Step-by-step instructions on the Github Actions as the semi-automated Changelog solution

Of course! Here's the complete procedure, including step-by-step terminal commands, to execute the entire process of automating the Changelog update using the GitHub Actions script, the Python script, and the Markdown template.

## Step 1: Set Up Your Repository

1. Create a new directory for your project or navigate to your existing project directory in the terminal/colab/text editor.

2. Initialize a Git repository:

 ```bash

git init

```



3. Create relevant files in your repository:

1. GitHub Actions Script (`changelog.yml`):

Create a `.github/workflows` directory at the root of your repository (if it doesn't exist) and place the `changelog.yml` file inside it. The path should be `.github/workflows/changelog.yml`.

2. Python Script (`changelog\_script.py`):

Place the `changelog\_script.py` file in the root directory of your repository or in a directory of your choice. Just make sure you specify the correct path when you reference it in your GitHub Actions script.

3. Markdown Template (`changelog\_template.md`):

You mentioned a Markdown template earlier. You can create a file named `changelog\_template.md` and place it in the root directory of your repository or in a designated folder for documentation. This file contains the structure and placeholders for your changelog entries.

4. CHANGELOG.md:

Create a file named `CHANGELOG.md` in the root directory of your repository. This is where the changelog entries will be updated.

Here's how your directory structure could look:

```

your-repo/

|-- .github/

| `-- workflows/

| `-- changelog.yml

|-- changelog\_script.py

|-- changelog\_template.md

|-- CHANGELOG.md

```



Make sure to replace placeholders like `your-repo`, `your-organization`, and adjust paths as necessary based on your repository's structure.

With these files placed appropriately, the GitHub Actions workflow will be triggered when a pull request is closed and merged. The Python script will generate changelog entries based on the pull request information and the manual input you've incorporated. The generated entries will be added to the `CHANGELOG.md` file.

## Step 2: Fill in Changelog Script

Create a `.github/workflows/changelog.yml` file in your repository with the GitHub Actions script as follows:

name: Update Changelog

on:

pull\_request:

types:

- open

jobs:

update-changelog:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v2

- name: Set up Python

uses: actions/setup-python@v2

with:

python-version: 3.x

- name: Install dependencies

run: |

python -m pip install --upgrade pip

pip install requests

- name: Run Changelog Script

run: |

python path/to/changelog\_script.py ${{ github.event.pull\_request.number }}



## Step 3: Fill in Python Script

1. Open the `changelog\_script.py` file and populate it as follows:

import sys

import requests

GITHUB\_TOKEN = '<TOKEN>'

ORG\_NAME = 'your-organization'

REPO\_NAME = 'your-repo'

def get\_pull\_request(pull\_request\_number):

url = f'https://api.github.com/repos/{ORG\_NAME}/{REPO\_NAME}/pulls/{pull\_request\_number}'

headers = {'Authorization': f'token {GITHUB\_TOKEN}'}

response = requests.get(url, headers=headers)

return response.json()

def manual\_input(prompt):

user\_input = input(prompt)

return user\_input.strip()

def generate\_changelog\_entry(pull\_request):

changelog\_entry = ""

if 'Added' in pull\_request['labels']:

changelog\_entry += f"### Added\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}.\n"

# Add logic to process pull request body, if needed.

if 'Changed' in pull\_request['labels']:

changelog\_entry += f"### Changed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}.\n"

# Add logic to process pull request body, if needed.

if 'Fixed' in pull\_request['labels']:

changelog\_entry += f"### Fixed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}.\n"

# Add logic to process pull request body, if needed.

if 'Removed' in pull\_request['labels']:

changelog\_entry += f"### Removed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}.\n"

# Add logic to process pull request body, if needed.

if 'Documentation' in pull\_request['labels']:

user\_input = manual\_input("Enter documentation details: ")

changelog\_entry += f"### Documentation\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"

if 'Miscellaneous' in pull\_request['labels']:

user\_input = manual\_input("Enter miscellaneous details: ")

changelog\_entry += f"### Miscellaneous\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"

return changelog\_entry

def update\_changelog\_file(changelog\_entry):

# Implement updating the CHANGELOG.md file

# Open the file, append the changelog\_entry, and save the changes

with open('CHANGELOG.md', 'a') as changelog\_file:

changelog\_file.write(changelog\_entry)

def main(pull\_request\_number):

pull\_request = get\_pull\_request(pull\_request\_number)

changelog\_entry = generate\_changelog\_entry(pull\_request)

update\_changelog\_file(changelog\_entry)

if \_\_name\_\_ == '\_\_main\_\_':

if len(sys.argv) != 2:

print("Usage: python changelog\_script.py <pull\_request\_number>")

sys.exit(1)

pull\_request\_number = sys.argv[1]

main(pull\_request\_number)



2. Modify the Python script by replacing placeholders like `<TOKEN>`, `<ORG\_NAME>`, `<REPO\_NAME>`, and adjusting the paths as needed.

