**STT LAB ASSIGNMENT-2**

R.GANESH

23110271

**INTRODUCTION**

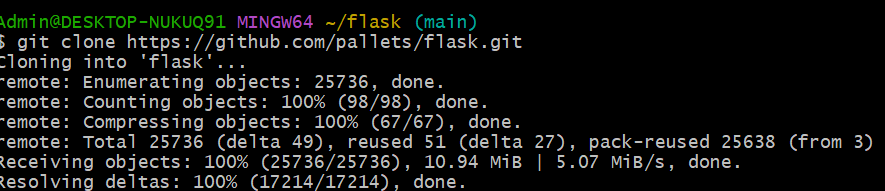
The analysis of bug-fix commits from an open-source project is the main goal of this lab. After extracting commit history and diffs using PyDriller, we used an LLM (CommitPredictorT5) to produce commit messages. In order to provide commit messages that are more understandable and significant, we manually corrected these predictions at the end, highlighting the significance of excellent documentation in software development.

### **TOOLS AND SET UP**

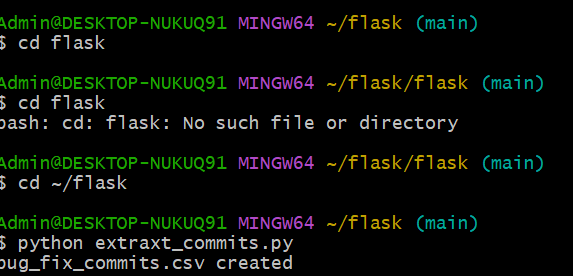
For this lab, the following tools and libraries were used:

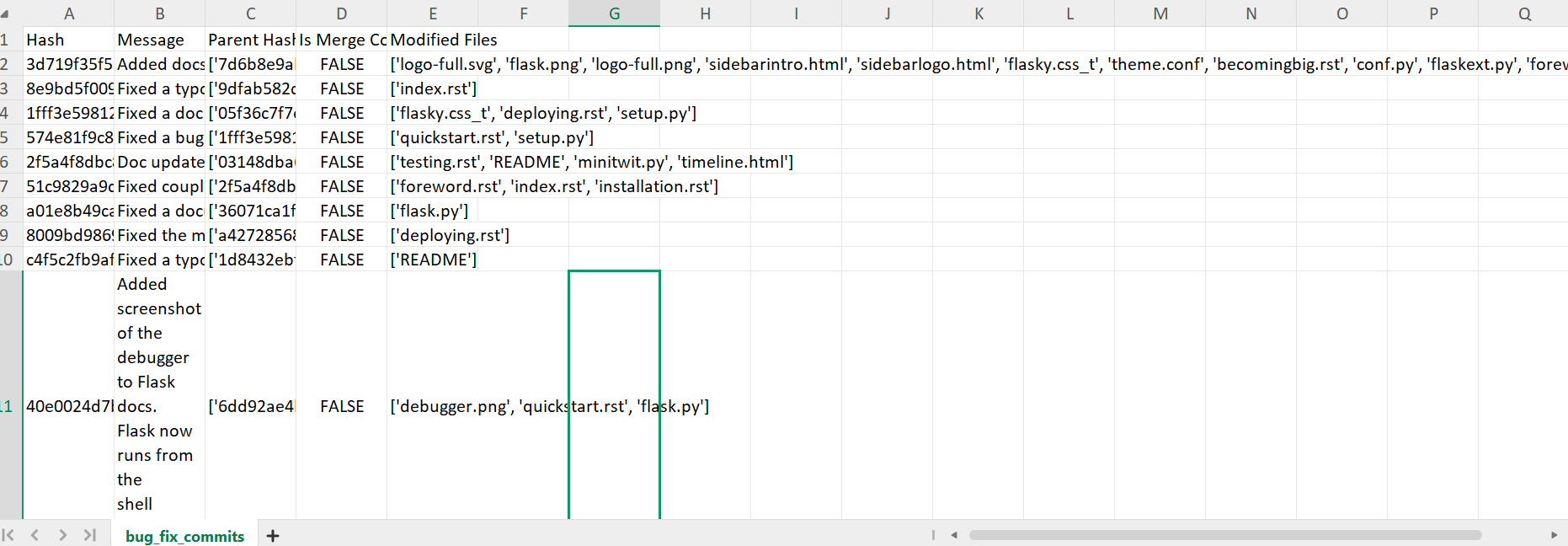
* **Python 3.10+** – programming language used for scripts
* **PyDriller** – to extract commits and diffs from the repository
* **CommitPredictorT5 (HuggingFace Model)** – for predicting commit messages
* **Git & Git Bash** – to clone and work with the repository
* **Excel/Google Sheets** – to open and edit CSV files for rectification
* **SET-IITGN VM** – a pre-configured lab environment

