Noor.AI – Skincare Product Recommendation Feature Development Report

1. Introduction

The Skincare Product Recommendation feature of Noor.AI helps users discover suitable skincare products based on their selected skin type. This standalone module was built using Python on Google Colab and was developed separately for testing before full integration into the Noor.AI web application. The recommendation logic is based on a Kaggle dataset of skincare products, which is filtered according to the user's selected skin type (dry, oily, or normal). If a product is marked suitable for all three skin types, it is also included in the recommended results. Once the logic was tested in Python, the module was turned into a Flask server with help from Gemini, and a simple HTML frontend was created with assistance from Claude.ai.

2. Tools & Technologies Used

Tool	Purpose
Python	Main language for building the recommendation system
Google Colab	Environment for developing and testing Python logic
Kaggle	Source of the skincare product dataset
Flask	Used to serve the recommendation feature as a web API
Git & GitHub	Version control and backup
Claude.ai	Helped build the HTML page for the frontend
Gemini	Helped integrate Python code into a Flask web server

3. Dataset & Preprocessing

Used a skincare product dataset from Kaggle, which included product names, descriptions, ingredients, and associated skin types.
Data was loaded and processed in Python using Pandas for easier filtering.
Each product was categorized under one or more skin types: dry, oily, normal, or all.

4. Recommendation System Design

A simple rule-based filtering logic was implemented using Python:

- If the user selects dry skin, products marked for dry and all skin types are shown.
- The same logic applies for oily and normal skin types.
- The system was first tested in Google Colab to validate filtering accuracy.
- After confirming the logic, the Python code was adapted into a Flask server.
- The final step was to connect the Flask API to an HTML form, where users could select their skin type and receive product suggestions.

5. Feature Workflow

1. User selects a skin type (dry, oily, normal) from a dropdown menu on the frontend.
2. The selection is sent to the Flask backend via a form submission.
3. Backend filters the product dataset and returns a list of suitable products:
    - Products explicitly match the selected skin type.
    - Products marked as suitable for all skin types.
4. Results are displayed on the HTML page for the user to view.

6. Challenges Faced

- Preparing and cleaning the dataset to ensure each product was properly categorized.
- Handling the additional logic for multi-type or “all skin type” products.
- Figuring out how to connect the Python logic to a working Flask server.
- Testing the dynamic display of results on the frontend HTML page.
- Validating feature accuracy and ensuring a consistent user experience.

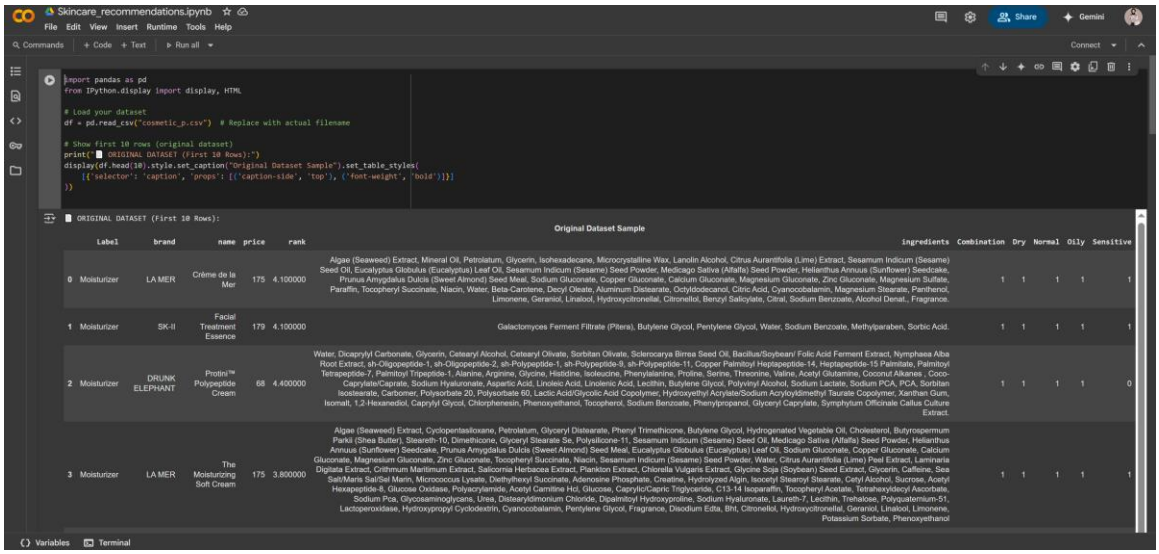
7. Future Improvements

- Add more skin concerns such as acne-prone, sensitive, or combination.
- Include filters like product price, brand, or key ingredients.
- Add product links, allowing users to access the official websites to purchase recommended skincare products.
- Enhance UI/UX by adding images, user reviews, and better layout design.

