Create a GitHub Actions CI workflow with one job (cross_build), matrix for ubuntu/windows/macos, triggers on push/PR to branches containing "develop" or "master", and steps: checkout, install CMake, run build script from stage 3, and deploy.

~~~
```name=.github/workflows/ci.yml
name: CI

on:
  push:
    branches:
      - '*develop*'
      - '*master*'
  pull_request:
    branches:
      - '*develop*'
      - '*master*'

jobs:
  cross_build:
    name: cross_build - ${{ matrix.os }}
    runs-on: ${{ matrix.os }}
    strategy:
      matrix:
        os: [ubuntu-latest, windows-latest, macos-latest]

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Setup CMake
        uses: lukka/get-cmake@v3
        with:
          cmake-version: '3.24.3'
~~~

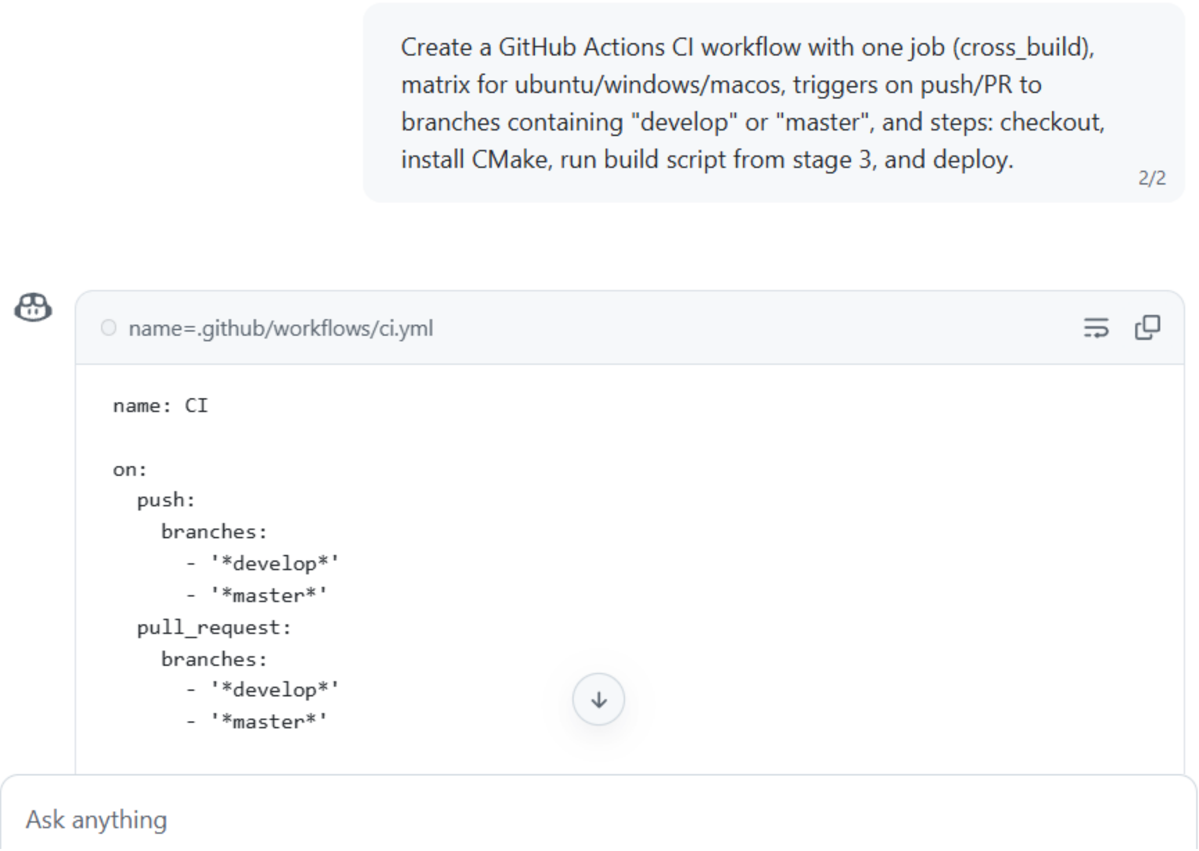
```
- name: Build & Test (Linux/macOS)
  if: matrix.os != 'windows-latest'
  run: |
    set -e
    mkdir -p build
    cd build
    cmake -DCMAKE_BUILD_TYPE=Release ..
    cmake --build . --config Release --parallel
    ctest --output-on-failure -C Release
  shell: bash

- name: Build & Test (Windows)
  if: matrix.os == 'windows-latest'
  run: |
    REM Use the project's Windows build script if present, otherwise run equivalent commands
    if exist "%GITHUB_WORKSPACE%\build_and_test.bat" (
      build_and_test.bat Release
    ) else (
      mkdir build
      cd build
      cmake -DCMAKE_BUILD_TYPE=Release ..
      cmake --build . --config Release -- /m
      ctest --output-on-failure -C Release
    )
  shell: cmd

- name: Upload build artifacts (deploy placeholder)
  uses: actions/upload-artifact@v4
  with:
    name: build-artifacts-${{ matrix.os }}
    path: |
      build/bin
      build\bin
```

`` ``` ``

I created a single-job GitHub Actions workflow named CI with a cross_build job that runs on a matrix of ubuntu-latest, windows-latest, and macos-latest. It checks out the code, installs CMake, runs platform-appropriate build-and-test steps (on Windows it prefers your build_and_test.bat if present; on Linux/macOS it runs equivalent CMake + CTest commands), and then uploads the build/bin output as a deploy placeholder artifact. The workflow triggers on push and pull_request events for branches whose names contain "develop" or "master". If you want a real deployment step (e.g., to a package registry, GitHub Releases, or a cloud provider) tell me which target and I’ll replace the artifact upload with the appropriate deploy action.



Create a GitHub Actions CI workflow with one job (cross_build), matrix for ubuntu/windows/macos, triggers on push/PR to branches containing "develop" or "master", and steps: checkout, install CMake, run build script from stage 3, and deploy.

name=.github/workflows/ci.yml

```yaml
name: CI

on:
  push:
    branches:
      - '*develop*'
      - '*master*'
  pull_request:
    branches:
      - '*develop*'
      - '*master*'
```

```
            cmake -DCMAKE_BUILD_TYPE=Release ..
            cmake --build . --config Release -- /m
            ctest --output-on-failure -C Release
          )
        shell: cmd

      - name: Upload build artifacts (deploy placeholder)
        uses: actions/upload-artifact@v4
        with:
          name: build-artifacts-${{ matrix.os }}
          path: |
            build/bin
            build\bin
```

I created a single-job GitHub Actions workflow named CI with a cross_build job that runs on a matrix of ubuntu-latest, windows-latest, and macos-latest. It checks out the code, installs CMake, runs platform-appropriate build-and-test steps (on Windows it prefers your build_and_test.bat if present; on Linux/macOS it runs equivalent CMake + CTest commands), and then uploads the build/bin output as a deploy placeholder artifact. The workflow triggers on push and pull_request events for branches whose names contain "develop" or "master". If you want a real deployment step (e.g., to a package registry, GitHub Releases, or a cloud provider) tell me which target and I'll replace the artifact upload with the appropriate deploy action.