due to inconvenient UI/UX design elements on the website
 - Lack of providing Selling Points
- + Data protection Using MongoDB and powerful encryption
 - + Collecting automated product information using an AI-based web crawling system
 - + Improving user convenience by improving the uncomfortable UI/UX design on the Website
 - + Provides selling point sections for all product

Project Plan(Gantt chart)



Technical Literature Review

Development Tool:

- Visual Studio Code

Tools used in creating online shopping site:

- JavaScript
- Node JS & React JS
- MongoDB

JavaScript



Tools used in creating AI web crawling system:

- Python
- Beautiful Soup
- OpenAI API(ChatGPT)



Research Methodology

Project Methodology

To gather user feedback on the UI/UX of an online shopping site, I visited online blogs and communities where users discussed their experiences.

I collected diverse opinions and perspectives from these posts and also analyzed the comments on the posts to gain insights into how users perceive various UI/UX elements

Here is my concept UI for a Nike online store product page. Would you shop here or nah?

Sort by: New 44 comments

+ Add a Comment

cndfr • 3년 전
Do you know you would not be able to develop this. unless you have the png image of airmax with actually transparent sole?
+ 1 Reply Share ...

nivzk • 3년 전
Yes. there are many reasons why I won't be able to develop this, including but not restricted to:

- lack of images to see other angles of the product
- no shopping cart button
- no user reviews or rating
- no information on sizing
- no similar product suggestions
- no search bar to navigate the site
- lack of categories and nav bar items to browse the larger ecommerce store

I think getting a PNG sole shoe would not be an issue for a brand as creatively powerful as Nike in my fantasy design case.

What are the best e-commerce websites you have seen?

59 22 Share

Sort by: Best

22 comments

+ Add a Comment

soulby • 7 yr. ago

Moment Lens Co is one of my favourites. It's responsive, clean and shows examples of what photos would look like with and without their lenses.

But they do have the advantage of having a pretty small product range so there's no need for search bars or product filters.

Edit: if you're looking for an e-commerce site with a much larger product range then ASOS is great too.

15 Reply Share ...

+ 2 more replies

[deleted] • 7 yr. ago

McMaster by far, IMHO.

6 Reply Share ...

scotiewontusis • 7 yr. ago

I'm a big fan of the Warby Parker website. I've found it to be a really well-rounded UX design. I also like the Bill Blass website.

3 Reply Share ...

emiltsch • 7 yr. ago

www.Bellroy.com - great site, wonderful experience.

2 Reply Share ...

+ 2 more replies

orcfull • 7 yr. ago

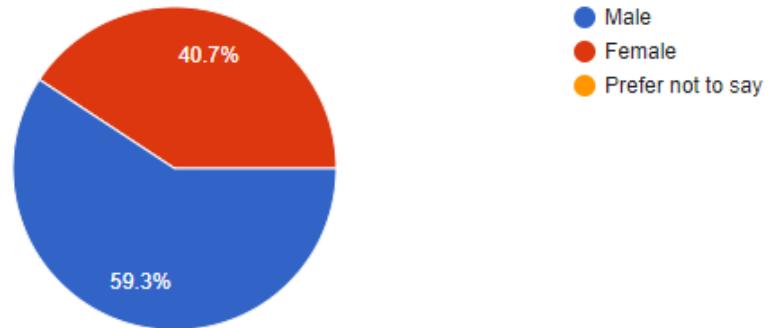
I fell in love with the Lush website a few years ago after interviewing with Method. The UK version of the site is amazing, each product has a background video and an ingredients trail linking to a backbone that shows all

Requirements Gathering: Questionnaire

- 27 Responses
- 15 Questions
- 3 Demographic Questions
- 12 UI/UX questions on online shopping sites

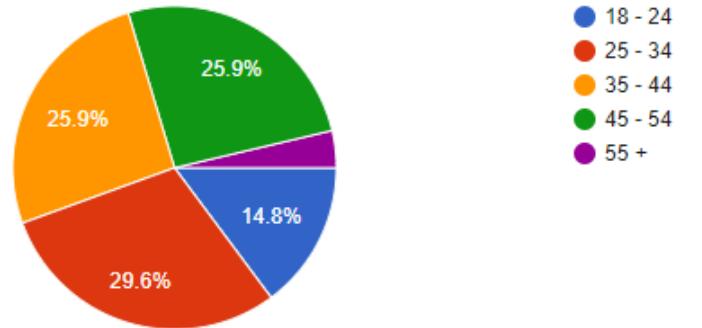
1. What is your gender?

