Lesson 07 Demo 02

Analyzing Requirements Using GenAI

Objective: To analyze requirements and generate detailed acceptance criteria for the user stories using ChatGPT for enhancing the Software Testing Life Cycle

Tools required: ChatGPT 4

Prerequisites: None

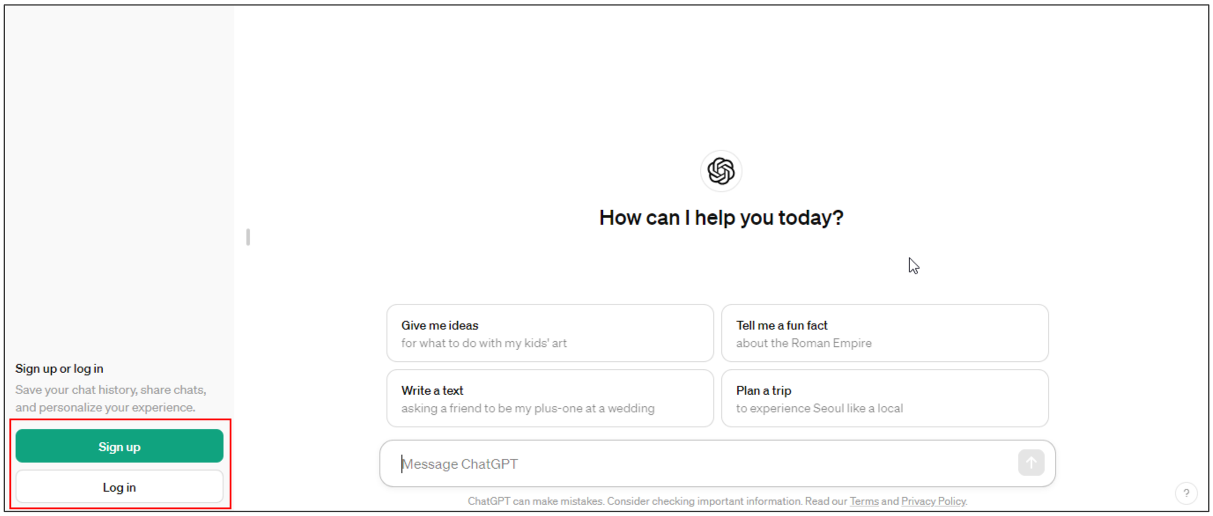
Steps to be followed:

1. Clarify requirements using ChatGPT 4
2. Generate the focus areas using ChatGPT 4
3. Generate acceptance criteria using ChatGPT 4

Note: Please be advised that ChatGPT, as an artificial intelligence tool, can produce varied outputs even when presented with similar prompts.

Step 1: Clarify requirements using ChatGPT 4

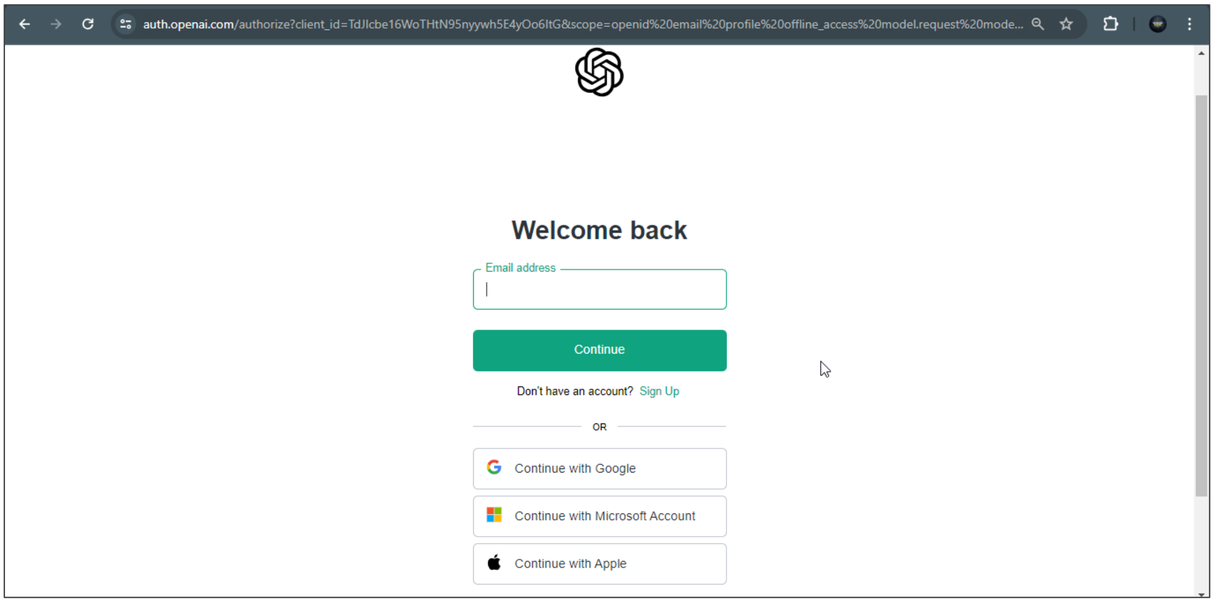
1.1 Browse to the **https://chat.openai.com** website and log in to your account


How can I help you today?
Give me ideas
for what to do with my kids' art
Tell me a fun fact
about the Roman Empire
Write a text
asking a friend to be my plus-one at a wedding
Plan a trip
to experience Seoul like a local
Message ChatGPT
Sign up or log in
Sign up
Log in

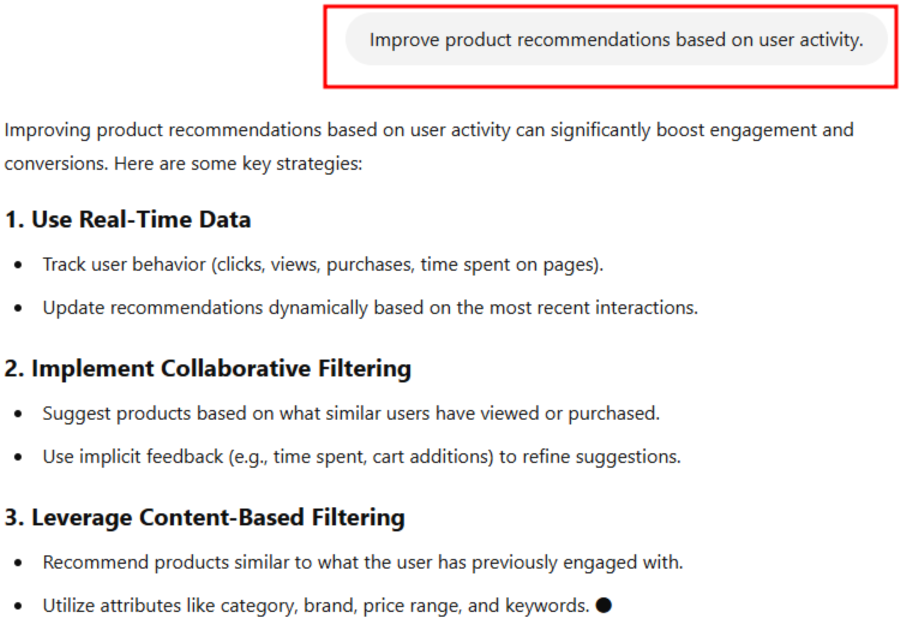
Save your chat history, share chats, and personalize your experience.