## Step 4: Fill in Markdown Template

1. Open your favorite text editor and create a Markdown template using the following structure:

# Changelog

## [Release Date] - Sprint X

### Added

- [PR Number] - Brief description of the feature/changes. [@Contributor]

- [PR Number] - Another addition or enhancement. [@Contributor]

### Changed

- [PR Number] - Description of changes or improvements. [@Contributor]

### Fixed

- [PR Number] - Details about the bug fix. [@Contributor]

### Removed

- [PR Number] - Explanation of removed features/components. [@Contributor]

### Documentation

- [PR Number] - Updates to documentation or inline comments. [@Contributor]

### Miscellaneous

- [PR Number] - Other minor updates or enhancements. [@Contributor]

## [Release Date] - Sprint X-1

... Repeat the structure for previous sprints ...



2. Save the template with a meaningful name like `changelog\_template.md`.

## Step 5: Terminal Commands

Here are the step-by-step terminal commands to execute the entire procedure:

1. Replace placeholders and modify the Python script, GitHub Actions script, and Markdown template according to your project's needs.

2. Run the following commands in your terminal:

```bash

# Add and commit the files

git add .

git commit -m "Initial commit"

# Replace placeholders in the terminal commands below

GITHUB\_TOKEN=your-github-token

ORG\_NAME=your-organization

REPO\_NAME=your-repo

# Create a remote repository on GitHub

curl -u $GITHUB\_TOKEN:x-oauth-basic -X POST https://api.github.com/orgs/$ORG\_NAME/repos -d '{"name":"'$REPO\_NAME'"}'

# Set the remote origin and push to GitHub

git remote add origin https://github.com/$ORG\_NAME/$REPO\_NAME.git

git branch -M main

git push -u origin main

# Manually trigger the GitHub Action to update the changelog

# If using the schedule, this step can be skipped

curl -X POST -H "Authorization: token $GITHUB\_TOKEN" \

-H "Accept: application/vnd.github.v3+json" \

https://api.github.com/repos/$ORG\_NAME/$REPO\_NAME/actions/workflows/changelog.yml/dispatches \

-d '{"ref":"main"}'

```



## Step 6: Automate the Process

If you're using the schedule in the GitHub Actions script, the Changelog update will be automatically triggered according to the schedule. Otherwise, you can manually trigger the GitHub Action as demonstrated in the last terminal command.

With these steps, you'll have set up a process to automate the Changelog update using GitHub Actions, Python scripting, and a Markdown template. Remember to replace placeholders with your actual information and adjust paths accordingly.

name: Update Changelog  
  
on:  
 pull\_request:  
 types:  
 - open  
  
jobs:  
 update-changelog:  
 runs-on: ubuntu-latest  
  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v2  
  
 - name: Set up Python  
 uses: actions/setup-python@v2  
 with:  
 python-version: 3.x  
  
 - name: Install dependencies  
 run: |  
 python -m pip install --upgrade pip  
 pip install requests  
  
 - name: Run Changelog Script  
 run: |  
 python path/to/changelog\_script.py ${{ github.event.pull\_request.number }}

name: Update Changelog  
  
on:  
 pull\_request:  
 types:  
 - open  
  
jobs:  
 update-changelog:  
 runs-on: ubuntu-latest  
  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v2  
  
 - name: Set up Python  
 uses: actions/setup-python@v2  
 with:  
 python-version: 3.x  
  
 - name: Install dependencies  
 run: |  
 python -m pip install --upgrade pip  
 pip install requests  
  