27 responses



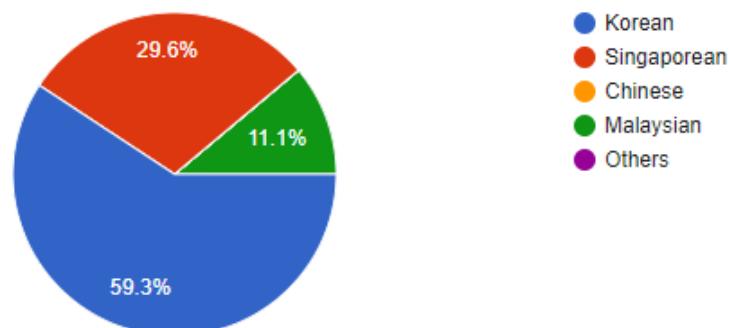
2. How old are you?

27 responses



3. What is your nationality?

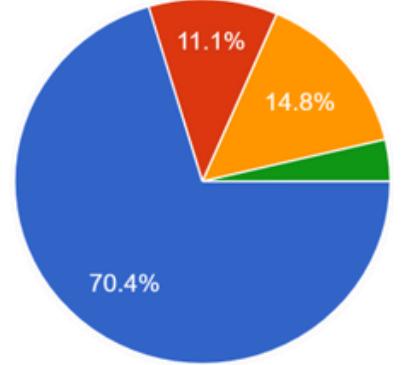
27 responses



Questionnaire

7. What is your preferred design for online shopping websites?

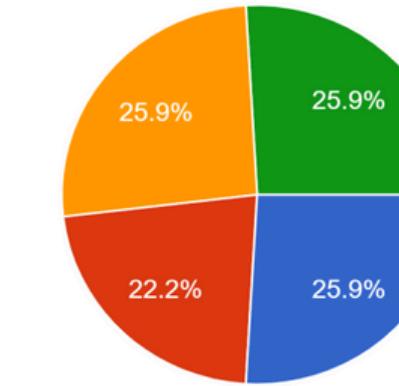
27 responses



- Clean and Minimalist Design
- Fresh and Iconic Design
- Modern and Flat Design
- Classic and Timeless Design
- Colorful and Vibrant Design

12. What information do you want the site to provide when you make a purchase decision?

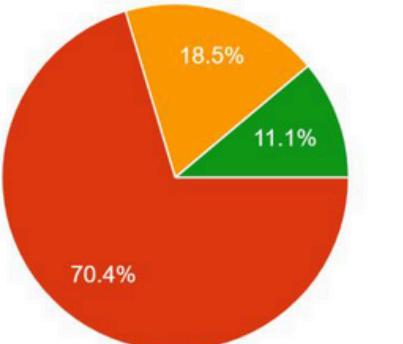
27 responses



- Detailed product descriptions
- High-quality product images
- Customer Reviews and Ratings
- Relevant product sales quantity

8. What type of color combinations do you prefer?

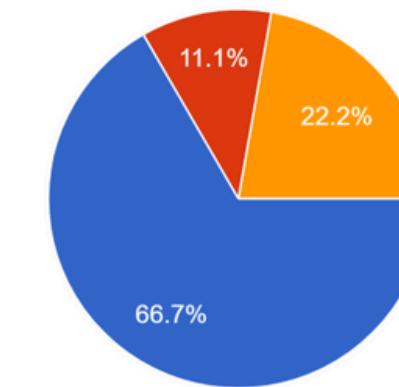
27 responses



- Monochromatic color theme
- High-contrast color combinations (e.g., black and white)
- Subtle variations in color (e.g., pastel tones)
- A mix of various colors