ChatGPT can make mistakes. Consider checking important information. Read our Terms and Privacy Policy.

Note: Sign up if you do not have an account


auth.openai.com/authorize?client_id=TdJlcbe16WoTHtN95nyywh5E4yOo6ltG&scope=openid%20email%20profile%20offline_access%20model.request%20mode...
Welcome back
Email address
Continue
Don't have an account? Sign Up
OR
Continue with Google
Continue with Microsoft Account
Continue with Apple

1.2 Use the following prompt to input broad requirements, as shown in the screenshot below:
Improve product recommendations based on user activity.



Improve product recommendations based on user activity.

Improving product recommendations based on user activity can significantly boost engagement and conversions. Here are some key strategies:

1. Use Real-Time Data

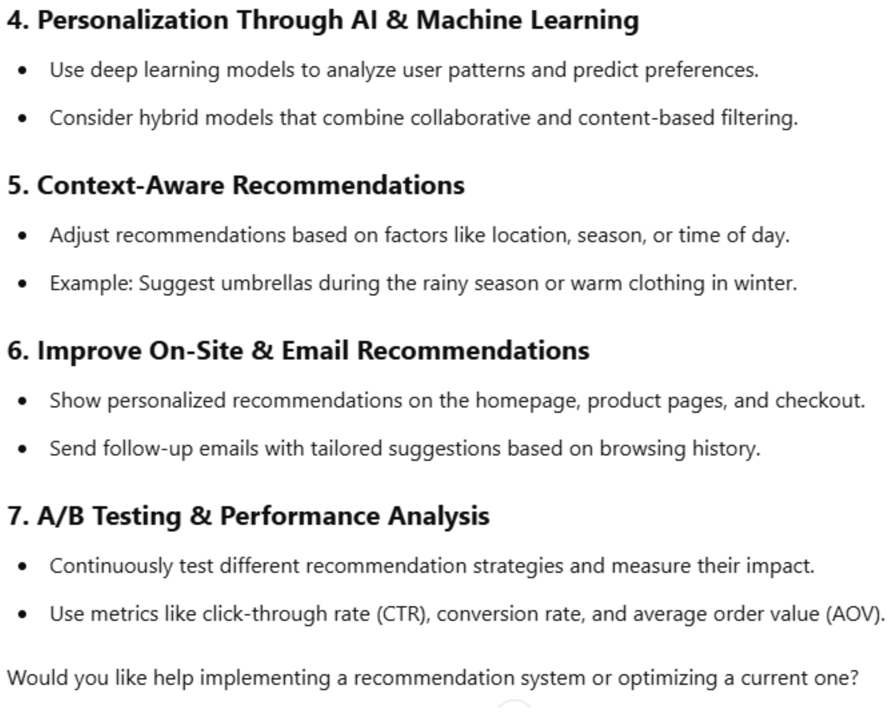
- Track user behavior (clicks, views, purchases, time spent on pages).
- Update recommendations dynamically based on the most recent interactions.

2. Implement Collaborative Filtering

- Suggest products based on what similar users have viewed or purchased.
- Use implicit feedback (e.g., time spent, cart additions) to refine suggestions.

3. Leverage Content-Based Filtering

- Recommend products similar to what the user has previously engaged with.
- Utilize attributes like category, brand, price range, and keywords. ●



4. Personalization Through AI & Machine Learning

- Use deep learning models to analyze user patterns and predict preferences.
- Consider hybrid models that combine collaborative and content-based filtering.

5. Context-Aware Recommendations

- Adjust recommendations based on factors like location, season, or time of day.
- Example: Suggest umbrellas during the rainy season or warm clothing in winter.

6. Improve On-Site & Email Recommendations

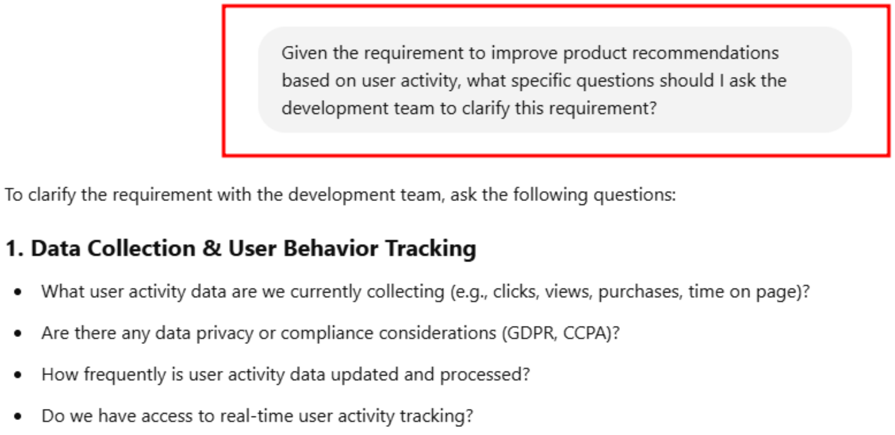
- Show personalized recommendations on the homepage, product pages, and checkout.
- Send follow-up emails with tailored suggestions based on browsing history.

7. A/B Testing & Performance Analysis

- Continuously test different recommendation strategies and measure their impact.
- Use metrics like click-through rate (CTR), conversion rate, and average order value (AOV).

Would you like help implementing a recommendation system or optimizing a current one?

1.3 Use the following prompt for clarification, as shown in the screenshot below:
Given the requirement to improve product recommendations based on user activity, what specific questions should I ask the development team to clarify this requirement?

