GIT workflow

1. Introduction

This document has the purpose of instructing the team on how to take advantage of the version management system and use it to its full potential.

To avoid merge conflicts and delays during the development phase, we will use a feature-based approach.

When a person starts working on a feature (ex. a page), they will create a new branch. They will do as many commits and pushes as they see fit, but only on their feature branch. They will do pulls from the main/master branch as often as possible, at least every time they open Visual Studio Code to make sure that their work is compatible with the rest of the website. Only when their feature is finished, they will do a push request, and their code will be reviewed to ensure that there are no bugs present. After the code is reviewed by at least one other person, the feature branch will be merged with the main/master branch. Nobody will be able to directly push to the main branch.

1. Setting up the workspace

Open PowerShell (or Mac OS or Linux equivalent) in the NHL\_Stenden\_PHP\_Docker\_Env-0.4\app\public directory and type the following command:

git clone <SSH/HTTPS of the repository>

Mac OS users will use .gitignore to keep the automatically generated Mac OS files out of the repository.

1. Creating feature branches

Branches follow this naming convention:

feature-<name of page/script>

For instance: feature-checkout or feature-login or feature-about-us

Before creating a branch, make sure you are on the main branch:

git checkout main

Make sure the code is up to date:

git pull origin main

Creating a branch is a simple process, just type the command:

git branch <branch name>

After creating a branch you need to select it with this command:

git checkout <branch name>

Then, the branch must be created on GitHub as well

git push –set-upstream origin <branch name>

1. Routine

Before starting work always make sure you are on the desired branch by doing:

git branch

Also make sure branch is up to date with main by doing:

git pull origin main

If a file you are working on is modified in the main branch (for instance style.css, because everyone is working on the same file), you will have to do a merge. For the style.css file all you will need to do will be to move your code under the code from the main branch.

Start coding and do a commit to save your progress whenever you get a new element working by doing:

git add \* (or another desired selector)

git commit -m “<describe the modification>”

For instance: git commit “finished input form styling”

Before pushing the changes to GitHub make sure everything is still compatible with the main branch

git pull origin main

Then upload your changes to GitHub

git push

1. Pull request

Pull requests are done on github.com by selecting your feature branch and selecting the option “Open pull request”. You will need to add a description and a title to the pull request.

1. Merging the feature branches with the main branch

After the pull request has been submitted, at least one other person will review it and check for any bugs. If there are any bugs, they will have to get fixed before trying to do a pull request again. Otherwise, the feature branch will be merged with main.