The following improvements were implemented:

Separation of Concerns:

Previously, the Main method in the Program class handled both user input and API interactions. This adds tight coupling in the code and would benefit from adding separate classes to handle individual tasks.

To address this, the codebase was refactored to separate these concerns into different classes: ConsoleAppHandler and IJokeService implemented by JokeService. ConsoleAppHandler is responsible for interacting with the console user, getting the input, validating the input, and then delegating the requests to the respective APIs. JokeService is responsible for calling the jokes API to retrieve random unique jokes and categories.

After making these changes, the Main method in the program.cs file is only responsible for dependency injection setup and app initialization.

Dependency Injection:

Dependency Injection was introduced to inject the IJokeService class into the ConsoleAppHandler class. This improved design promotes loose coupling between components and enables easier testing and flexibility in the codebase. This makes the codebase much more extensible allowing for easier integration of new features.

User Input Validation:

Enhanced user input validation was implemented to ensure that only valid inputs are accepted. For example, when specifying a category for jokes, the application prompts the user to re-enter the category if an invalid(cannot be a number) or non-existent category (checks against a list of valid categories) is provided. This improves the overall user experience and prevents errors.

In addition, ConsoleAppHandler class has a separate method called "ValidateNumberOfJokes" to validate the numeric input to retrieve a number of random unique jokes. It allows only the numbers between 1 and 9. If the user inputs any other value, the user will be prompted to correct the input and enter only a number between 1 and 9. The "GetCategories" method in the JokeService class has an input validator to check if the string is non-null and contains only characters by validating it with a regex.

Error Handling:

Improved error-handling mechanisms were implemented in the JokeService class to gracefully handle scenarios such as failed API requests and insufficient jokes in a category. Exceptions are now being handled inside the try-catch block. This ensures that the application remains robust and provides meaningful feedback to users in case of errors.

Retry Logic: It is possible that some of the categories may not have the requested number of unique jokes. This could cause the method to loop indefinitely attempting to find a unique number of jokes per the requested. This edge case has been gracefully handled by adding maximum retries for the API call. If the requested number of unique jokes is not found, it returns the maximum number of unique jokes available in the requested category and sends feedback to the console stating the same.

Improved Console Output:

The console output was refined to provide clearer instructions and error messages to users. Added extra functionality for the user to exit or terminate the app by inputting "e" or "exit" while running the app.

Type-Safety: The older version of the app uses JsonConvert.DeserializeObject<dynamic> to parse the Http response. Instead of deserializing the JSON string response to a dynamic object, a new response type called "JokeResponse" was added to provide type safety. This enables compile-time type checking instead of dealing with a

dynamic unknown structure.
(JsonConvert.DeserializeObject<JokeResponse>)

URL Concatenation: The URL concatenation in the `GetRandomJokes` method of JokeService could result in malformed URLs, especially if the `_url` field is not properly formatted while handling query parameters. The URLs are now constructed using `UriBuilder.`

Test Coverage:

Unit tests were written using NUnit to verify the behaviour of individual methods in the JokeService class. Test coverage ensures that the code functions correctly under various conditions and helps identify and fix bugs more effectively.

Other Bug Fixes:

Renamed JsonFeed.cs class to JokeService to make it more suggestive. Made the endpoints a global type declaration instead of allowing the caller to pass the URLs to the JokeService constructor.

ConsolePrinter.cs class seems unnecessary as it is simply writing output to the console which can be achieved by simply calling the "console.WriteLine()" directly. Removed the usage of this class in the codebase.