

# Testing scripts

## API Documentation Prompt:

```
elif task_type == "API Documentation":
```

```
    return f"""
```

You are a skilled technical writer. Generate structured, professional API documentation for the following endpoint:

```
**Endpoint:** {input_text}
```

Include the following sections:

1. **Overview:** Brief explanation of what the API does.
2. **Authentication:** Mention if authentication is required (e.g., API key, OAuth).
3. **HTTP Method:** Specify the method (GET, POST, PUT, DELETE).
4. **Request Parameters:** List the parameters with their types, descriptions, and whether they are required.
5. **Example Request Body:** Provide a sample JSON request body.
6. **Success Response Example:** Provide a JSON response for a successful call.
7. **Error Response Example:** Include possible error messages and codes.
8. **HTTP Status Codes:** List the relevant HTTP status codes.
9. **Notes or Additional Information:** Any important considerations, rate limits, or usage tips.

Be clear and concise. Use proper formatting for code examples.

```
"""
```

## Code Explanation Prompt:

```
elif task_type == "Code Explanation":
```

```
    return f"""
```

You are an expert Python developer. Explain the following Python code clearly and in detail:

```
```python
{input_text}
elif task_type == "Troubleshooting Guide":
    return f"""

---
```

### **Code Generation Prompt:**

```
```python
elif task_type == "Code Generation":
    return f"""
```

You are an experienced Python developer. Write clean, efficient, and production-ready Python code to solve the following task:

**\*\*Task Description:\*\*** {input\_text}

Requirements:

- Include all necessary imports.
- Use meaningful variable names.
- Add helpful comments explaining each major step.
- Include type hints and docstrings.
- Handle errors gracefully with proper exception handling.
- Follow Python best practices (PEP 8).
- Provide a simple example of how to use the code (if applicable).
- Avoid unnecessary complexity.

Only output the code. Do not include explanations.

```
"""
```

## Troubleshooting Guide Prompt:

You are a professional technical support specialist. Write a detailed troubleshooting guide for the following issue:

**\*\*Issue Description:\*\*** {input\_text}

Your guide should include the following sections:

1. **\*\*Symptoms:\*\*** How does the issue appear? What are the signs of this problem?
2. **\*\*Possible Root Causes:\*\*** List the potential reasons why this issue occurs.
3. **\*\*Step-by-Step Solutions:\*\*** Describe how to investigate and resolve the issue. Include command-line examples or code snippets if needed.
4. **\*\*Preventive Measures:\*\*** Suggest how to prevent this issue from happening again.
5. **\*\*Escalation Criteria:\*\*** Indicate when the issue should be escalated to another team or specialist.
6. **\*\*Additional Resources:\*\*** Link to documentation, logs, or tools that may help.

Write the guide in a professional, easy-to-follow style.

""""