Vietnam National University- Ho Chi Minh City

University of Science

Faculty of Information Technology

HW03. DATA GENERATION & SCENARIO TESTING

Course Software Testing
Class 22KTPM3
Group Group10
Student **22127225 – Trần Thị Thiên Kim**

Ho Chi Minh City, 2025

Table of Contents

1 Task Allocation

Student ID	Student	Selected Table
20127233	Huỳnh Thế Long	- Invoice Table - Invoice Item Table
22127225	Trần Thị Thiên Kim	- Categories Table - Contact Request Table
22127312	Nguyễn Thị Yến Nhi	- Contact Requests Replies Table - Favorites Table
22127316	Nguyễn Ngô Như Ngọc	- Products Table - Users Table

2 Overview of Data Generation

This report documents the comprehensive process of generating realistic and structured test data for two critical database tables in the ToolShop application: `categories` and `contact_requests`. These tables were selected based on their importance to the system's core functionalities—product categorization and customer communication.

The goal of this data generation task was to create datasets that closely simulate real-world usage, ensuring that system testing reflects actual user behavior and operational scenarios. Each table was populated with at least 500 rows of valid and meaningful entries. Randomized data was generated with logical constraints and relationships to ensure referential integrity (e.g., parent-child relationships in categories).

By preparing large volumes of valid data in advance, the system could be tested at scale for performance, validation rules, and feature correctness. The generated datasets were also utilized as the foundation for scenario-based testing in later phases of this assignment.

Key aspects of this process include:

- Selecting appropriate tools (ChatGPT) for rapid and flexible data generation.
- Defining clear rules and constraints for each data field.
- Verifying consistency, format, and edge-case handling of generated data.
- Ensuring timestamp accuracy for created/updated fields to reflect realistic data lifecycles.

This approach to data generation not only accelerates the test preparation phase but also improves test reliability by using datasets that align with the actual structure and logic of the ToolShop application.

3 Selected Tables and Field Rules

3.1 Categories Table

- `id`: Integer — auto-increment from 1 to 500.
- `name`: String — meaningful product category names (e.g., "Power Tools", "Hand Tools").

- `slug`: String — generated from the `name`, formatted for URLs (e.g., "power-tools").
- `parent_id`: Integer (Nullable) — 80% of rows randomly assigned to an existing `id`, the rest NULL.
- `created_at`: Datetime — random timestamp between 2023-01-01 and 2025-06-15.
- `updated_at`: Datetime — same as `created_at` or later by up to 30 days.

3.2 Contact Requests Table

- `id`: Integer — auto-increment from 1 to 500.
- `name`: String — full names (e.g., "John Doe", "Alice Nguyen").
- `email`: String — valid email format (e.g., "alice@example.com").
- `subject`: String — selected from ["Product Inquiry", "Support Request", "Order Issue"].
- `message`: Text — short and meaningful sentences that reflect user requests (e.g., "Can I get more details about drill X?").
- `created_at`: Datetime — random timestamp between 2023-01-01 and 2025-06-15.
- `updated_at`: Datetime — same as `created_at` or later by up to 30 days.

4 Selected Tool: ChatGPT (OpenAI)

To generate the datasets, I used **ChatGPT (OpenAI)**, a generative AI tool capable of producing structured and semantically meaningful data based on natural language prompts.

Tool Information

- **Tool Name:** ChatGPT
- **Model:** GPT-4
- **Access Method:** `https://chat.openai.com`

5 Prompts Used for AI-Based Data Generation

To generate meaningful and realistic datasets for the `categories` and `contact_requests` tables in the ToolShop application, I utilized ChatGPT (OpenAI). The following prompts were crafted with clarity and specificity to ensure that the generated data adhered to the schema and business logic of the system:

5.1 Prompt for `categories` Table

```
"You are a professional data generator. Generate 500 rows of data for the
'categories' table of an e-commerce tool shop system. Each row should include:
id (auto-increment), name (tool-related category), slug (URL-friendly version
of name), parent_id (nullable, foreign key to id), created_at, and updated_at
(in valid ISO datetime format, updated_at is the same or later than created_at)."
```

5.2 Prompt for `contact_requests` Table

```
"Generate 500 rows of realistic customer contact requests for an online tool
shop database. Each record should contain: id (auto-increment), name (realistic
full name), email (valid format), subject (e.g., Product Inquiry, Support Request,
Order Issue), message (related sentence or paragraph), created_at, and updated_at
(updated_at should be the same as or later than created_at). Output should
be tabular and ready for CSV export."
```

5.3 Prompt Refinement and Validation

To ensure quality and usability:

- I reviewed the initial AI outputs to eliminate meaningless, repetitive, or out-of-domain content.
- I manually validated datetime values and formatting.
- I adjusted prompts iteratively when data did not meet expectations (e.g., duplicate slugs, invalid emails).
- I used **regex checks** and preview tools in CSV editors to finalize formatting.

