Building a command-line weather checking application in Python involves several steps. Here’s a high-level outline of how you can approach this task:

1. **Set up the Project:**

- Create a new directory for your project.
- Initialize a Python virtual environment to manage dependencies.
- Install the required packages, such as `requests` for making HTTP requests.

2. **Create a Configuration File:**

- Create a configuration file to store your API key for the Weather API (you can sign up on https://www.weatherapi.com/ to get an API key).

- Store other configuration settings as well.

3. **Implement Command-Line Interface (CLI):**

- Use a library like `argparse` to create a user-friendly command-line interface.
- Define options for checking weather by city name, managing a favourite list of cities, and setting auto-refresh intervals.

4. **Make API Requests:**

- Use the `requests` library to make HTTP GET requests to the Weather API.
- Fetch weather data based on the user’s input (city name).

5. **Error Handling and Data Validation:**

- Implement error handling to deal with network issues and invalid user input.
- Validate user input to ensure it matches the expected format.

6. **CRUD Operations for Favourite Cities:**

- Create functions for adding, listing, updating, and deleting favourite cities.
- Store this data in a local file or a database.

7. **Auto Refresh:**

- Implement an auto-refresh mechanism using a timer or a background thread to periodically update weather information.

8. **Documentation:**

- Create a clear and concise documentation file or readme that explains how to use your application.
- Include examples of commands and their expected outputs.

9. **Testing:**

- Write unit tests to ensure the functionality of your application.
- Test edge cases and error scenarios.

10. **Final Touches:**

- Ensure the code is well-structured, organized, and follows best practices.
- Add comments and docstrings to make the code readable.