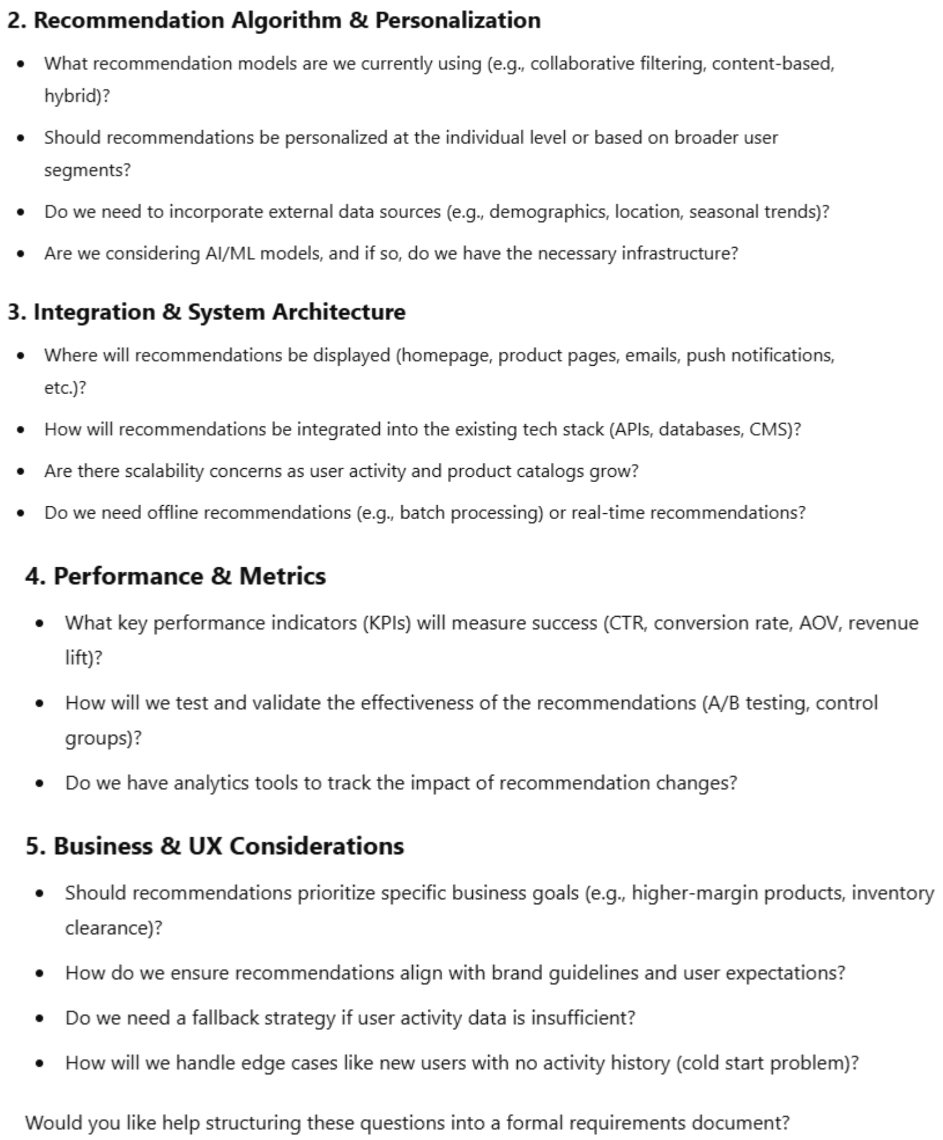


Given the requirement to improve product recommendations based on user activity, what specific questions should I ask the development team to clarify this requirement?

To clarify the requirement with the development team, ask the following questions:

1. Data Collection & User Behavior Tracking

- What user activity data are we currently collecting (e.g., clicks, views, purchases, time on page)?
- Are there any data privacy or compliance considerations (GDPR, CCPA)?
- How frequently is user activity data updated and processed?
- Do we have access to real-time user activity tracking?



2. Recommendation Algorithm & Personalization

- What recommendation models are we currently using (e.g., collaborative filtering, content-based, hybrid)?
- Should recommendations be personalized at the individual level or based on broader user segments?
- Do we need to incorporate external data sources (e.g., demographics, location, seasonal trends)?
- Are we considering AI/ML models, and if so, do we have the necessary infrastructure?

3. Integration & System Architecture

- Where will recommendations be displayed (homepage, product pages, emails, push notifications, etc.)?
- How will recommendations be integrated into the existing tech stack (APIs, databases, CMS)?
- Are there scalability concerns as user activity and product catalogs grow?
- Do we need offline recommendations (e.g., batch processing) or real-time recommendations?

4. Performance & Metrics

- What key performance indicators (KPIs) will measure success (CTR, conversion rate, AOV, revenue lift)?
- How will we test and validate the effectiveness of the recommendations (A/B testing, control groups)?
- Do we have analytics tools to track the impact of recommendation changes?

5. Business & UX Considerations

- Should recommendations prioritize specific business goals (e.g., higher-margin products, inventory clearance)?
- How do we ensure recommendations align with brand guidelines and user expectations?
- Do we need a fallback strategy if user activity data is insufficient?
- How will we handle edge cases like new users with no activity history (cold start problem)?

Would you like help structuring these questions into a formal requirements document?

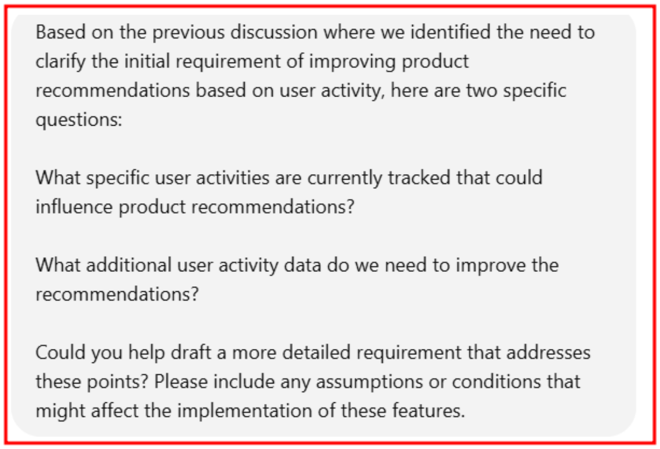
1.4 Use the following prompt to refine the requirements, as shown in the screenshot below:

Based on the previous discussion where we identified the need to clarify the initial requirement of improving product recommendations based on user activity, here are two specific questions:

What specific user activities are currently tracked that could influence product recommendations?

What additional user activity data do we need to improve the recommendations?

Could you help draft a more detailed requirement that addresses these points? Please include any assumptions or conditions that might affect the implementation of these features.

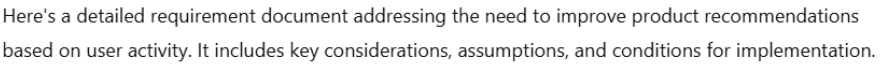


Based on the previous discussion where we identified the need to clarify the initial requirement of improving product recommendations based on user activity, here are two specific questions:

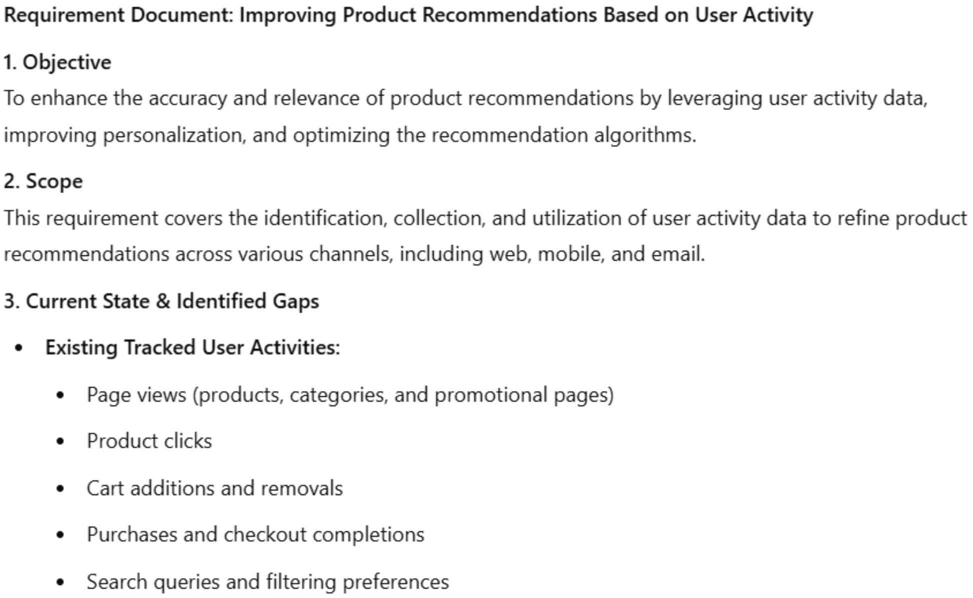
What specific user activities are currently tracked that could influence product recommendations?