 - name: Run Changelog Script  
 run: |  
 python path/to/changelog\_script.py ${{ github.event.pull\_request.number }}

import sys  
import requests  
  
GITHUB\_TOKEN = '<TOKEN>'  
ORG\_NAME = 'your-organization'  
REPO\_NAME = 'your-repo'  
  
def get\_pull\_request(pull\_request\_number):  
 url = f'https://api.github.com/repos/{ORG\_NAME}/{REPO\_NAME}/pulls/{pull\_request\_number}'  
 headers = {'Authorization': f'token {GITHUB\_TOKEN}'}  
 response = requests.get(url, headers=headers)  
 return response.json()  
  
def manual\_input(prompt):  
 user\_input = input(prompt)  
 return user\_input.strip()  
  
def generate\_changelog\_entry(pull\_request):  
 changelog\_entry = ""  
  
 if 'Added' in pull\_request['labels']:  
 changelog\_entry += f"### Added\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Changed' in pull\_request['labels']:  
 changelog\_entry += f"### Changed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Fixed' in pull\_request['labels']:  
 changelog\_entry += f"### Fixed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Removed' in pull\_request['labels']:  
 changelog\_entry += f"### Removed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Documentation' in pull\_request['labels']:  
 user\_input = manual\_input("Enter documentation details: ")  
 changelog\_entry += f"### Documentation\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 if 'Miscellaneous' in pull\_request['labels']:  
 user\_input = manual\_input("Enter miscellaneous details: ")  
 changelog\_entry += f"### Miscellaneous\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 return changelog\_entry  
  
def update\_changelog\_file(changelog\_entry):  
 # Implement updating the CHANGELOG.md file  
 # Open the file, append the changelog\_entry, and save the changes  
 with open('CHANGELOG.md', 'a') as changelog\_file:  
 changelog\_file.write(changelog\_entry)  
  
def main(pull\_request\_number):  
 pull\_request = get\_pull\_request(pull\_request\_number)  
 changelog\_entry = generate\_changelog\_entry(pull\_request)  
 update\_changelog\_file(changelog\_entry)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 if len(sys.argv) != 2:  
 print("Usage: python changelog\_script.py <pull\_request\_number>")  
 sys.exit(1)  
  
 pull\_request\_number = sys.argv[1]  
 main(pull\_request\_number)

import sys  
import requests  
  
GITHUB\_TOKEN = '<TOKEN>'  
ORG\_NAME = 'your-organization'  
REPO\_NAME = 'your-repo'  
  
def get\_pull\_request(pull\_request\_number):  
 url = f'https://api.github.com/repos/{ORG\_NAME}/{REPO\_NAME}/pulls/{pull\_request\_number}'  
 headers = {'Authorization': f'token {GITHUB\_TOKEN}'}  
 response = requests.get(url, headers=headers)  
 return response.json()  
  
def manual\_input(prompt):  
 user\_input = input(prompt)  
 return user\_input.strip()  
  
def generate\_changelog\_entry(pull\_request):  
 changelog\_entry = ""  
  
 if 'Added' in pull\_request['labels']:  
 changelog\_entry += f"### Added\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Changed' in pull\_request['labels']:  
 changelog\_entry += f"### Changed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Fixed' in pull\_request['labels']:  
 changelog\_entry += f"### Fixed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Removed' in pull\_request['labels']:  
 changelog\_entry += f"### Removed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Documentation' in pull\_request['labels']:  
 user\_input = manual\_input("Enter documentation details: ")  
 changelog\_entry += f"### Documentation\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 if 'Miscellaneous' in pull\_request['labels']:  
 user\_input = manual\_input("Enter miscellaneous details: ")  
 changelog\_entry += f"### Miscellaneous\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 return changelog\_entry  
  
def update\_changelog\_file(changelog\_entry):  
 # Implement updating the CHANGELOG.md file  
 # Open the file, append the changelog\_entry, and save the changes  
 with open('CHANGELOG.md', 'a') as changelog\_file:  
 changelog\_file.write(changelog\_entry)  
  
def main(pull\_request\_number):  
 pull\_request = get\_pull\_request(pull\_request\_number)  
 changelog\_entry = generate\_changelog\_entry(pull\_request)  
 update\_changelog\_file(changelog\_entry)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 if len(sys.argv) != 2:  
 print("Usage: python changelog\_script.py <pull\_request\_number>")  
 sys.exit(1)  
  
 pull\_request\_number = sys.argv[1]  
 main(pull\_request\_number)

# Changelog  
  
## [Release Date] - Sprint X  
  
### Added  
- [PR Number] - Brief description of the feature/changes. [@Contributor]  
- [PR Number] - Another addition or enhancement. [@Contributor]  
  
