Continuous Integration

What is Continuous Integration?

- Continuous Integration (CI) is a practice where team members make changes on *development branches* and *frequently merge* them into the `main` branch.
- The `main` branch is *protected* - nobody should push to it directly.
- Each commit triggers the *CI platform* to build the project and run its tests in the cloud. If tests fail, the commit is labelled as *failing*.
- A development branch shouldn’t be merged if its latest commit is failing.

What's the point? *Greater Software Quality*

- If anyone in the team can freely push to the `main` branch, then there's nothing stopping them from pushing buggy code.
- A bug introduced in one commit may not be discovered until a later point in time when someone decides to run the tests.
- Because tests run after every commit, team members immediately know if they've introduced bugs.
- Also, because failing branches don't get merged, the team has greater assurance that the code in `main` is correct.

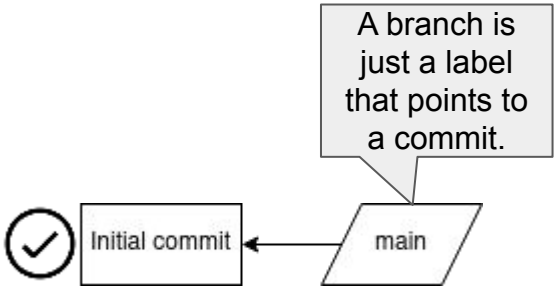
What's the point? *Smoother Collaboration*

- So far, you've most likely been working on a single branch - `main`.
- This means if multiple team members are working simultaneously, one person's changes could impact what somebody else is doing.
- By having separate development branches, it's as if you're the only person making changes to the project at any time.
- The point of CI is to use development branches for small changes and merge them into `main` frequently enough to avoid large *merge conflicts*.

Team member 1

```
git clone project
```

Team member 2



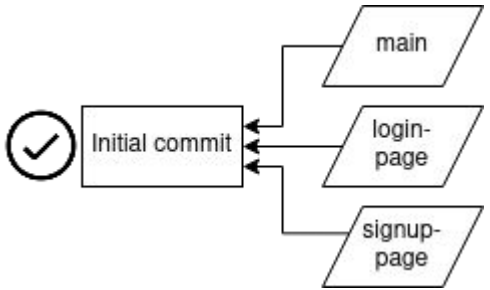

```
git clone project
```

Team member 1

```
git clone project
git branch login-page
git checkout login-page
git push -u origin login-page
```

Team member 2

```
git clone project
git branch signup-page
git checkout signup-page
git push -u origin signup-page
```



Team member 1

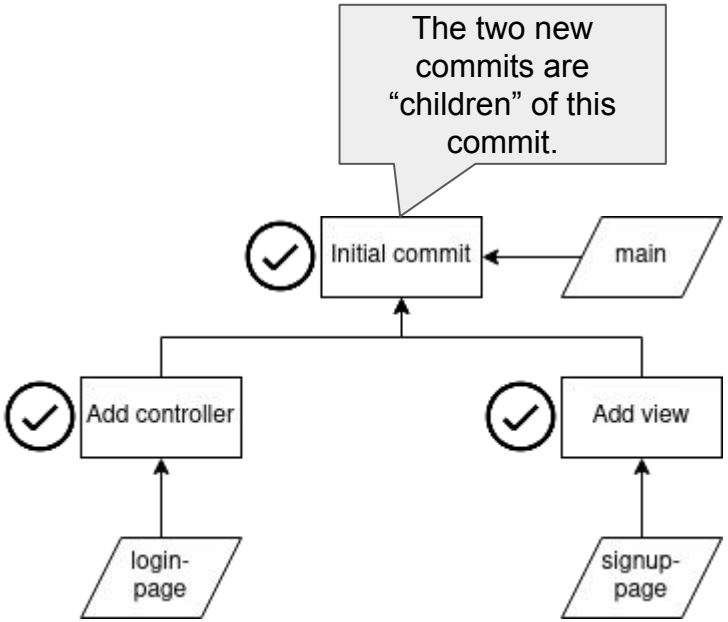
Can merge

```
git clone project
git branch login-page
git checkout login-page
git push -u origin login-page
git commit -m “Add controller”
git push
```