What additional user activity data do we need to improve the recommendations?

Could you help draft a more detailed requirement that addresses these points? Please include any assumptions or conditions that might affect the implementation of these features.



Here's a detailed requirement document addressing the need to improve product recommendations based on user activity. It includes key considerations, assumptions, and conditions for implementation.



Requirement Document: Improving Product Recommendations Based on User Activity

1. Objective

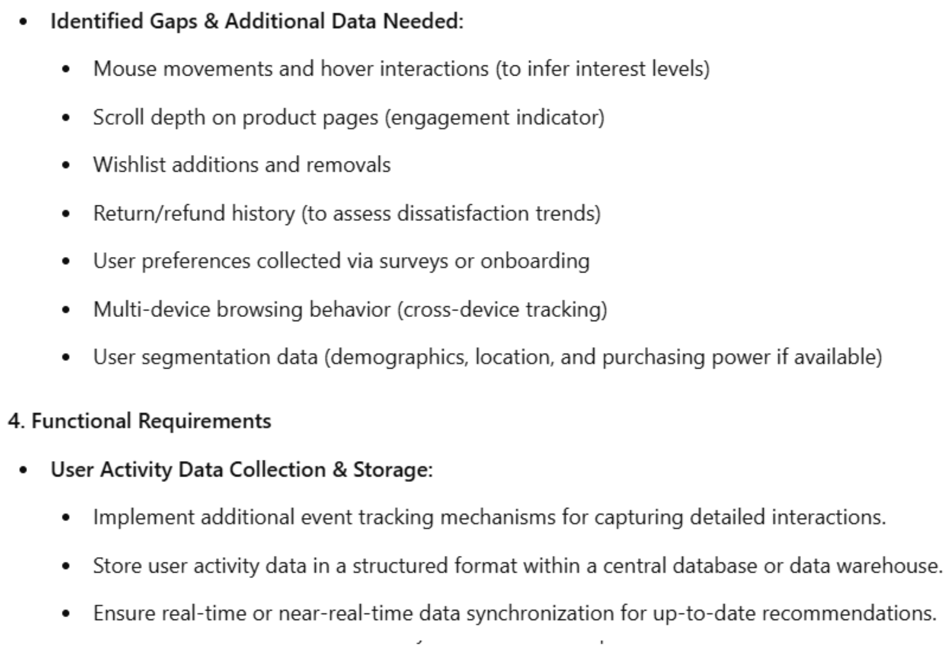
To enhance the accuracy and relevance of product recommendations by leveraging user activity data, improving personalization, and optimizing the recommendation algorithms.

2. Scope

This requirement covers the identification, collection, and utilization of user activity data to refine product recommendations across various channels, including web, mobile, and email.

3. Current State & Identified Gaps

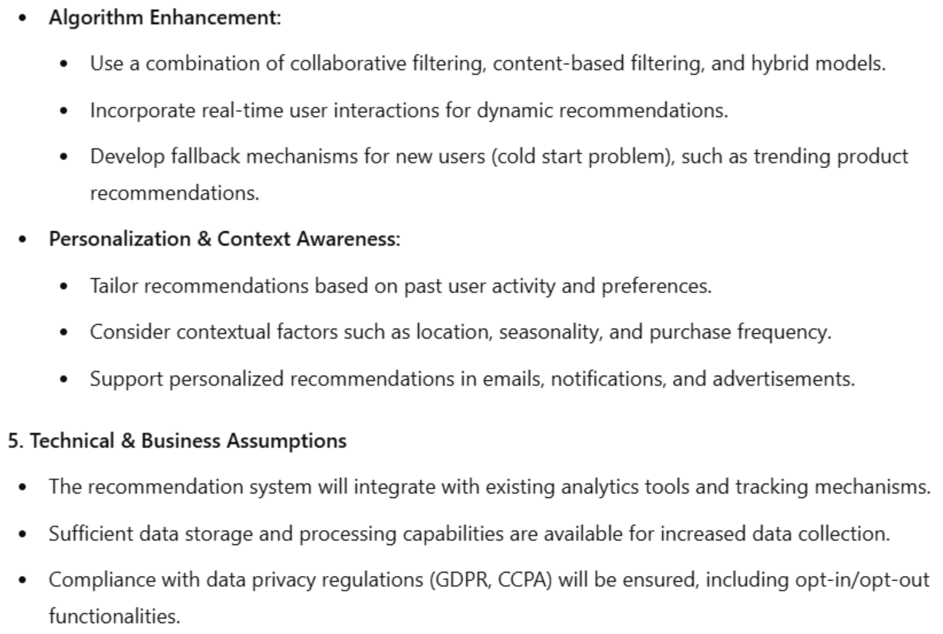
- **Existing Tracked User Activities:**
  - Page views (products, categories, and promotional pages)
  - Product clicks
  - Cart additions and removals
  - Purchases and checkout completions
  - Search queries and filtering preferences



- **Identified Gaps & Additional Data Needed:**
  - Mouse movements and hover interactions (to infer interest levels)
  - Scroll depth on product pages (engagement indicator)
  - Wishlist additions and removals
  - Return/refund history (to assess dissatisfaction trends)
  - User preferences collected via surveys or onboarding
  - Multi-device browsing behavior (cross-device tracking)
  - User segmentation data (demographics, location, and purchasing power if available)

4. Functional Requirements

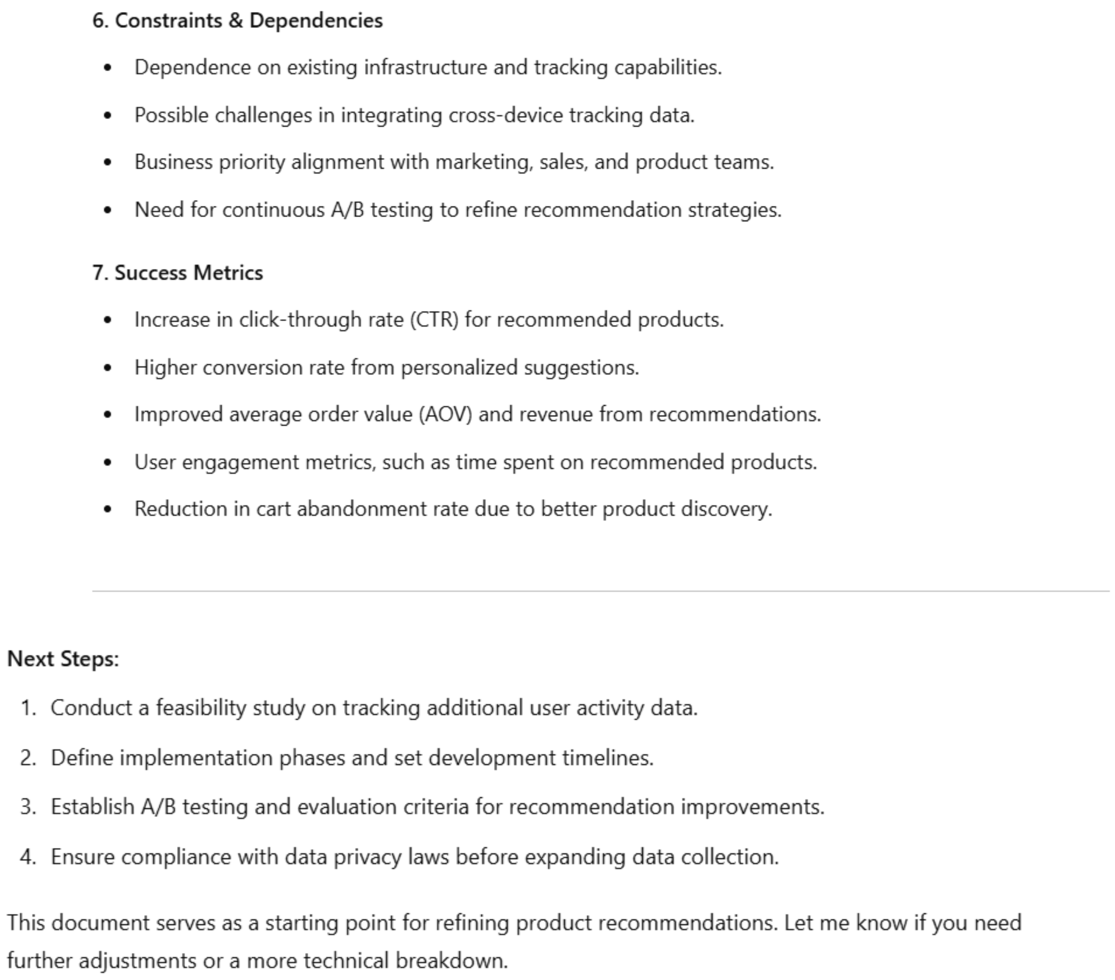
- **User Activity Data Collection & Storage:**
  - Implement additional event tracking mechanisms for capturing detailed interactions.
  - Store user activity data in a structured format within a central database or data warehouse.
  - Ensure real-time or near-real-time data synchronization for up-to-date recommendations.