The final data sets, once reviewed and refined, were exported to CSV format and imported into the database for testing purposes.

6 Explanation of Process and Steps

To ensure meaningful, valid, and schema-compliant test data, I followed a structured and systematic data generation workflow. Below are the detailed steps taken throughout the process:

1. **Understanding the Database Schema:** I began by analyzing the official ToolShop database schema available in the GitHub repository `https://github.com/testsmith-io/practice-software-testing`

   I identified the relevant tables (`categories` and `contact_requests`) and their fields, including data types, constraints, and relationships (e.g., foreign key from `categories.parent_id` to `categories.id`).

2. **Designing Prompts for AI-Based Generation:** Based on the schema, I crafted detailed prompts for ChatGPT. These prompts included:

   - Field names and their expected data types.

- Business logic (e.g., realistic tool category names, valid email formats).
- Structural rules (e.g., `parent_id` must reference a valid `id`, timestamps must be chronologically consistent).

For example, I instructed the model to generate tool-related categories, convert them into slugs, and assign `parent_id` values with a certain probability to form a category tree.

3. **Generating and Reviewing the Data:** ChatGPT generated the requested datasets for each table in batches. I reviewed the output for the following aspects:

   - Format correctness (e.g., email, slug, timestamps).
   - Referential integrity (e.g., ensuring `parent_id` values are valid).
   - Realism (e.g., natural-sounding names, messages, subjects).

   Minor adjustments were made manually where necessary to correct edge-case inconsistencies or regenerate subsets of data.

4. **Exporting Data to Structured Format:** Once validated, the data was exported to CSV format using Excel and verified for compatibility with database import tools. For columns such as `created_at` and `updated_at`, I ensured the timestamps reflected realistic lifecycle intervals, where `updated_at` was equal to or slightly later than `created_at`.

5. **Data Readiness for Testing:** The finalized datasets were loaded into MySQL and visually inspected using DBMS tools (e.g., phpMyAdmin) to confirm data quality, adherence to constraints, and readiness for use in scenario-based testing.

