Overview

The **morning_greetings** assignment aimed to create a Python package that automates the process of sending personalized "Good Morning" messages to a list of friends. The package encompasses various modules for managing contacts, generating messages, simulating message sending, and logging sent messages. This report reflects my experience throughout the assignment, including challenges faced and how I overcame them.

Development Experience

Project Structure

I started by designing the structure of the package, ensuring modularity by breaking down functionalities into separate modules. The primary modules included:

- **contacts.py**: For managing contact information.
- **message_generator.py**: For generating personalized messages.
- **message_sender.py**: For simulating the sending of messages.
- **logger.py**: For logging sent messages.
- **contacts_manager.py**: For managing the contact list in a structured manner.

Implementation Challenges

1. **Module Imports**:
    - After moving **main.py** outside the **morning_greetings** package, I encountered issues with module imports. The error **ModuleNotFoundError** indicated that Python couldn't locate the modules. I resolved this by updating the import paths to reflect the new structure of the project.

2. **Error Handling**:
    - Implementing error handling across the modules was essential to ensure robustness. Initially, I overlooked error scenarios like sending messages to an empty contact list. After testing the code, I added exception handling to provide meaningful error messages to the user, enhancing user experience.

3. **Simulating Message Sending**:
    - The requirement to simulate message sending posed a challenge since I wanted to ensure the output mimicked a real-world application. I decided to print messages to the console while logging them for reference, allowing me to verify that the system worked as intended without requiring actual email integration.

Integration and Testing

Once the individual modules were implemented, I integrated them in **main.py** to create a cohesive application. I conducted several tests to ensure all components functioned correctly together. This phase was crucial for identifying any overlooked issues, such as incorrect data handling or logging problems.

Installation Guide

To install and use the **morning_greetings** package, follow these steps:

1. **Clone or Download the Repository**:
    - Use the following command to clone the repository:

```
“git clone https://github.com/yourusername/morning_greetings.git”
“cd morning_greetings”
```

2. **Install the Package**:

- Navigate to the directory containing **setup.py** and run the following command to install the package:
  “pip install -e .”

3. **Run the Application**:

- Once installed, you can execute the main script to send greetings:
  “python main.py”

Conclusion

Overall, the **morning_greetings** assignment was a rewarding experience that enhanced my understanding of modular programming in Python. It taught me valuable lessons in package structuring, error handling, and integration testing. The challenges faced, particularly with module imports and error management, provided opportunities for learning and growth.