15. What would you like to improve for UI/UX convenience of online shopping mall sites?

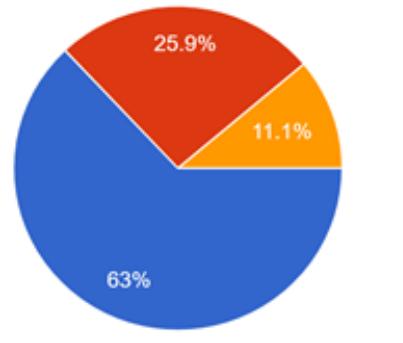
27 responses



- Remove unnecessary advertising pop-ups
- On the detail page, place letters in a way that is easy to recognizeable
- Search functions are located in the same place to make searching seamless
- Create large images and text

5. When you visit a shopping mall website, which design do you think should stand out the most on the first page?

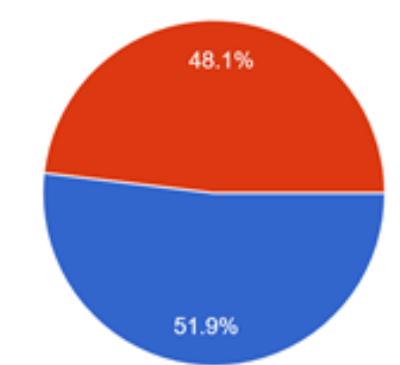
27 responses



- Product category list
- Popular product listings
- New Product listings
- Discount and promotion information

10. If you've ever experienced an ad pop-up that appeared out of nowhere on your favorite online shopping website, how would you rate it?

27 responses



- 1 (Very Poor)
- 2 (Poor)
- 3 (Average)
- 4 (Good)
- 5 (Excellent)

Online Shopping Site Requirements

Functional Requirements
Through API integration, the product listings, member information, and payment details should be connected to MongoDB.
If product listings or member information is deleted from MongoDB, it should also be reflected on the shopping website.
Product reviews and questions should be stored in the shopping site's own database.
Users should be able to write reviews for products on the review page.
Users should have the ability to add or remove products from their cart.