8. Screenshots

Step 1: Dataset and Filtering Logic

Google Colab was used to test and refine filtering logic based on skin type, including support for products that are suitable for all.

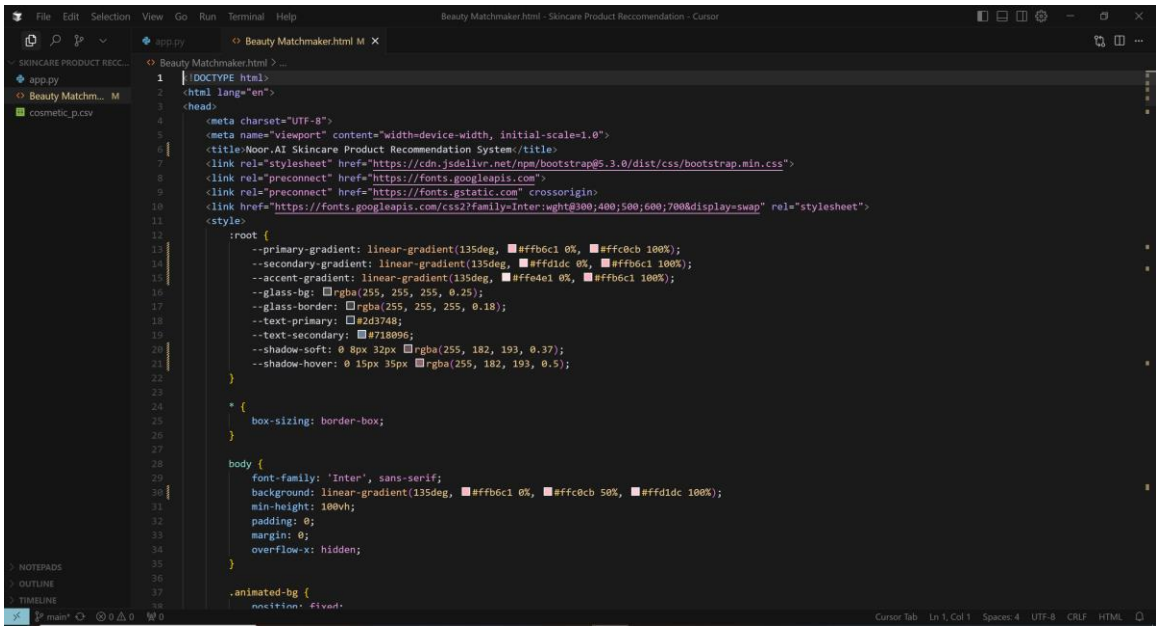


Step 2: Flask Integration

With Gemini's help, the logic was wrapped into a Flask server, allowing input to be received via a browser and appropriate results to be returned.

