

Welcome to ShopWise Solutions

ShopWise Solutions is an innovative and fast-growing e-commerce company based in Austin, Texas, USA. Our online platform hosts a wide range of consumer products, spanning electronics, apparel, home goods, and much more. ShopWise Solutions has built a reputation for exceptional customer experience, streamlined order fulfillment, and a diverse catalog of quality products.

Problem Description

ShopWise Solutions is now expanding into the realm of AI-driven customer service and is in need of a new AI-powered product support assistant for our online store. This assistant will serve as the first line of support for customer inquiries regarding products, orders, returns, and refunds, all by interfacing with our existing e-commerce database.

The assistant should be able to handle customer inquiries related to products and orders. It should be capable of natural language understanding, database integration, personalized responses, multi-turn dialogues, and avoidance of hallucinations. Additionally, it should be able to manage orders and track their status.

Requirements

- ShopAssistant AI capabilities:
 - Natural Language Understanding
 - Database Integration
 - Personalized Responses
 - Multi-Turn Dialogues
 - Avoidance of hallucinations
 - Order Management

Submission

Submissions should be made via a simple url/website that can demonstrate the assistants/agent's capability. That should be accompanied by a corresponding github repository that contains the source code for the project so judges can match the functionalities observed to the code.

Submission guidelines:

1. **Showcase Website and GitHub Repository:**

Participants should deploy their AI-powered customer support assistant on an accessible website that demonstrates the assistant's functionality. This website must be accompanied by a GitHub repository containing the complete implementation. The repository should be publicly accessible and the corresponding github repository should accompany it.

2. **Alternative Submission (if Website isn't Infeasible):**

If deployment on a website is not possible due to technical constraints, participants must submit a thoroughly organized GitHub repository as an alternative. This repository should demonstrate the implementation with additional rigor in documentation, code comments, and a clear README that guides users through the features and functionalities of the assistant. Comprehensive instructions for testing the solution locally or on a personal server should be included, along with clear usage examples to showcase the assistant's capabilities.

In both cases, it is preferred to have:

- A clear explanation of the tech stack chosen and architecture design.
- A set of instructions for installation and running the solution.
- Documentation that explains key features, code structure, and how the solution meets the evaluation criteria.

However, if the website with the solution works, we will use the github repo only as a reference, while in the context of item 2, these points will be weighted as criteria.

Evaluation Criteria

The assistants produced by the participants should be evaluated based on an (ideally) runnable solution, deployed as a simple accessible website (accompanied by a github repository) where judges will evaluate the implementation according to the following:

Performance Metrics:

- **Accuracy** based on a test dataset composed of standard prompts (a set of examples will be provided as reference for the contestants)

- **Avoidance of common pitfalls** such as:
 - Hallucinations (not providing fake information about products)
 - Not being biased towards certain products (in scenarios where users might repetitively request for information about a general group of products)
- **Response Time** (how smooth is the user experience and latency of responses, this will be evaluated subjectively and measures will be implemented by our judges to mitigate internet connection effects)
- **Overall Solution**
 - Which tech stack/framework they picked? With reasoning behind it
 - The architecture implemented
 - Final Presentation
- **Rubric (1-5)**
 - **1 Unsatisfactory: Application produces inaccurate, low quality output and is very slow**
 - **2 Needs Improvement: Application produces some accurate output but is of low quality and very slow**
 - **3 Satisfactory: Application produces fairly accurate output but is of average quality and somewhat slow**
 - **4 Proficient: Application produces mostly accurate and high quality output with lower latency**
 - **5 Exemplary: Application is exceptional, innovative, and surpasses expectations.**

Evaluation Guidelines and Example Prompts

To standardize the evaluation process as much as possible, we'll have a set of evaluation scenarios that all judges will consider when performing individual evaluation of submissions.