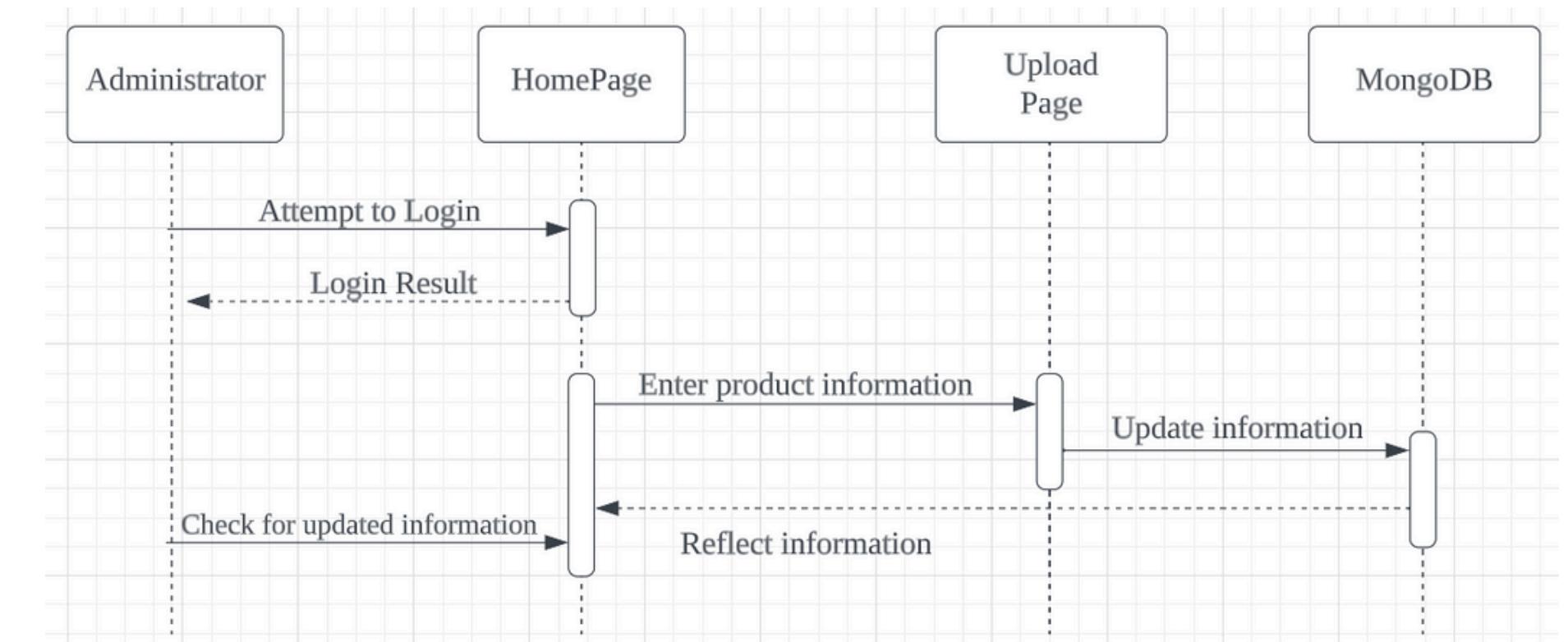
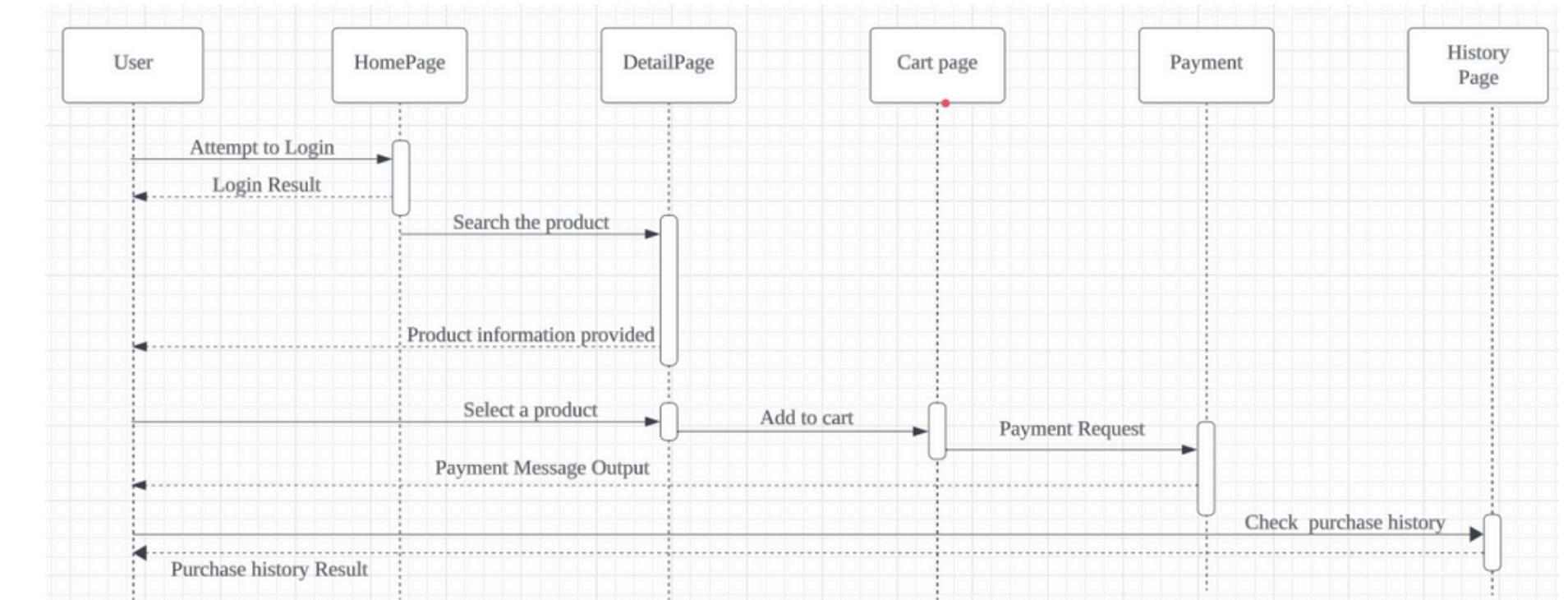
Non-Functional Requirements
The page layout is high-contrast color combinations (black and white).
When moving to each page, it takes only a few seconds to proceed.
The site encrypt and store user passwords using a security protocol.
The user interface should be intuitive and function smoothly on mobile devices as well.
In the event of a system error, a clear error message should be provided to the user.

