CASE STUDY - GIT WORKFLOW

Problem Statement:
You work as a Devops Architect in Zendriix Softwares. The company has been struggling to manage their product releases. The releases should happen on 25th of every month. Suggest a
Git Workflow Architecture for this requirement.
Simulate this workflow, by creating a pseudo code files and branches, and upload the same to your GitHub Account.
As a part of solution, share the link to your GitHub repository.

Solution:

Git Workflow Architecture

1. **Main Branches:**
   - `master`: Contains production-ready code.
   - `develop`: Used for ongoing development and integration of new features.
2. **Feature Branches:**
   - Created from `develop` for specific features. Example: `feature/feature1`.
3. **Release Branches:**
   - Created from `develop` to stabilize the code for an upcoming release. Example: `release/2025-01-25`.
4. **Hotfix Branches:**
   - Created from `master` to address critical issues in production. Example: `hotfix/urgent-fix`.

Workflow Steps

1. **Create a Feature Branch:**
   - Branch off from `develop`: git checkout -b feature1 develop
2. **Work on the Feature Branch:**

   Add code and commit changes:

- echo "Code for feature1" > feature1.txt
- git add feature1.txt
- git commit -m "Add feature1.txt for feature1"

Merge Feature Branch into Develop:

- Once the feature is complete, merge it back into `develop`:

  git checkout develop

  git merge feature1

  git push origin develop

Create a Release Branch:

- Typically created around the 20th of each month to prepare for the release on the 25th:

  git checkout -b release/2025-01-25 develop

Prepare and Finalize the Release:

- Make any final changes and commit:

  echo "Finalized code for release 2025-01-25" > release-notes.txt

  git add release-notes.txt

  git commit -m "Prepare release 2025-01-25"

Merge Release Branch into Master and Develop:

- Merge the release branch into `master` and tag the release:

  git checkout master

  git merge release/2025-01-25

  git tag -a v2025-01-25 -m "Release 2025-01-25"

  git push origin master

  git push origin –tags

  Merge the release branch back into `develop` to keep it updated:

  git checkout develop

  git merge release/2025-01-25

  git push origin develop

Handle Urgent Hotfixes:

- Create a hotfix branch from `master`:

  git checkout master

  git checkout -b hotfix/urgent-fix

  Add the urgent fix, commit, and merge it back into `master` and `develop`:

  echo "Urgent fix for production" > urgent.txt

git add urgent.txt

git commit -m "Add urgent.txt for hotfix"

git checkout master

git merge hotfix/urgent-fix

git push origin master

git checkout develop

git merge hotfix/urgent-fix

git push origin develop

pseudo code file to simulate the Gitflow workflow architecture for managing product releases at Zendriix Softwares.

Step-by-Step Instructions:

1. **Initialize the Repository and Create Branches:**
2. **Develop a Feature and Merge It:**
3. **Prepare for Release:**
4. **Handle an Urgent Hotfix:**

Pseudo Code for Gitflow Workflow

Let's put this into a `README.md` file as pseudo code:

Link : https://github.com/poulomi-design999/Devops/blob/main/README.md