7 Screenshots of Tool in Use

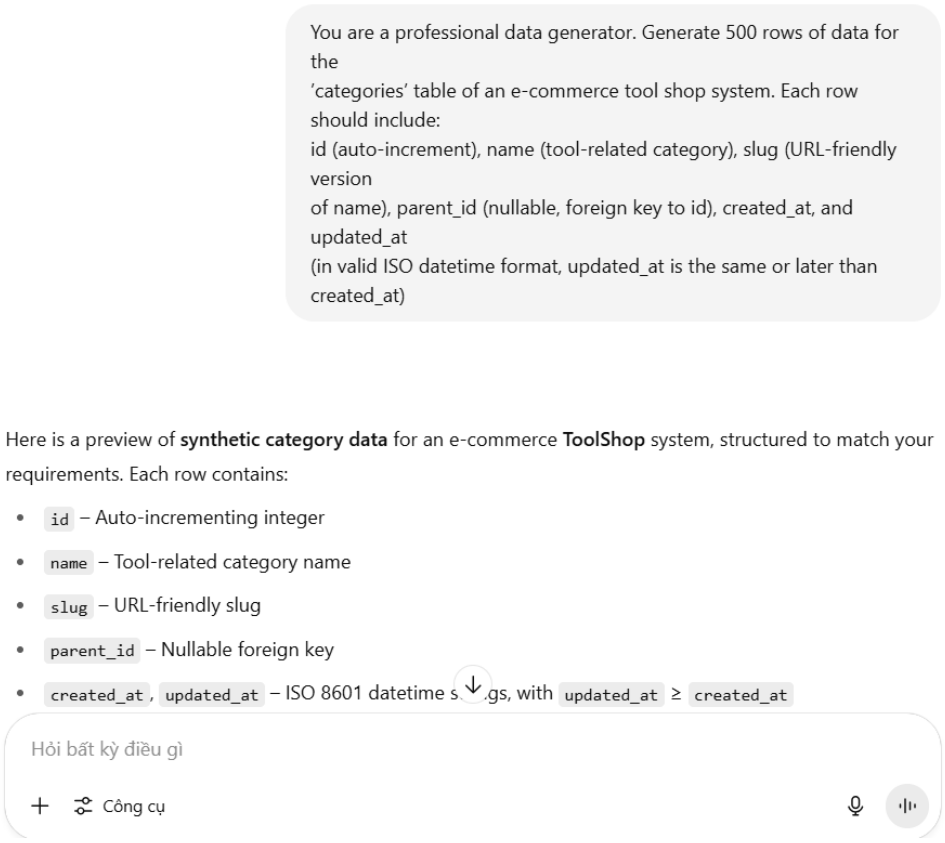

Figure 1: Prompting ChatGPT to generate 500 rows for the Categories table

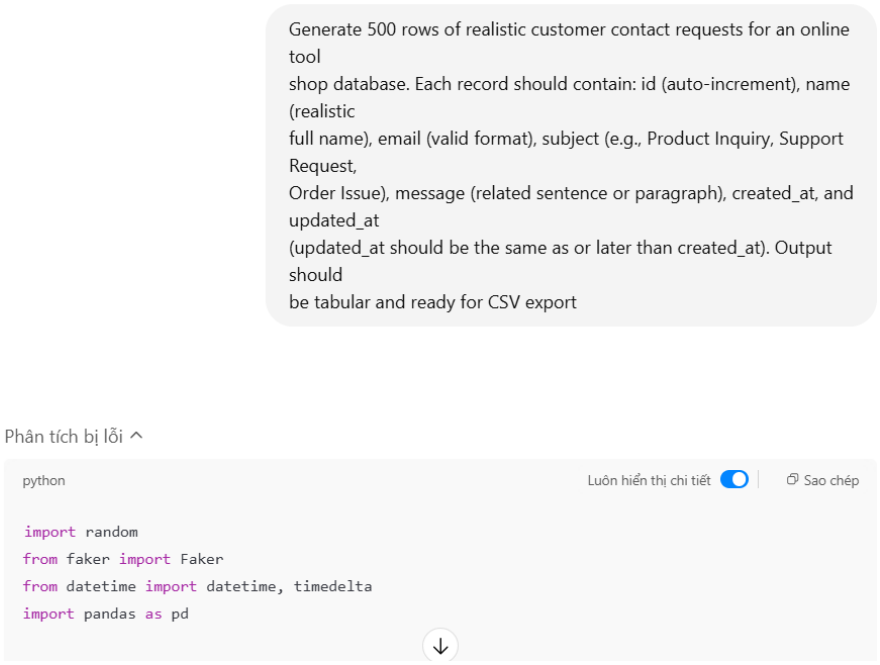

Figure 2: Prompting ChatGPT to generate data for Contact Requests with meaningful messages

8 Sample Data Output

8.1 Sample from Categories Table

Sample Output (First 10 rows)

csv — Sao chép — Chỉnh sửa

```
id,name,slug,parent_id,created_at,updated_at
1,Hand Tools,hand-tools,,2022-01-15T09:24:12,2022-03-10T10:55:32
2,Power Tools,power-tools,,2022-02-03T14:32:17,2022-07-21T16:43:09
3,Gardening Tools,gardening-tools,,2021-09-08T08:41:51,2022-01-22T10:29:01
4,Measuring Tools,measuring-tools,,2020-05-19T11:19:00,2020-08-05T12:27:45
5,Automotive Tools,automotive-tools,,2022-06-27T13:08:05,2022-06-28T09:12:07
6,Electric Hand Tools,electric-hand-tools,1,2022-09-05T15:55:42,2023-01-01T13:42:12
7,Heavy-Duty Power Tools,heavy-duty-power-tools,2,2023-02-18T11:23:32,2023-02-22T16:40:14
8,Manual Gardening Tools,manual-gardening-tools,3,2021-12-11T07:40:25,2022-01-19T15:21:30
9,Precision Measuring Tools,precision-measuring-tools,4,2020-07-23T09:10:03,2021-03-01T10:30:44
10,Socket Wrenches,socket-wrenches,1,2021-03-20T11:25:45,2022-04-11T13:11:00
```

Figure 3: Sample from Categories Table

8.2 Sample from Contact Requests Table

Here's a **sample preview of 10 realistic customer contact request records** for an online tool shop. The data is structured and ready for CSV export:

csv Sao chép Chỉnh sửa

```
id,name,email,subject,message,created_at,updated_at
1,Johnathan Stewart,j.stewart@example.com,Product Inquiry,"Hi, I'm interested in your new line of
2,Amanda Lewis,amanda.lewis84@example.com,Support Request,"Hello, my hammer drill stopped working
3,David Kim,david.kim91@example.com,Order Issue,"I placed an order last Friday, but haven't receiv
4,Natalie Chen,natalie.chen22@example.com,Return/Refund,"The tool set I received has a broken piec
5,Thomas Nguyen,tomnguyen@example.com,Shipping Delay,"My order #58312 was supposed to arrive 3 day
6,Sophia Martinez,s.martinez@example.com,Technical Question,"Does the 18V impact driver support ur
7,Brian O'Connor,b.oconnor@example.com,General Feedback,"Just wanted to say your customer service
8,Emily Davis,emily.davis98@example.com,Order Issue,"I received the wrong item in my shipment. I c
9,Michael Scott,m.scott@example.com,Product Inquiry,"Do you have a bulk discount for workshop hanc
10,Grace Patel,grace.patel@example.com,Support Request,"My angle grinder makes a loud clicking sou
```

Figure 4: Sample from Contact Requests Table

9 Conclusion

Using ChatGPT enabled me to efficiently generate high-quality test data that mirrors realistic user input. The generated datasets were validated for consistency, uniqueness, and semantic relevance. These datasets support effective scenario testing and reduce the need for manual data entry or custom scripting.