```
import os
import subprocess

import requests
from dotenv import load_dotenv

load_dotenv

# Constants
GITHUB_USERNAME = os.getenv("GITHUB_USERNAME")
REPO_NAME = os.getenv("GITHUB_REPO_NAME")
GITHUB_TOKEN = os.getenv("GITHUB_TOKEN")
ISSUES_URL = f"https://api.github.com/repos/{GITHUB_USERNAME}/{REPO_NAME}/issues"

# Headers for authentication
headers = {
    "Authorization": f"token {GITHUB_TOKEN}",
    "Accept": "application/vnd.github.v3+json",
}


def run_pytest():
    result = subprocess.run(
        ["pytest"], capture_output=True, text=True
    )
    return result.stdout + result.stderr
```

```
def parse_pytest_output(output):
    errors = []
    current_error = None

    for line in output.split("\n"):
        if line.startswith("_________________________"):
            if current_error:
                errors.append(current_error)
            current_error = {"title": "", "body": ""}
        elif current_error is not None:
            if not current_error["title"]:
                current_error["title"] = line.strip()
            current_error["body"] += line + "\n"

    if current_error:
        errors.append(current_error)
    return errors


def create_github_issue(title, body):
    issue = {"title": title, "body": body}
    response = requests.post(ISSUES_URL, headers=headers, json=issue)
    return response.json()
```

```
def main():

    pytest_output = run_pytest()

    errors = parse_pytest_output(pytest_output)


    for error in errors:

        issue_response = create_github_issue(

            error["title"], error["body"]

        )

        print(f"Issue created: {issue_response.get('html_url')}")



if __name__ == "__main__":

    main()
```