- **Algorithm Enhancement:**
  - Use a combination of collaborative filtering, content-based filtering, and hybrid models.
  - Incorporate real-time user interactions for dynamic recommendations.
  - Develop fallback mechanisms for new users (cold start problem), such as trending product recommendations.
- **Personalization & Context Awareness:**
  - Tailor recommendations based on past user activity and preferences.
  - Consider contextual factors such as location, seasonality, and purchase frequency.
  - Support personalized recommendations in emails, notifications, and advertisements.

5. Technical & Business Assumptions

- The recommendation system will integrate with existing analytics tools and tracking mechanisms.
- Sufficient data storage and processing capabilities are available for increased data collection.
- Compliance with data privacy regulations (GDPR, CCPA) will be ensured, including opt-in/opt-out functionalities.



6. Constraints & Dependencies

- Dependence on existing infrastructure and tracking capabilities.
- Possible challenges in integrating cross-device tracking data.
- Business priority alignment with marketing, sales, and product teams.
- Need for continuous A/B testing to refine recommendation strategies.

7. Success Metrics

- Increase in click-through rate (CTR) for recommended products.
- Higher conversion rate from personalized suggestions.
- Improved average order value (AOV) and revenue from recommendations.
- User engagement metrics, such as time spent on recommended products.
- Reduction in cart abandonment rate due to better product discovery.

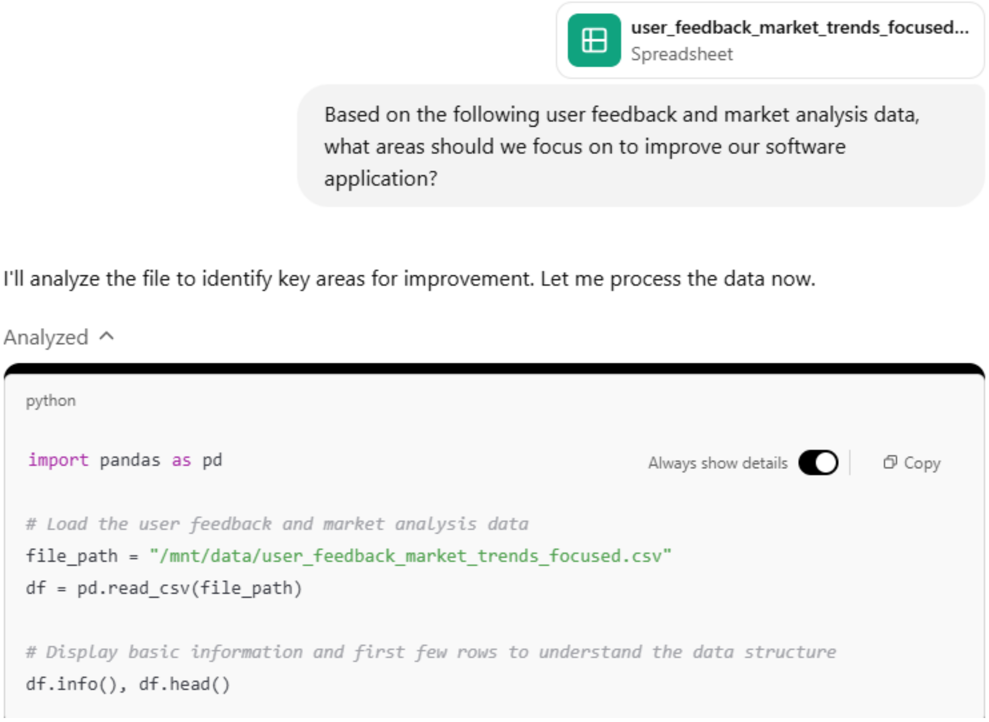
Next Steps:

1. Conduct a feasibility study on tracking additional user activity data.
2. Define implementation phases and set development timelines.
3. Establish A/B testing and evaluation criteria for recommendation improvements.
4. Ensure compliance with data privacy laws before expanding data collection.

This document serves as a starting point for refining product recommendations. Let me know if you need further adjustments or a more technical breakdown.

Step 2: Generate the focus areas using ChatGPT 4

2.1 Upload the market and user data and generate targeted suggestions using the prompt provided in the screenshot below:
Based on the following user feedback and market analysis data, what areas should we focus on to improve our software application?



Based on the following user feedback and market analysis data, what areas should we focus on to improve our software application?

I'll analyze the file to identify key areas for improvement. Let me process the data now.