AI Web Crawling System Requirements

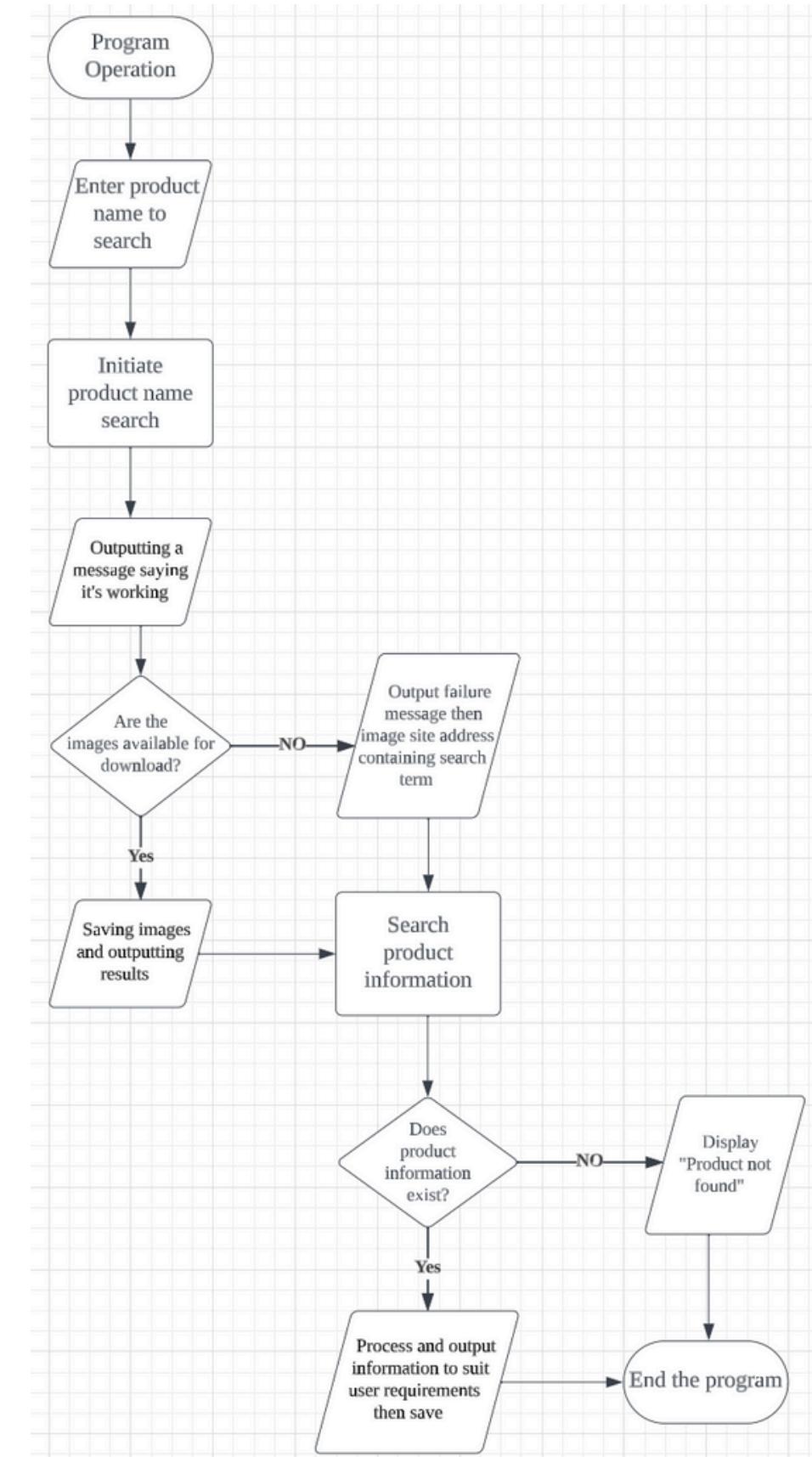
Functional Requirements
The system needs to retrieve HTML data from a given URL.
It should save the image from the collected HTML data.
The system needs to extract specific data from web pages.
The extracted information should be stored in a database in TXT format.
It should provide natural language processing capabilities for the extracted text data.

Non-Functional Requirements
In case of errors during the crawling process, the system should output error messages.
Web crawling and data processing tasks should be performed quickly.
The system should be able to run on various operating systems.
The system should consider modularity and scalability to easily add new features and modules.
For user convenience, an intuitive interface should be provided.