Evaluation Scenarios

1. **Context Understanding** - Questions that test the model's ability to understand context, not just retrieve information.
 - a. Prompt example 1: "I see there are some Sony TVs in your catalog. Can you compare the features and prices between the Sony KD75XF8596BU and other TV models you have?"
 - b. Prompt example 2: "Looking at the ratings distribution across different product categories, which category would you recommend for the most reliable purchases based on customer satisfaction?"
 - c. Prompt example 3: "I notice you have mobile phones and digital cameras. For someone interested in photography, would you recommend the Sony Xperia XA2 Ultra or the Pentax K-1 camera? Please explain the pros and cons of each for photography."

- d. Prompt example 4: "Based on the stock quantities and prices shown in your inventory, which TV models offer the best value for money while still being readily available?"
- 2. **Product Comparison** - Questions that test model's ability to compare prices of at least 2 different products
 - a. Prompt example: "What's the difference between Product A and Product B?"
- 3. **Alternative Suggestions** - Questions that test model's ability to suggest or recommend products matching a given criteria
 - a. Prompt example: "I'm looking for a laptop under \$1,000 with at least 16GB RAM."
- 4. **Product Availability** - Questions that test the model's ability to assess correctly the availability of products, as well products that match given criterias
 - a. Prompt example: Is Product C available in red?
- 5. **Nested Requests** - Questions that test the model's ability to answer complex nested requests, containing 2 or more nested specifications
 - a. Prompt example: "Can you recommend a smartphone with a good camera and tell me if it's compatible with wireless charging?"
- 6. **Order Status Check:**
 - a. Prompt example: *"Can you tell me the status of my order with Order ID X?"*
- 7. **Tracking and Shipping Information:**
 - a. Prompt Example: "When is my order with Order ID X expected to be delivered?"
- 8. **Return Eligibility:**
 - a. Prompt Example "Is my order with Order ID 54321 eligible for return?"
- 9. **Order History:**
 - a. Prompt Example "Can you list my previous orders? My customer id is X."
- 10. **Shipping Delay Inquiry:**
 - a. Prompt Example: "My order with Order ID 24680 seems delayed. Can you provide an update on when it might ship?"
- 11. **Order Assistance for Multiple Items:**
 - a. Prompt Example: "I ordered multiple items in Order ID 16180. Can you check if each item has shipped?"

Dataset Overview

2 datasets in the form of .csv files will be provided:

- 1. Product information dataset: [synthetic-product-data.csv](#)
- 2. Product order information dataset: [synthetic-orders-data.csv](#)

Product information dataset: synthetic-product-data.csv

This dataset contains product information for various electronic appliances and devices. Each record represents a unique product with the following key attributes:

- **Product ID:** Unique identifier for each product !!!!
- **ProductName:** Full name/description of the product
- **Category:** Product category (e.g., TVs, Fridges, Mobile Phones)
- **Price:** Product price in currency units
- **Rating:** Customer rating on a scale of 1-5
- **Description:** Technical description of the product
- Other metadata fields

The dataset covers 10 main product categories:

- Fridges
- TVs
- Mobile Phones
- Digital Cameras
- Fridge Freezers
- Dishwashers
- CPUs
- Freezers
- Washing Machines
- Microwaves

Product order information dataset: synthetic-orders-data.csv

This dataset contains order transaction records for the products. Each record represents a single order with the following key attributes:

- **OrderID:** Unique identifier for each order
- **Product ID:** Links to the product in the products dataset !!!!
- **CustomerID:** Identifier for the customer making the order
- **OrderStatus:** Current status of the order (e.g., Shipped, Delivered, Cancelled, Pending)
- **ReturnEligible:** Boolean indicating if the order can be returned
- **ShippingDate:** Timestamp of when the order was/will be shipped

The orders appear to span across late 2024, with various order statuses tracking the customer purchase lifecycle.

Relationship Between Datasets
The two datasets are related through the !!!! Product ID field !!!!, allowing you to join order information with detailed product data. This enables analysis of ordering patterns across different product categories, price points, and customer ratings.