Analyzed

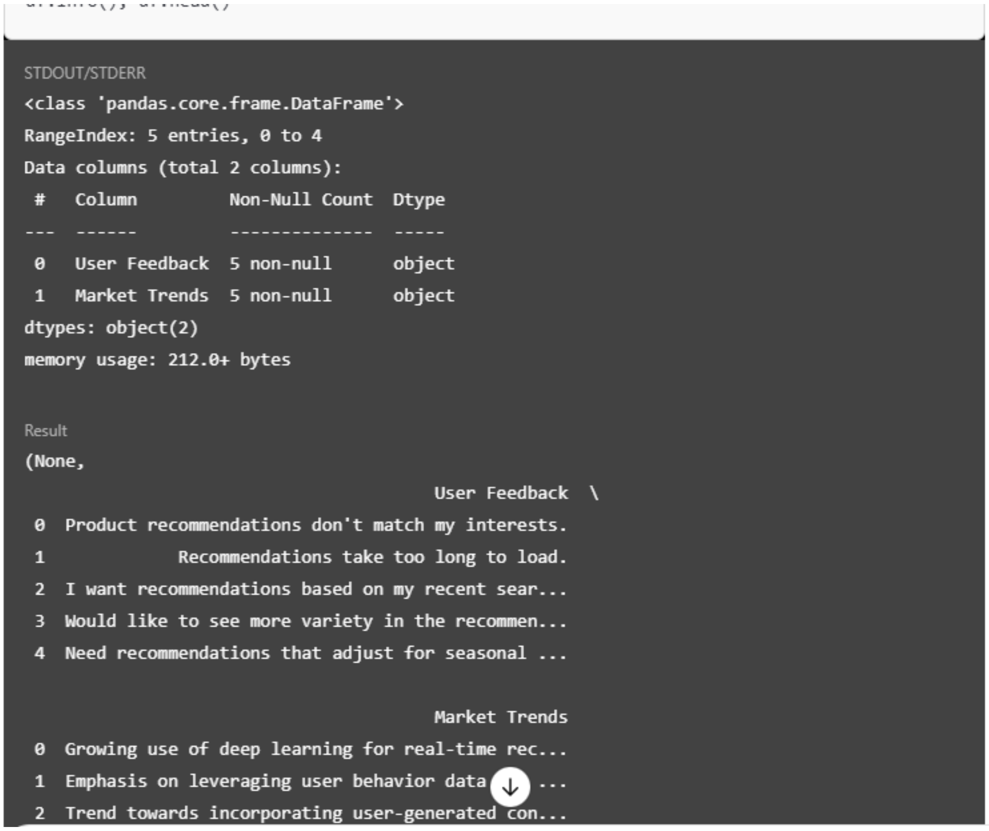
python

```python
import pandas as pd

# Load the user feedback and market analysis data
file_path = "/mnt/data/user_feedback_market_trends_focused.csv"
df = pd.read_csv(file_path)

# Display basic information and first few rows to understand the data structure
df.info(), df.head()
```

2.2



```
df.info(), df.head()
```

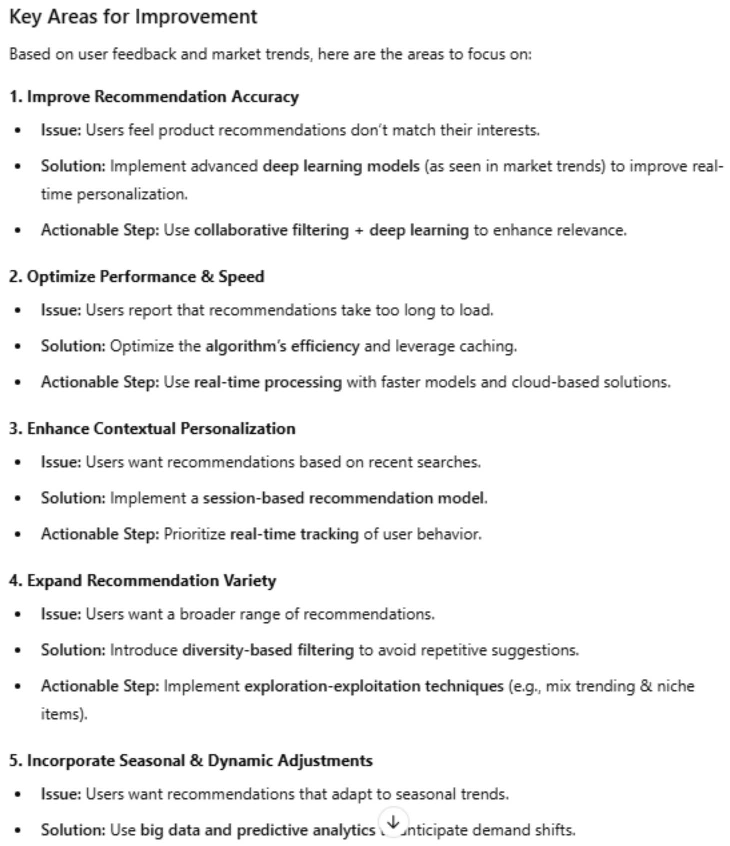
STDOUT/STDERR

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   User Feedback   5 non-null      object
 1   Market Trends   5 non-null      object
dtypes: object(2)
memory usage: 212.0+ bytes
```

Result

```
(None,
                                          User Feedback  \
 0  Product recommendations don't match my interests.   
 1             Recommendations take too long to load.   
 2  I want recommendations based on my recent sear...   
 3  Would like to see more variety in the recommen...   
 4  Need recommendations that adjust for seasonal ...   