Use Case Diagram & Sequence Diagram for Online Shopping Site



Flow Chart for the AI Web Crawling System



Prototype : Wireframe for Online Shopping site

Page 1 https://www.default.com

AppleRun

APPLE sales

Search

AirPods iPhone iPad Mac Watch TV&Home Accessories

0 ~ 500SG\$ 501 ~ 1000SG\$ 1001 ~ 1500SG\$ 1501 ~ 2000SG\$ 2001SG\$ *

Product Name Product Price Product Name Product Price Product Name Product Price Product Name Product Price

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun

LOGIN

Email

Password

Login

New member? Register here

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun

REGISTER

Email

Name

Password

Register

Already member? LOGIN

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun

Product Name

Product Detail

Price: Sold: Selling Point: Description:

Add to Cart

Product Image

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun

My Cart

Product	Quantity	SG\$	Delete
Product Image	1 Quantity	118.56 SG\$	Delete

Total: 118.56SG\$

Buy Now

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun

My Cart

Cart is empty.

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Prototype: Wireframe for Online Shopping Site

Page 1 https://www.default.com

AppleRun Review QnA History Log out

History

Payment Id	Price	Quantity	Date of Purchase
1234abcd	2345	3	2023-07-26 02:42:27

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun Review QnA MongoDB Upload History Log out

Product Upload

+ Image Upload function

Name
Selling Point
Description
Price
Category

Register

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun Review QnA MongoDB Upload History Log out

Ratings & Reviews

Product Name: iPad
Detail: Good
★★★★☆
Date created: 2023. 9. 25. 3:14:02
Delete

Product Name: iPhone
Detail: Soso
★★☆☆☆
Date created: 2023. 10. 1. 9:14:02
Delete

Leave a review!

Make a review

Search Product Name

Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Page 1 https://www.default.com

AppleRun Review QnA MongoDB Upload History Log out

Questions and Answers

Question: Is iPad Good?
Submitted: 2023. 9. 25. 3:52:15
Answer: Yes
Delete

Question: Is iPhone Good?
Submitted: 2023. 10. 2. 5:32:22
Your answer: Submit Answer
Search questions...

Ask a new question:

Your question
Submit Question

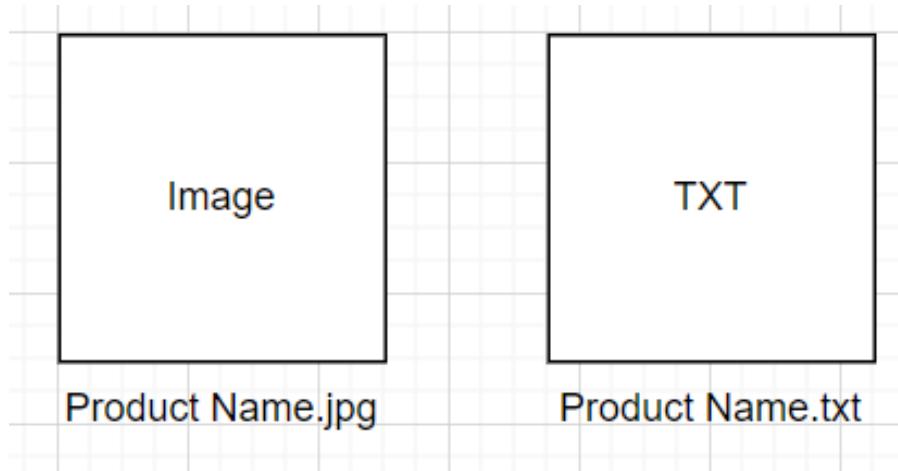
Contact us: Call +65 1234 5678 (9AM~6PM) / E-mail: qwe123@abc.com

Prototype: Architecture for AI web crawling system

User interaction:

1. When the user executes the program, enter the product name to search.
2. User can obtain the product and requested description (sales pitch message) through the terminal or command prompt.