### Changed  
- [PR Number] - Description of changes or improvements. [@Contributor]  
  
### Fixed  
- [PR Number] - Details about the bug fix. [@Contributor]  
  
### Removed  
- [PR Number] - Explanation of removed features/components. [@Contributor]  
  
### Documentation  
- [PR Number] - Updates to documentation or inline comments. [@Contributor]  
  
### Miscellaneous  
- [PR Number] - Other minor updates or enhancements. [@Contributor]  
  
## [Release Date] - Sprint X-1

# Changelog  
  
## [Release Date] - Sprint X  
  
### Added  
- [PR Number] - Brief description of the feature/changes. [@Contributor]  
- [PR Number] - Another addition or enhancement. [@Contributor]  
  
### Changed  
- [PR Number] - Description of changes or improvements. [@Contributor]  
  
### Fixed  
- [PR Number] - Details about the bug fix. [@Contributor]  
  
### Removed  
- [PR Number] - Explanation of removed features/components. [@Contributor]  
  
### Documentation  
- [PR Number] - Updates to documentation or inline comments. [@Contributor]  
  
### Miscellaneous  
- [PR Number] - Other minor updates or enhancements. [@Contributor]  
  
## [Release Date] - Sprint X-1

name: Update Changelog  
  
on:  
 pull\_request:  
 types:  
 - open  
  
jobs:  
 update-changelog:  
 runs-on: ubuntu-latest  
  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v2  
  
 - name: Set up Python  
 uses: actions/setup-python@v2  
 with:  
 python-version: 3.x  
  
 - name: Install dependencies  
 run: |  
 python -m pip install --upgrade pip  
 pip install requests  
  
 - name: Run Changelog Script  
 run: |  
 python path/to/changelog\_script.py ${{ github.event.pull\_request.number }}

import sys  
import requests  
  
GITHUB\_TOKEN = '<TOKEN>'  
ORG\_NAME = 'your-organization'  
REPO\_NAME = 'your-repo'  
  
def get\_pull\_request(pull\_request\_number):  
 url = f'https://api.github.com/repos/{ORG\_NAME}/{REPO\_NAME}/pulls/{pull\_request\_number}'  
 headers = {'Authorization': f'token {GITHUB\_TOKEN}'}  
 response = requests.get(url, headers=headers)  
 return response.json()  
  
def manual\_input(prompt):  
 user\_input = input(prompt)  
 return user\_input.strip()  
  
def generate\_changelog\_entry(pull\_request):  
 changelog\_entry = ""  
  
 if 'Added' in pull\_request['labels']:  
 changelog\_entry += f"### Added\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Changed' in pull\_request['labels']:  
 changelog\_entry += f"### Changed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Fixed' in pull\_request['labels']:  
 changelog\_entry += f"### Fixed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Removed' in pull\_request['labels']:  
 changelog\_entry += f"### Removed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Documentation' in pull\_request['labels']:  
 user\_input = manual\_input("Enter documentation details: ")  
 changelog\_entry += f"### Documentation\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 if 'Miscellaneous' in pull\_request['labels']:  
 user\_input = manual\_input("Enter miscellaneous details: ")  
 changelog\_entry += f"### Miscellaneous\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 return changelog\_entry  
  
def update\_changelog\_file(changelog\_entry):  
 # Implement updating the CHANGELOG.md file  
 # Open the file, append the changelog\_entry, and save the changes  
 with open('CHANGELOG.md', 'a') as changelog\_file:  
 changelog\_file.write(changelog\_entry)  
  
def main(pull\_request\_number):  
 pull\_request = get\_pull\_request(pull\_request\_number)  
 changelog\_entry = generate\_changelog\_entry(pull\_request)  
 update\_changelog\_file(changelog\_entry)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 if len(sys.argv) != 2:  
 print("Usage: python changelog\_script.py <pull\_request\_number>")  
 sys.exit(1)  
  
 pull\_request\_number = sys.argv[1]  
 main(pull\_request\_number)

# Changelog  
  
## [Release Date] - Sprint X  
  
### Added  
- [PR Number] - Brief description of the feature/changes. [@Contributor]  
- [PR Number] - Another addition or enhancement. [@Contributor]  
  