                                          Market Trends  
 0  Growing use of deep learning for real-time rec...  
 1  Emphasis on leveraging user behavior data     ...  
 2  Trend towards incorporating user-generated con...  
```



Key Areas for Improvement

Based on user feedback and market trends, here are the areas to focus on:

1. Improve Recommendation Accuracy

- **Issue:** Users feel product recommendations don't match their interests.
- **Solution:** Implement advanced **deep learning models** (as seen in market trends) to improve real-time personalization.
- **Actionable Step:** Use **collaborative filtering + deep learning** to enhance relevance.

2. Optimize Performance & Speed

- **Issue:** Users report that recommendations take too long to load.
- **Solution:** Optimize the **algorithm's efficiency** and leverage caching.
- **Actionable Step:** Use **real-time processing** with faster models and cloud-based solutions.

3. Enhance Contextual Personalization

- **Issue:** Users want recommendations based on recent searches.
- **Solution:** Implement a **session-based recommendation model**.
- **Actionable Step:** Prioritize **real-time tracking** of user behavior.

4. Expand Recommendation Variety

- **Issue:** Users want a broader range of recommendations.
- **Solution:** Introduce **diversity-based filtering** to avoid repetitive suggestions.
- **Actionable Step:** Implement **exploration-exploitation techniques** (e.g., mix trending & niche items).

5. Incorporate Seasonal & Dynamic Adjustments

- **Issue:** Users want recommendations that adapt to seasonal trends.
- **Solution:** Use **big data and predictive analytics** to anticipate demand shifts.

Once you get to the focus areas, you may discuss the feasibility and relevance of the focus areas with the project team.

2.3 Use the following prompt to adjust the priorities, as shown in the screenshot below: **Given the recommended focus areas to enhance our recommendation engine with deep learning, expand data analytics, and improve personalization and variety, how should we adjust our software development and testing priorities to best implement these improvements efficiently and effectively?**

Given the recommended focus areas to enhance our recommendation engine with deep learning, expand data analytics, and improve personalization and variety, how should we adjust our software development and testing priorities to best implement these improvements efficiently and effectively?

To efficiently and effectively implement these improvements, your **software development and testing priorities** should focus on **incremental enhancements, rigorous testing, and agile development cycles**. Here's a structured approach:

---

1. Software Development Priorities

A. Upgrade the Recommendation Engine with Deep Learning

◆ **Actionable Steps:**

- Implement a **hybrid recommendation model** (Collaborative Filtering + Deep Learning).
- Utilize **Neural Networks (e.g., Transformers, RNNs) or Reinforcement Learning** for better personalization.
- Optimize for **real-time processing** to reduce latency.
- Leverage **cloud-based AI services (AWS SageMaker, Google Vertex AI)** for scalable computation.

◆ **Tech Stack Recommendations:**

- TensorFlow/PyTorch for deep learning models.
- Apache Spark or Kafka for real-time data streaming.
- Scalable databases (e.g., MongoDB, BigQuery) for handling large datasets.

B. Expand Data Analytics for Contextual & Seasonal Personalization

◆ **Actionable Steps:**

- Implement **user session tracking** for real-time adjustments.
- Use **predictive analytics** to anticipate seasonal and trend-based preferences.
- Integrate **big data processing** (Hadoop/Spark) to analyze behavioral patterns efficiently.

◆ **Tech Stack Recommendations:**

- Snowflake or Redshift for big data analysis.
- Python (Pandas, NumPy, SciPy) for data processing.
- A/B testing tools (Optimizely, Google Optimize) to evaluate changes.

2. Software Testing Priorities

A. Model Performance & Accuracy Testing

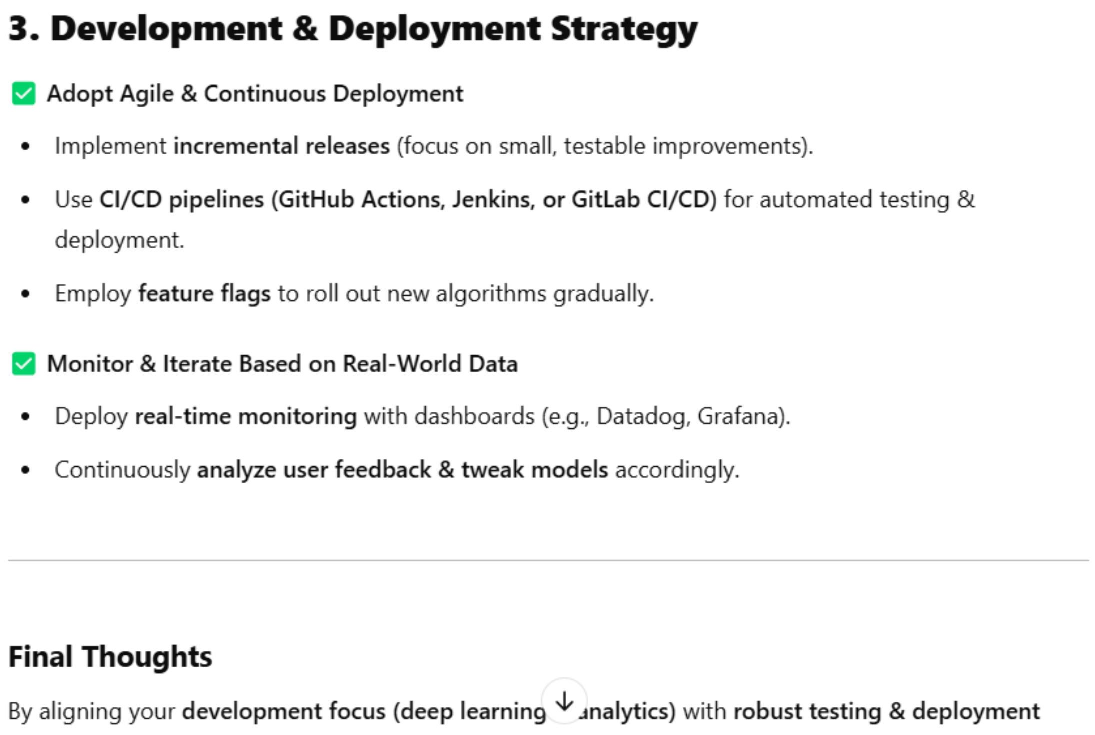
- **Evaluate recommendation accuracy** using **Precision, Recall, and F1 Score**.
- **A/B test models** to measure **CTR (Click-Through Rate) and Conversion Rate improvements**.
- Use **offline validation (cross-validation) and online evaluation (live user testing)**.

B. System Performance & Scalability Testing

- Conduct **load testing** to ensure real-time recommendation response times.
- Optimize **database queries and caching** (e.g., Redis, Memcached).
- Implement **distributed computing** for scalable deep learning inference.

C. User Experience (UX) Testing

- Conduct **usability testing** to ensure recommendations are intuitive.
- Monitor **engagement metrics** (e.g., bounce rate, dwell time).
- Ensure **multi-device consistency** (web, mobile, ↓ rt devices).



3. Development & Deployment Strategy

☑ Adopt Agile & Continuous Deployment

- Implement **incremental releases** (focus on small, testable improvements).
- Use **CI/CD pipelines (GitHub Actions, Jenkins, or GitLab CI/CD)** for automated testing & deployment.
- Employ **feature flags** to roll out new algorithms gradually.

☑ Monitor & Iterate Based on Real-World Data