Team member 2

Can merge

```
git clone project
git branch signup-page
git checkout signup-page
git push -u origin signup-page
git commit -m “Add view”
git push
```



Team member 1

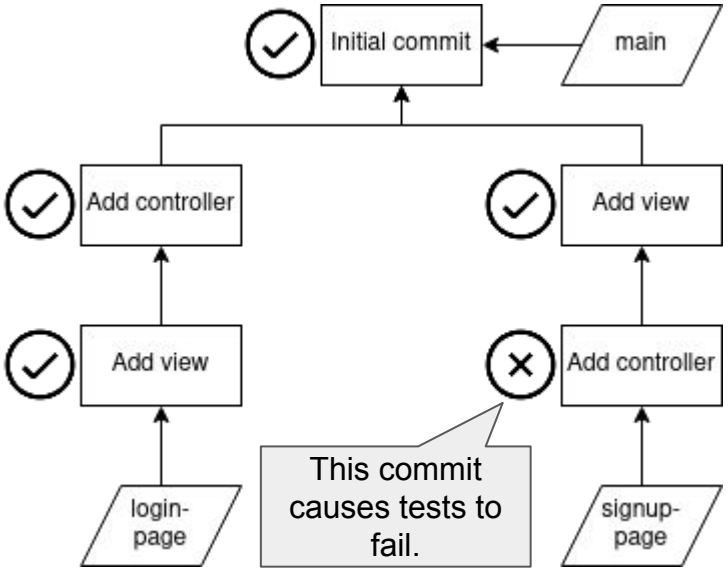
Can merge

```
git checkout login-page
git push -u origin login-page
git commit -m “Add controller”
git push
git commit -m “Add view”
git push
```

Team member 2

Cannot merge

```
git checkout signup-page
git push -u origin signup-page
git commit -m “Add view”
git push
git commit -m “Add controller”
git push
```



Team member 1

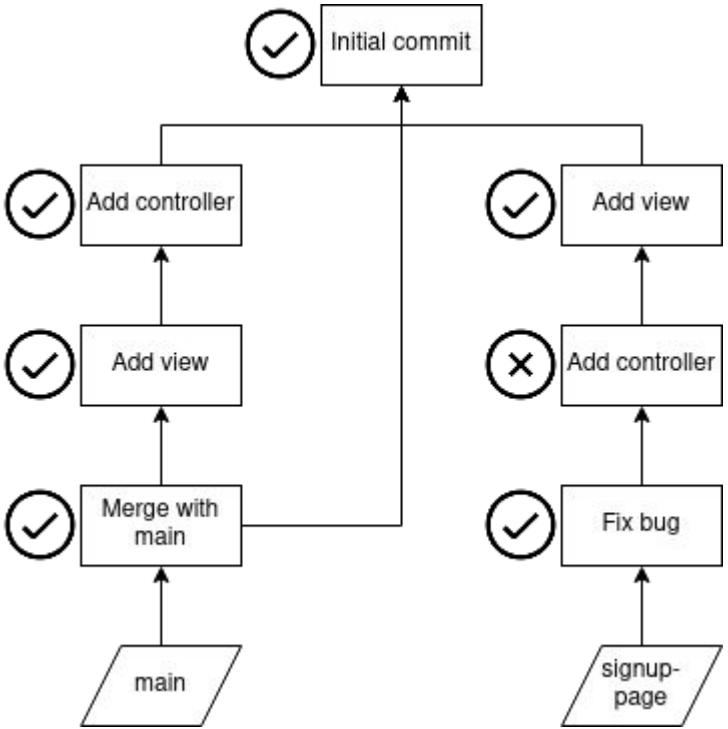
Merged

```
git checkout login-page
git push -u origin login-page
git commit -m “Add controller”
git push
git commit -m “Add view”
git push
```

Team member 2

Can merge

```
git commit -m “Add view”
git push
git commit -m “Add controller”
git push
git commit -m “Fix bug”
git push
```



Team member 1

Merged

```
git checkout login-page
git push -u origin login-page
git commit -m “Add controller”
git push
git commit -m “Add view”
git push
```

Team member 2

Merged

```
git commit -m “Add view”
git push
git commit -m “Add controller”
git push
git commit -m “Fix bug”
git push
```