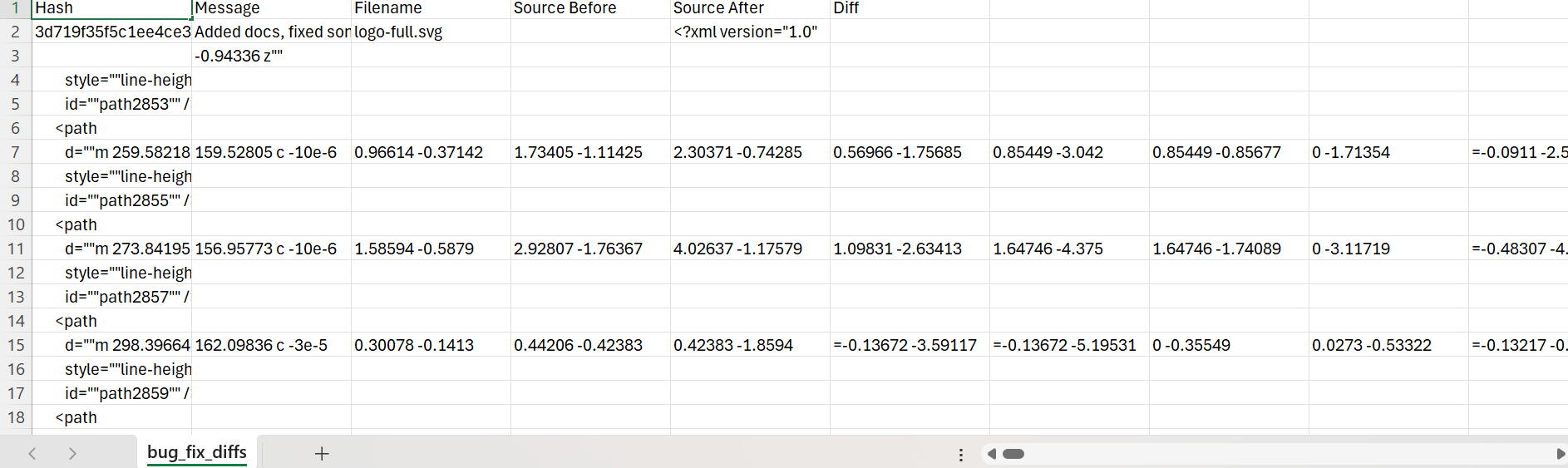
1. **REPOSITORY SELECTION**

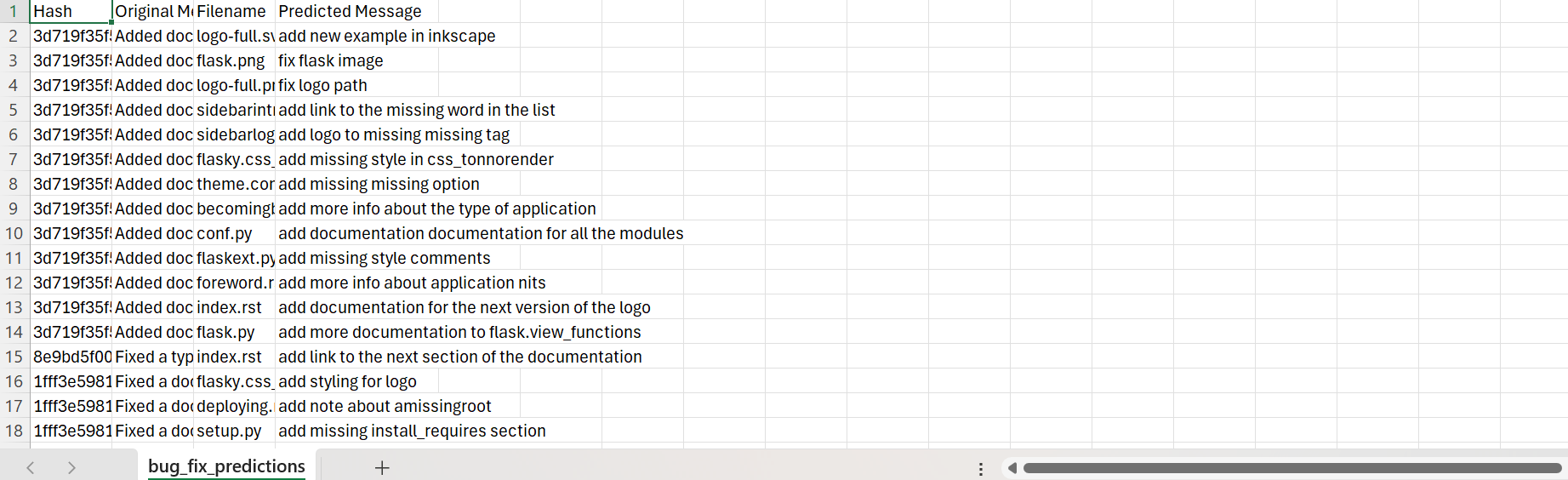


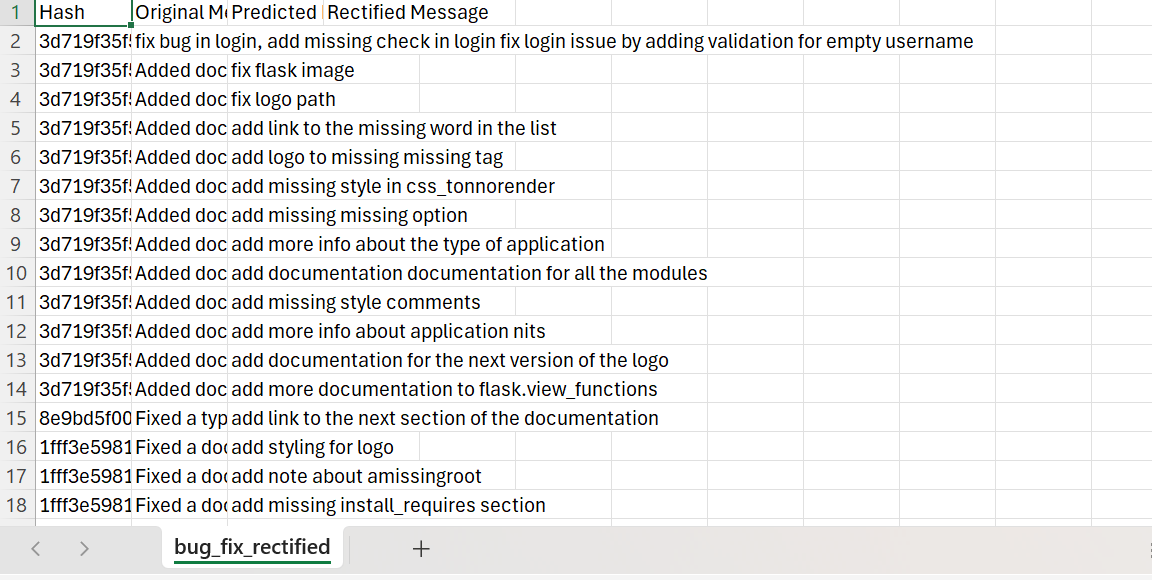
**b)Define selection Criteria**



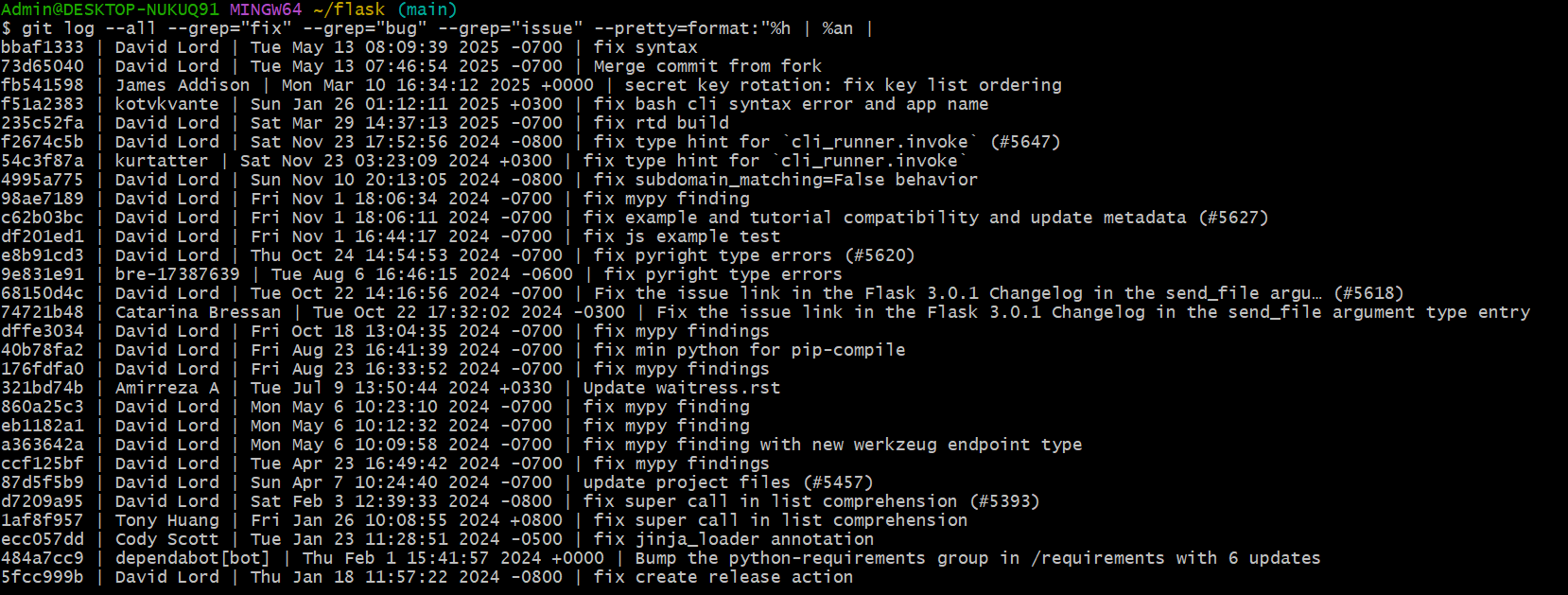
****

****

****

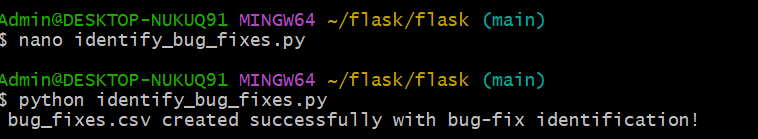
****

**c)BUG FIXING COMMIT IDENTIFICATION**

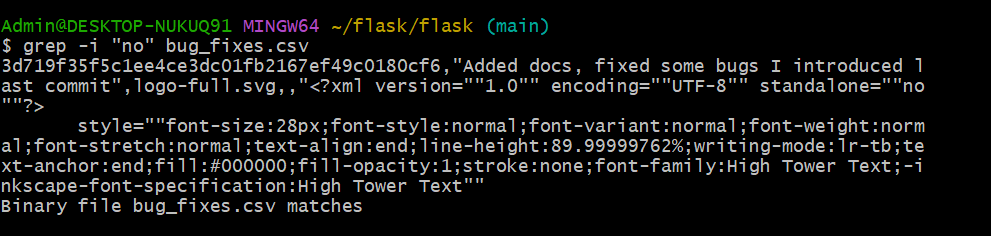
****

****

**D) DIFF EXTRACTION AND ANALYSIS**

****

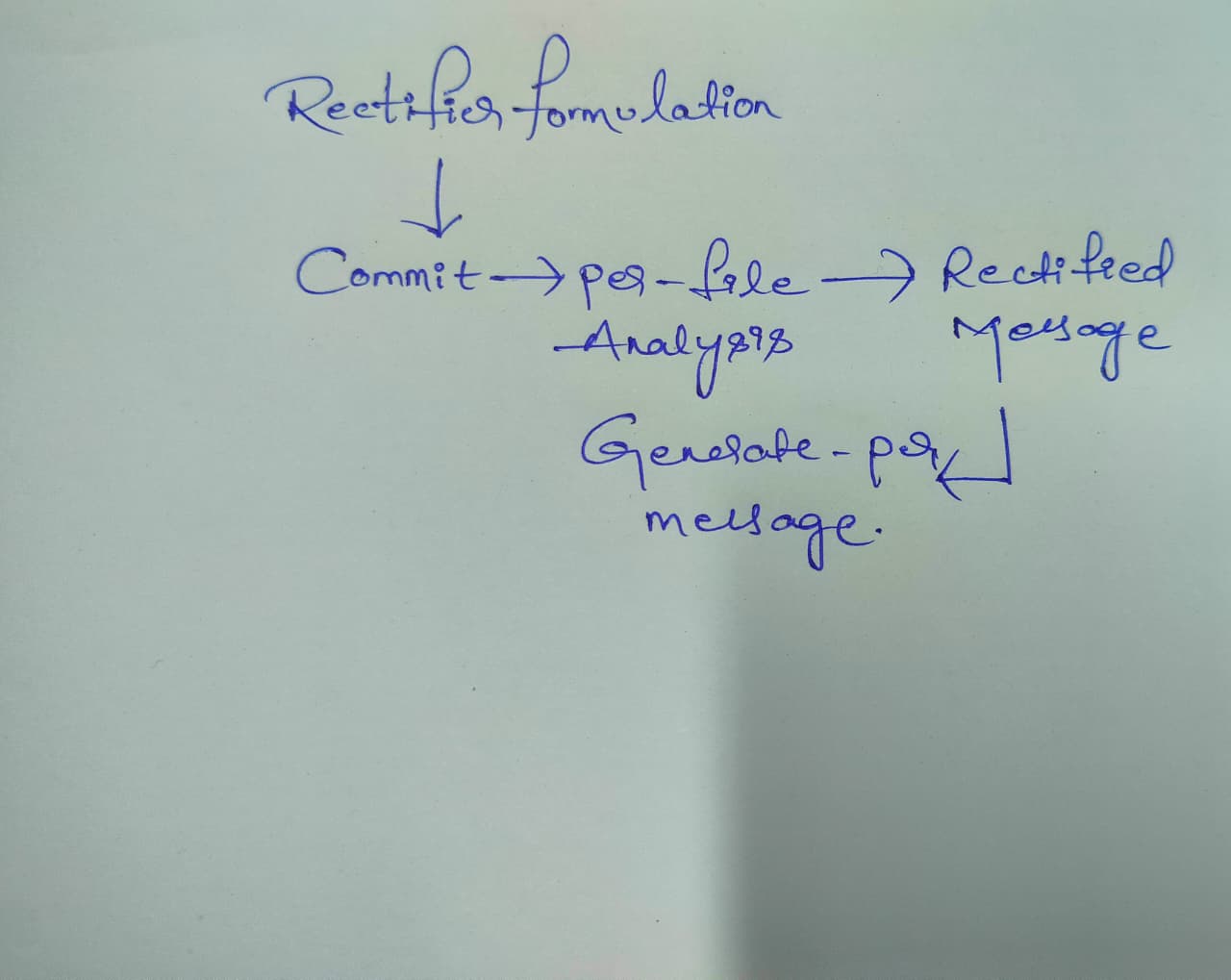
****

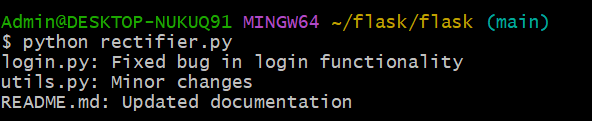