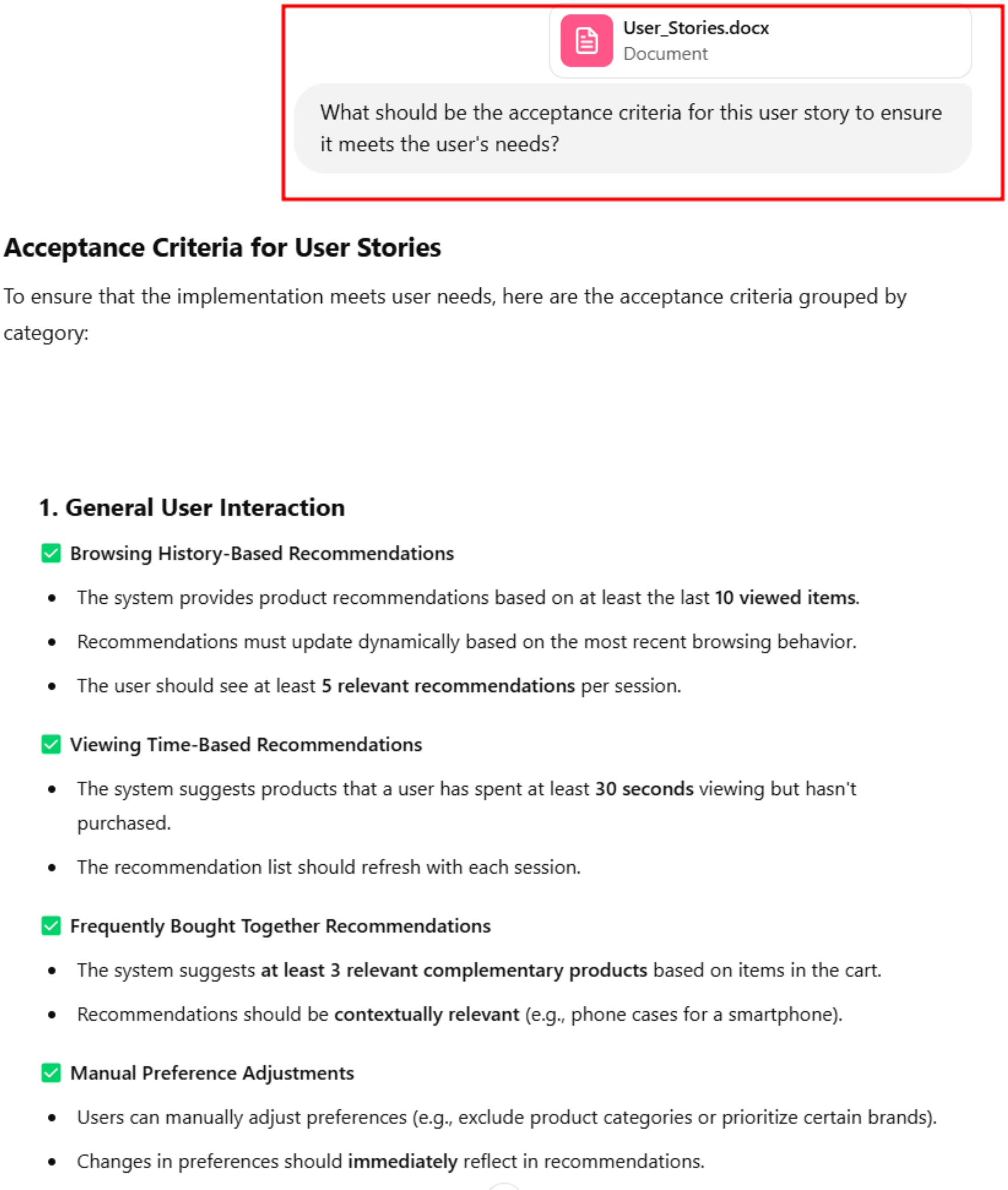
- Deploy **real-time monitoring** with dashboards (e.g., Datadog, Grafana).
- Continuously **analyze user feedback & tweak models** accordingly.

Final Thoughts

By aligning your **development focus (deep learning ↓ analytics)** with **robust testing & deployment**

Step 3: Generate acceptance criteria using ChatGPT 4

3.1 Upload the user stories and use the following prompt to generate the acceptance criteria, as shown in the screenshot below:
What should be the acceptance criteria for this user story to ensure it meets the user's needs?



User_Stories.docx
Document

What should be the acceptance criteria for this user story to ensure it meets the user's needs?

Acceptance Criteria for User Stories

To ensure that the implementation meets user needs, here are the acceptance criteria grouped by category:

1. General User Interaction

✅ **Browsing History-Based Recommendations**

- The system provides product recommendations based on at least the last **10 viewed items**.
- Recommendations must update dynamically based on the most recent browsing behavior.
- The user should see at least **5 relevant recommendations** per session.

✅ **Viewing Time-Based Recommendations**

- The system suggests products that a user has spent at least **30 seconds** viewing but hasn't purchased.
- The recommendation list should refresh with each session.

✅ **Frequently Bought Together Recommendations**

- The system suggests **at least 3 relevant complementary products** based on items in the cart.
- Recommendations should be **contextually relevant** (e.g., phone cases for a smartphone).

✅ **Manual Preference Adjustments**

- Users can manually adjust preferences (e.g., exclude product categories or prioritize certain brands).
- Changes in preferences should **immediately** reflect in recommendations.

2. User Data Analysis

✅ **Clickstream Data Extraction**

- The system captures **user clicks, dwell time, and purchase history.**
- Data should be stored securely and retrievable for trend analysis.

✅ **Purchase History-Based Recommendations**

- The system analyzes past purchases and suggests **at least 5 similar items.**
- Users should be able to mark recommendations as "Not Interested."

✅ **Marketing Trend Analysis**

- The system provides a dashboard that visualizes trends based on aggregated user behavior.
- Data should be updated **daily** for real-time insights.

3. Real-Time & Contextual Recommendations

✅ **Real-Time Updates**

- Recommendations must refresh **within 2 seconds** of user interactions.
- The system should prioritize **most recent** engagement over older data.

✅ **Session-Based Promotions**

- Users receive personalized promotional offers based on current browsing activity.
- Discounts or promotions appear within **30 seconds** of relevant activity.

✅ **Time-Based Adjustments**

- Morning sessions (6 AM - 12 PM) prioritize breakfast-related items.
- Evening sessions (6 PM - 12 AM) prioritize entertainment and relaxation products.

4. Performance & Load Times

✅ **Speed Optimization**

- Recommendations should load **within 1 second.**
- System performance should not degrade for users on mobile devices.

✅ **Scalability**

- The system should handle at least **100,000 concurrent users** without latency issues.
- A fallback mechanism should exist if the recommendation engine fails.

5. Privacy & Security

✅ **Data Protection**

- All user activity data must be encrypted **at rest and in transit**.
- Users should have the ability to opt out of personalized recommendations.

✅ **Regulatory Compliance**

- The system must adhere to **GDPR** and **CCPA** guidelines.
- A privacy policy must be accessible, detailing how recommendation data is used.

6. User Feedback & Adaptability

✅ **User Feedback Mechanism**

- Users should be able to **like/dislike** recommendations.
- Feedback should update recommendations within **24 hours**.

✅ **Product Manager Insights**

- A report summarizing user feedback trends should be available **weekly**.
- Adjustments based on feedback should be tested via **A/B testing**.

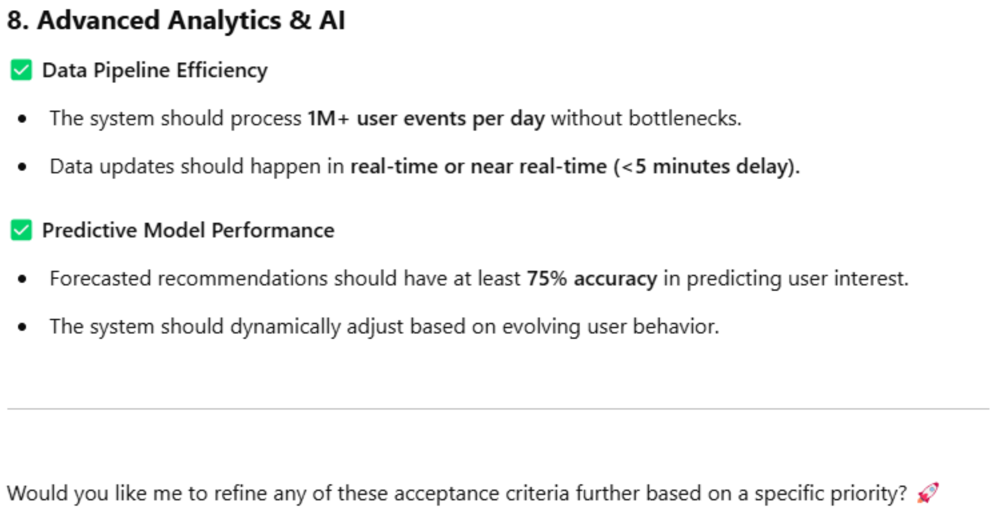
7. Integration & Testing

✅ **Machine Learning Model Integration**

- The recommendation engine should update **without downtime**.
- New models should be deployed via **CI/CD pipelines**.

✅ **QA Testing for Accuracy**

- At least **90% of recommendations** should be relevant based on test cases.
- Testing should cover **multiple user segments** (e.g., frequent shoppers, new users).



8. Advanced Analytics & AI

✅ **Data Pipeline Efficiency**

- The system should process **1M+ user events per day** without bottlenecks.
- Data updates should happen in **real-time or near real-time (<5 minutes delay).**

✅ **Predictive Model Performance**

- Forecasted recommendations should have at least **75% accuracy** in predicting user interest.
- The system should dynamically adjust based on evolving user behavior.

Would you like me to refine any of these acceptance criteria further based on a specific priority? 🚀

By following these steps, you have successfully analyzed requirements and generated detailed acceptance criteria using ChatGPT 4 to enhance the Software Testing Life Cycle.