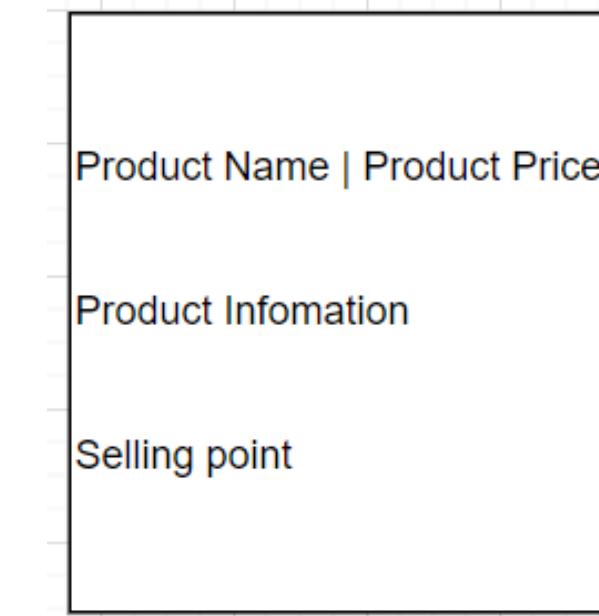
Output file after running the program:



Program operation process:

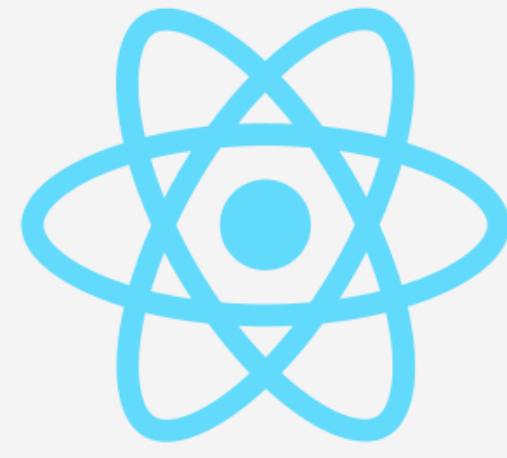
1. Receive search keywords from the user.
2. Crawl the search results page of an online shopping site (Naver Shopping Mall) based on the entered keywords.
3. Extract product name, price, detailed information, and product image URLs from the crawled data.
4. Translate collected product names from Korean to English.
5. Download images using the URLs of the crawled product images. Images will be saved based on the translated product names in English.
6. Use an AI model to generate conversation messages based on the collected product information and specified requirements.
7. The AI model will generate messages that include product descriptions, price information, selling points, etc. The generated messages will be saved in a text file with the title being the translated product name in English.
8. Provide users with product information, image download links, AI-generated conversational messages, and more, all output to a terminal window.

Contents of the txt file:



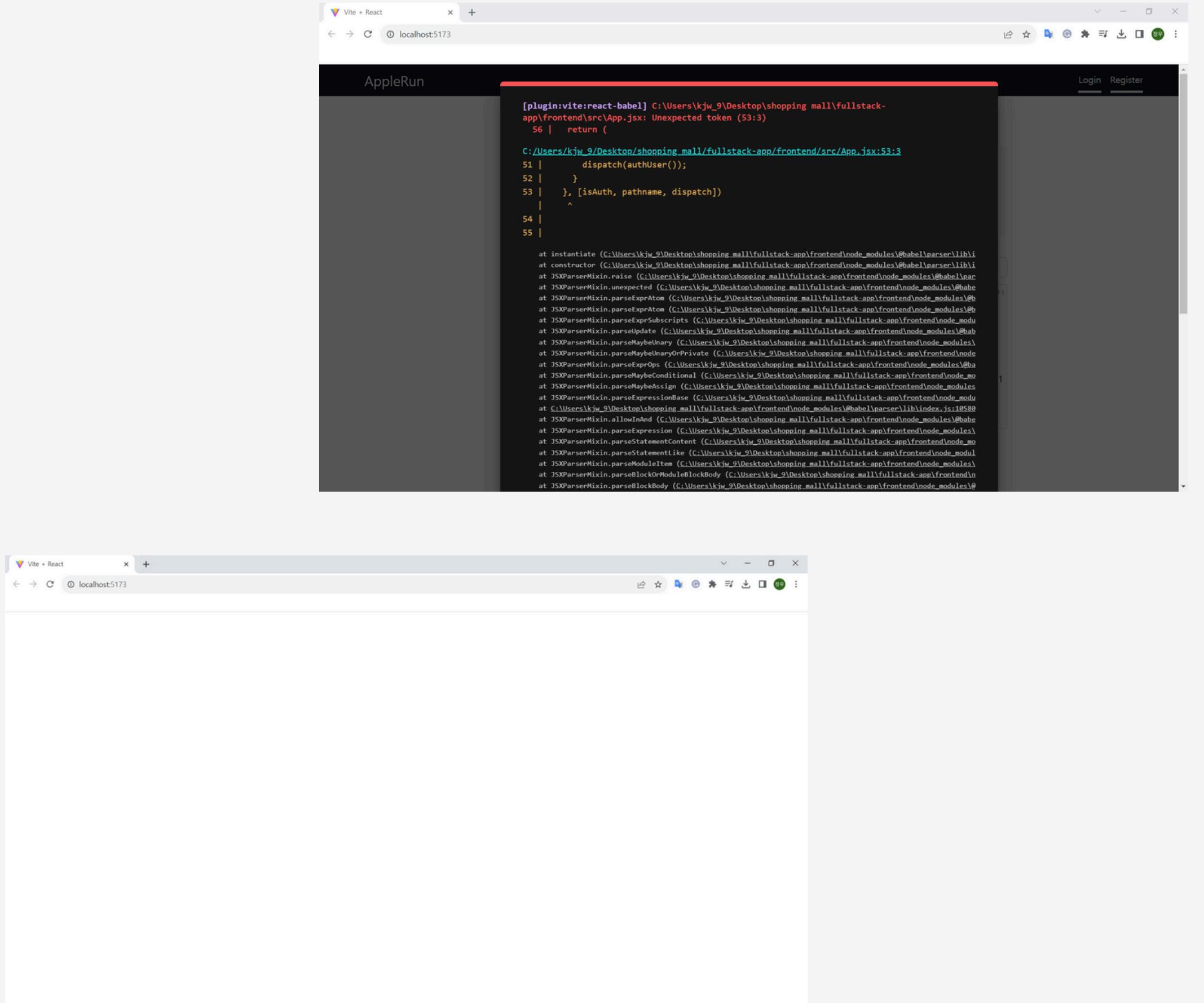
Achievements

- Learned React and JavaScript for the first time, and successfully applied them in a project.
- Built a project using MongoDB and UI/UX design principles learned from a previous class. Also incorporated the latest popular technology, GPT, into the project.
- Mastered backend technologies for server and database management.
- Improved skills in analyzing online survey data and organizing information.
- Developed an AI-based web crawling system in Python, improving Python skills, AI understanding, and data analysis and machine learning abilities.



Challenges

- Without GPT, converting and structuring data obtained through crawling would have been difficult.
- Due to my limited knowledge in server configuration during the backend development process, unexpected delays occurred.
- UI/UX survey responses were slow, causing unforeseen project delays.
- A complex issue with a white screen during development required significant time and effort to resolve, posing a challenging task.



The image shows two screenshots of a web browser window. The top screenshot displays a modal dialog titled "AppleRun" with a red border. The modal contains a stack trace from a JavaScript error:

```
[plugin:vite:react-babel] C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\src\App.jsx: Unexpected token (53:3)
56 |     return (
C:/Users/kjw_9/Desktop/shopping_mall/fullstack-app/frontend/src/App.jsx:53:3
51 |         dispatch(authUser());
52 |     }
53 | }, [isAuth, pathname, dispatch])
|
54 |
55 |

at instantiate (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at constructor (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.raise (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.unexpected (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseExprAtom (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseExprAtom (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseExprSubscripts (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseUpdate (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseMaybeUnary (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseMaybeUnaryOrPrivate (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseExprOp (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseMaybeConditional (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseMaybeAssign (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseExpressionBase (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10
at JSXParserMixin.allowInAnd (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseExpression (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseStatementContent (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseStatementLike (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseModuleItem (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseBlockModuleBody (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
at JSXParserMixin.parseBlockBody (C:\Users\kjw_9\Desktop\shopping_mall\fullstack-app\frontend\node_modules\babel\parser\lib\index.js:10589:10)
```

The bottom screenshot shows a blank white page with the browser's address bar indicating "localhost:5173".

Thank you!

Please feel free to contact me
if you have any questions.