### Changed  
- [PR Number] - Description of changes or improvements. [@Contributor]  
  
### Fixed  
- [PR Number] - Details about the bug fix. [@Contributor]  
  
### Removed  
- [PR Number] - Explanation of removed features/components. [@Contributor]  
  
### Documentation  
- [PR Number] - Updates to documentation or inline comments. [@Contributor]  
  
### Miscellaneous  
- [PR Number] - Other minor updates or enhancements. [@Contributor]  
  
## [Release Date] - Sprint X-1

name: Update Changelog  
  
on:  
 pull\_request:  
 types:  
 - open  
  
jobs:  
 update-changelog:  
 runs-on: ubuntu-latest  
  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v2  
  
 - name: Set up Python  
 uses: actions/setup-python@v2  
 with:  
 python-version: 3.x  
  
 - name: Install dependencies  
 run: |  
 python -m pip install --upgrade pip  
 pip install requests  
  
 - name: Run Changelog Script  
 run: |  
 python path/to/changelog\_script.py ${{ github.event.pull\_request.number }}

import sys  
import requests  
  
GITHUB\_TOKEN = '<TOKEN>'  
ORG\_NAME = 'your-organization'  
REPO\_NAME = 'your-repo'  
  
def get\_pull\_request(pull\_request\_number):  
 url = f'https://api.github.com/repos/{ORG\_NAME}/{REPO\_NAME}/pulls/{pull\_request\_number}'  
 headers = {'Authorization': f'token {GITHUB\_TOKEN}'}  
 response = requests.get(url, headers=headers)  
 return response.json()  
  
def manual\_input(prompt):  
 user\_input = input(prompt)  
 return user\_input.strip()  
  
def generate\_changelog\_entry(pull\_request):  
 changelog\_entry = ""  
  
 if 'Added' in pull\_request['labels']:  
 changelog\_entry += f"### Added\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Changed' in pull\_request['labels']:  
 changelog\_entry += f"### Changed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Fixed' in pull\_request['labels']:  
 changelog\_entry += f"### Fixed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Removed' in pull\_request['labels']:  
 changelog\_entry += f"### Removed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Documentation' in pull\_request['labels']:  
 user\_input = manual\_input("Enter documentation details: ")  
 changelog\_entry += f"### Documentation\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 if 'Miscellaneous' in pull\_request['labels']:  
 user\_input = manual\_input("Enter miscellaneous details: ")  
 changelog\_entry += f"### Miscellaneous\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 return changelog\_entry  
  
def update\_changelog\_file(changelog\_entry):  
 # Implement updating the CHANGELOG.md file  
 # Open the file, append the changelog\_entry, and save the changes  
 with open('CHANGELOG.md', 'a') as changelog\_file:  
 changelog\_file.write(changelog\_entry)  
  
def main(pull\_request\_number):  
 pull\_request = get\_pull\_request(pull\_request\_number)  
 changelog\_entry = generate\_changelog\_entry(pull\_request)  
 update\_changelog\_file(changelog\_entry)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 if len(sys.argv) != 2:  
 print("Usage: python changelog\_script.py <pull\_request\_number>")  
 sys.exit(1)  
  
 pull\_request\_number = sys.argv[1]  
 main(pull\_request\_number)

import sys  
import requests  
  
GITHUB\_TOKEN = '<TOKEN>'  
ORG\_NAME = 'your-organization'  
REPO\_NAME = 'your-repo'  
  
def get\_pull\_request(pull\_request\_number):  
 url = f'https://api.github.com/repos/{ORG\_NAME}/{REPO\_NAME}/pulls/{pull\_request\_number}'  
 headers = {'Authorization': f'token {GITHUB\_TOKEN}'}  
 response = requests.get(url, headers=headers)  
 return response.json()  
  
def manual\_input(prompt):  
 user\_input = input(prompt)  
 return user\_input.strip()  
  
def generate\_changelog\_entry(pull\_request):  
 changelog\_entry = ""  
  
 if 'Added' in pull\_request['labels']:  
 changelog\_entry += f"### Added\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Changed' in pull\_request['labels']:  
 changelog\_entry += f"### Changed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Fixed' in pull\_request['labels']:  
 changelog\_entry += f"### Fixed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Removed' in pull\_request['labels']:  
 changelog\_entry += f"### Removed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Documentation' in pull\_request['labels']:  
 user\_input = manual\_input("Enter documentation details: ")  
 changelog\_entry += f"### Documentation\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 if 'Miscellaneous' in pull\_request['labels']:  
 user\_input = manual\_input("Enter miscellaneous details: ")  
 changelog\_entry += f"### Miscellaneous\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 return changelog\_entry  
  
