

# 11 · Git Repository & Branch Management

## 11.1 Repository Setup

The project code is hosted on GitHub. To get started, clone the repository using:

```
git clone https://github.com/Guipini/movies-app.git
cd movies-app
npm install
```

## 11.2 Branch Strategy

Branch Type	Naming Convention	Purpose
main	main	Production-ready code
develop	develop	Integration branch for features
feature	feature/name	New features development
bugfix	bugfix/name	Bug fixes

## 11.3 Developer Workflow

### 1. Clone the repository (first time only)

```
git clone https://github.com/Guipini/movies-app.git
cd movies-app
```

### 2. Create a new feature branch

```
git checkout develop
git pull origin develop
git checkout -b feature/your-feature-name
```

### 3. Make changes and commit

```
git add .
git commit -m "Descriptive message about your changes"
```

### 4. Push your branch to GitHub

```
git push -u origin feature/your-feature-name
```

## 5. Create a Pull Request

- Go to the [repository on GitHub](#)
- Click "Compare & pull request"
- Set the base branch to `develop`
- Add a description of your changes
- Request review from team members

## 11.4 Code Review Process

- All code changes require at least one review
- Reviewers should check for:
  - Functionality (meets requirements)
  - Code quality and style
  - Test coverage
  - Documentation

## 11.5 Merging Guidelines

- Feature branches merge into `develop`
- `develop` merges into `main` for releases
- Use squash merging to keep history clean
- Delete branches after merging

## 11.6 Conflict Resolution

If you encounter merge conflicts:

```
git checkout develop
git pull origin develop
git checkout your-branch
git merge develop
# Resolve conflicts in your editor
git add .
git commit -m "Resolved merge conflicts"
git push origin your-branch
```