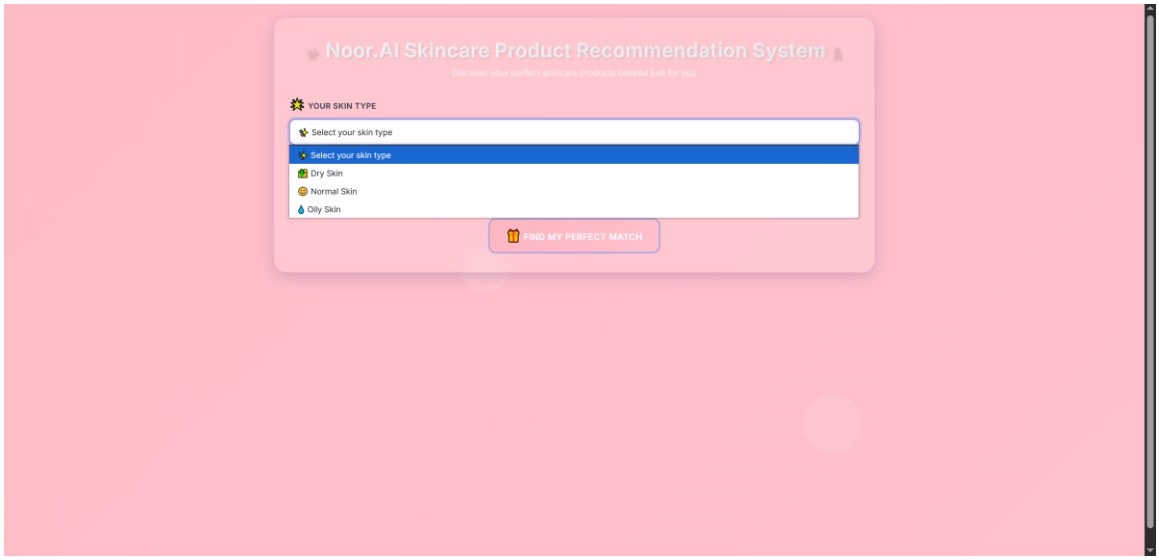
Step 3: Frontend with HTML

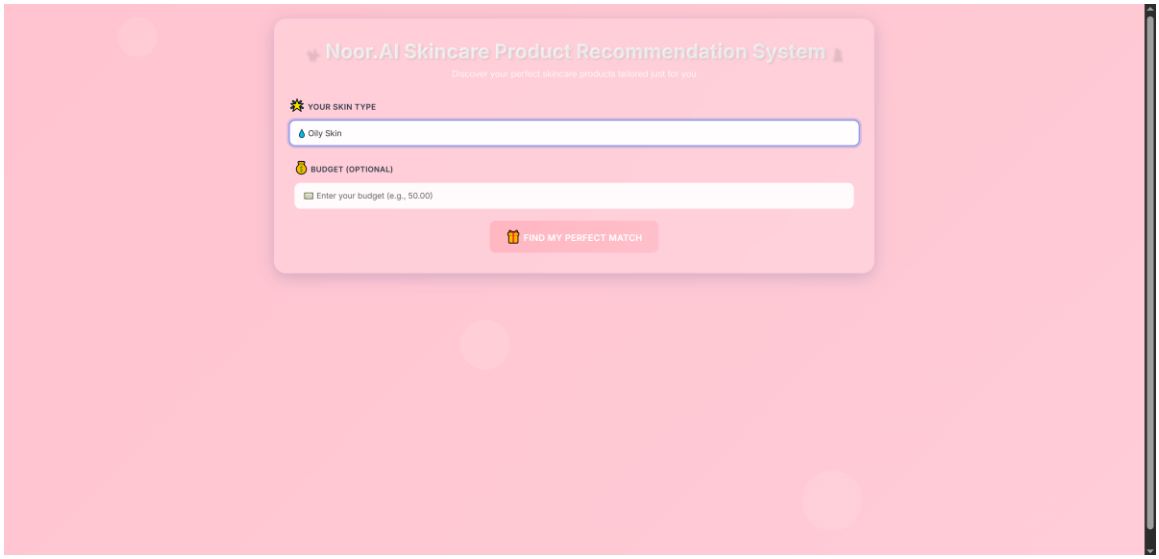
Claude.ai assisted in creating a basic HTML page featuring a dropdown for skin type selection. The HTML page connects to the Flask backend and displays recommended products.

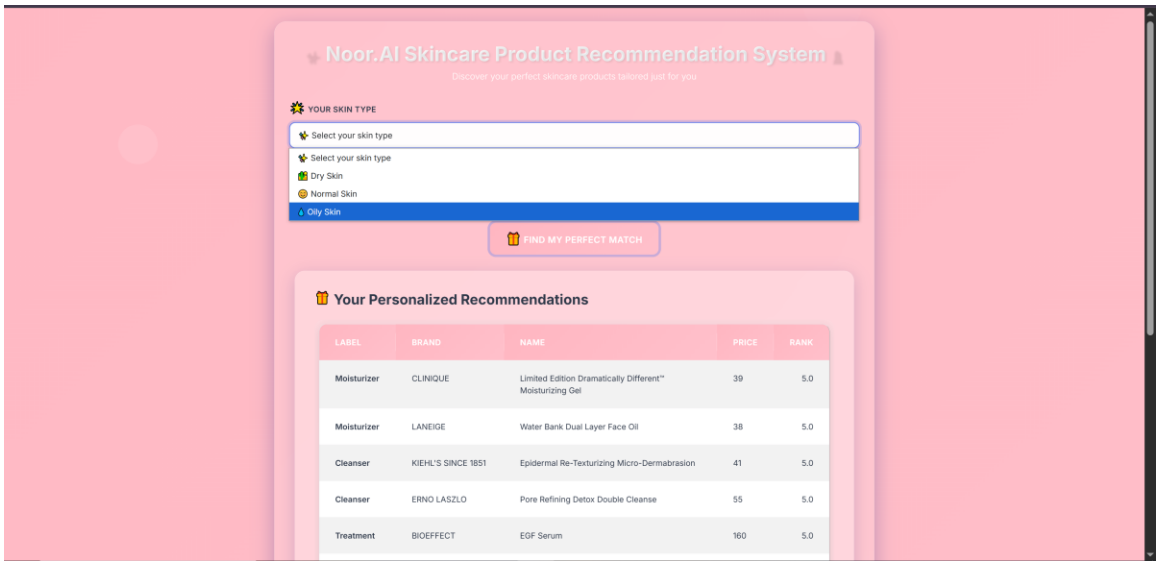


Step 4: End-to-End Testing

All components—selection input, backend logic, and frontend result display—were tested to ensure products for selected all different skin types were accurately recommended.







9. Conclusion

This module was developed as a testable standalone feature, focusing on accurate skincare product recommendations based on user-selected skin types. Products tagged for all skin types are intelligently included to broaden recommendations. After successful development and validation, it is now ready for integration into the Noor.AI web application.