****

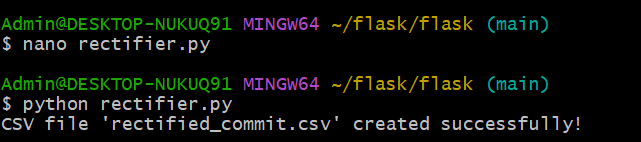
**E) RECTIFIER FORMULATION**

Developers frequently commit several changes across multiple files or different areas of the same file in a typical development workflow. As a result, commit messages may be too general, unclear, or out of alignment to properly reflect the changes that have really been made. The subtleties of every modification might not always be captured by an LLM, even when it is used to create commit messages. A rectifier, which refines and contextualizes commit messages on a per-file basis, is introduced to address this. The rectifier creates a more precise and detailed message for every file in a commit by examining the modifications made, such as added or removed lines.

For instance, for a particular Python file, an initial commit message such as "Fixed bugs" might be changed to "Corrected index error in function X," while another file inside the same commit might get a statement such as "Refined UI rendering in component Y." This method's primary innovation is its capacity to improve commit message precision, which will increase the clarity of subsequent analysis and strengthen the dependability of bug-fix tracking.







| **FILE NAME** | **ORIGINAL COMMIT MESSAGE** | **RECTIFIED MESSAGE** |
| --- | --- | --- |
| login.py | Fixed bug in login and updated readme for docs | Fixed bug in login functionality |
| utils.py | Fixed bug in login and updated readme for docs | Minor changes |
| README.MD | Fixed bug in login and updated readme for docs | Updated documentation. |