The original code had a code smell of repetitive code and lack of organization. It prompted users multiple times for input filenames and performed file operations without encapsulating them into reusable functions or a well-structured class. This made the code harder to read, maintain, and reuse, violating the DRY (Don't Repeat Yourself) principle.

The refactored code significantly addressed these issues. It introduced a GenerateHTML class that encapsulates the file operations, enabling reusability and clearer code organization. Functions like get\_user\_input were created to handle user prompts, promoting modularity and reducing redundancy. The GenerateHTML class now initializes with input and output filenames and includes a generate method responsible for reading content from the input file and writing it to the output file. The main function orchestrates the process, making the code more readable and maintaining the original functionality.

This refactoring not only enhanced the code's readability but also improved its maintainability and reusability. Now, adding new features or making changes would be easier without duplicating code, as the file operations are encapsulated within the class methods. Additionally, the use of functions for user input ensures consistent and reusable behavior, promoting a more structured and organized codebase for future modifications.