def update\_changelog\_file(changelog\_entry):  
 # Implement updating the CHANGELOG.md file  
 # Open the file, append the changelog\_entry, and save the changes  
 with open('CHANGELOG.md', 'a') as changelog\_file:  
 changelog\_file.write(changelog\_entry)  
  
def main(pull\_request\_number):  
 pull\_request = get\_pull\_request(pull\_request\_number)  
 changelog\_entry = generate\_changelog\_entry(pull\_request)  
 update\_changelog\_file(changelog\_entry)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 if len(sys.argv) != 2:  
 print("Usage: python changelog\_script.py <pull\_request\_number>")  
 sys.exit(1)  
  
 pull\_request\_number = sys.argv[1]  
 main(pull\_request\_number)

import sys  
import requests  
  
GITHUB\_TOKEN = '<TOKEN>'  
ORG\_NAME = 'your-organization'  
REPO\_NAME = 'your-repo'  
  
def get\_pull\_request(pull\_request\_number):  
 url = f'https://api.github.com/repos/{ORG\_NAME}/{REPO\_NAME}/pulls/{pull\_request\_number}'  
 headers = {'Authorization': f'token {GITHUB\_TOKEN}'}  
 response = requests.get(url, headers=headers)  
 return response.json()  
  
def manual\_input(prompt):  
 user\_input = input(prompt)  
 return user\_input.strip()  
  
def generate\_changelog\_entry(pull\_request):  
 changelog\_entry = ""  
  
 if 'Added' in pull\_request['labels']:  
 changelog\_entry += f"### Added\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Changed' in pull\_request['labels']:  
 changelog\_entry += f"### Changed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Fixed' in pull\_request['labels']:  
 changelog\_entry += f"### Fixed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Removed' in pull\_request['labels']:  
 changelog\_entry += f"### Removed\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n"  
 # Add logic to process pull request body, if needed.  
  
 if 'Documentation' in pull\_request['labels']:  
 user\_input = manual\_input("Enter documentation details: ")  
 changelog\_entry += f"### Documentation\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 if 'Miscellaneous' in pull\_request['labels']:  
 user\_input = manual\_input("Enter miscellaneous details: ")  
 changelog\_entry += f"### Miscellaneous\n- {pull\_request['title']} (#{pull\_request['number']}) - {pull\_request['user']['login']}\n Details: {user\_input}\n"  
  
 return changelog\_entry  
  
def update\_changelog\_file(changelog\_entry):  
 # Implement updating the CHANGELOG.md file  
 # Open the file, append the changelog\_entry, and save the changes  
 with open('CHANGELOG.md', 'a') as changelog\_file:  
 changelog\_file.write(changelog\_entry)  
  
def main(pull\_request\_number):  
 pull\_request = get\_pull\_request(pull\_request\_number)  
 changelog\_entry = generate\_changelog\_entry(pull\_request)  
 update\_changelog\_file(changelog\_entry)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 if len(sys.argv) != 2:  
 print("Usage: python changelog\_script.py <pull\_request\_number>")  
 sys.exit(1)  
  
 pull\_request\_number = sys.argv[1]  
 main(pull\_request\_number)

# Changelog  
  
## [Release Date] - Sprint X  
  
### Added  
- [PR Number] - Brief description of the feature/changes. [@Contributor]  
- [PR Number] - Another addition or enhancement. [@Contributor]  
  
### Changed  
- [PR Number] - Description of changes or improvements. [@Contributor]  
  
### Fixed  
- [PR Number] - Details about the bug fix. [@Contributor]  
  
### Removed  
- [PR Number] - Explanation of removed features/components. [@Contributor]  
  
### Documentation  
- [PR Number] - Updates to documentation or inline comments. [@Contributor]  
  
### Miscellaneous  
- [PR Number] - Other minor updates or enhancements. [@Contributor]  
  
## [Release Date] - Sprint X-1

name: Update Changelog  
  
on:  
 pull\_request:  
 types:  
 - open  
  
jobs:  
 update-changelog:  
 runs-on: ubuntu-latest  
  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v2  
  
 - name: Set up Python  
 uses: actions/setup-python@v2  
 with:  
 python-version: 3.x  
  
 - name: Install dependencies  
 run: |  
 python -m pip install --upgrade pip  
 pip install requests  
  
 - name: Run Changelog Script  
 run: |  
 python path/to/changelog\_script.py ${{ github.event.pull\_request.number }}