```
# prompts.py

# Analyze the user's idea to extract key concepts, requirements, and desired outcomes
IDEA_INTAKE_PROMPT = """
Analyze and expand upon the user's idea, extracting key concepts, requirements, and desired outcomes. Represent the user's idea in a highly detailed structured format, including key features, constraints, and desired outcomes. Idea: {idea}
"""

# Develop a high-level plan for the codebase, including directory structure and file organization
CODEBASE_PLANNING_PROMPT = """
Develop a high-level plan for the codebase, including directory structure and file organization. Try to keep the number of files to a maximum of 7 for efficiency, and make sure there is one file that ties it all together for the user to run all the code. Design the software architecture to determine the overall structure
of the codebase based on the following requirements: {requirements}
"""

# Translate the high-level codebase plan into specific, actionable development tasks
TASK_PLANNING_PROMPT = """
Translate the high-level codebase plan into specific, actionable development tasks. For each identified component or feature in the plan, create a detailed task that includes necessary actions, technologies involved, and expected outcomes. Structure each task to ensure clear guidance for the development team or subsequent AI code generation agents.

High-Level Codebase Plan: {codebase_plan}
```

Guidelines for Task Planning:

- Identify distinct components or features from the codebase plan.

- For each component or feature, specify the development tasks required.

- Include any imports, technology stacks, frameworks, or libraries that should be used.

- Detail the expected outcomes or objectives for each task.

- Format the tasks as structured data for easy parsing and automation.

"""

Generate individual code files based on the detailed task descriptions

FILE_WRITING_PROMPT = """

Generate individual code files based on the codebase plan. Write code in the specified programming language using programming language

generation techniques. For each file required by the project,

please include the one-word file name wrapped in tags <!--START_FILE_PATH--> and <!--END_FILE_PATH-->, followed by the file content wrapped in

<!--START_CONTENT--> and <!--END_CONTENT--> tags. Ensure each file's details are clearly separated. Here are the details: {details}

"""

Analyze the generated code for correctness, efficiency, and adherence to best practices

CODE_REVIEW_PROMPT = """

Analyze the generated code for correctness, efficiency, and adherence to best practices. Meticulously review the codebase to find any errors, bugs, missing imports, improper integration,or

broken logic. Output a detailed list of improvements for our engineering team including all issues (ESPECIALLY import issue) and how to fix them. Here is the code: {code}.
"""

# Refactor the generated code to improve its structure, maintainability, and extensibility
CODE_REFACTORING_PROMPT = """
Given the code provided, refactor it to improve its structure, maintainability, and extensibility. Ensure the refactored code adheres to best practices and addresses the specified areas for improvement.

When presenting the refactored code, use the same format as in the file writing step: Wrap the one-word file name with <!--START_FILE_PATH--> and <!--END_FILE_PATH--> tags, and enclose the file content with <!--START_CONTENT--> and <!--END_CONTENT--> tags. ENSURE that the end of your output contains an "<!--END_CONTENT-->" tag. This format will facilitate direct parsing and file saving from the output.

Areas to improve: {improvements}

The code to refactor:
{code}

Note: The expectation is that the refactored code will be structured and tagged appropriately for automated parsing and saving as individual code files.
"""

```
# Push the final codebase to a GitHub repository, managing code changes and revisions
GITHUB_PUSH_PROMPT = """
Push the final codebase to a GitHub repository. Manage code changes and maintain a history of revisions using version control integration. Here are the final changes: {changes}
"""
```