Workshop 07: Git & Gitlab

Brainster Web Development Academy

Table of contents

Intro

For this week's workshop, you will need to work on one exercise. Using the scenario below you need to go step by step and implement the git commands from the previous lectures to solved the exercise.

Scenario: A company starts a new project, working by agile methodology, the first few weeks will be worked on MVP, after that we will constantly work on adding new functionalities and solving bugs.

01

- In this exercise, you going to work as a team. It is easiest to use the same teams as for a mentoring exercise, although you are free to mingle with each other. The team should have a minimum of 2 members. The tasks of the team are as follows:

    - You need to create a Gitlab repository for the project. One member of the group is creating it(not everyone creates a different repository, but everyone will work on the same). When creating a project, check the ''Initialize repository with a README" to start a master branch.

    - The member who create the project need to add other team members to the project with the role of Developer(**Project -> Members**).

    - All team members are cloning the repository on their local devices.

- Every team member will work on a different part of the website, so in the end, every part will make the hall site complete. Before you start with work (writhing code) team should decompose the problem into much smaller problems, each of these small problems will be created as an Issue in Gitlab (**Issues -> Newissue**).

    - ❏ **e.g. -** Create Nabar, Create Hero Banner, Optimize Mobile, Add Meta Tags ...

01

- Once the team has created all the tasks that will complete the project, each team member assigns a task to themselves and then creates a branch where is going to work.

- Along the way, he can make as many commits as he thinks that he needs. When the task is complete, the working member is making "push" on the branch where the task is, sending the branch to Gitlab.

- Then, on Gitlab, it opens a Merge Request (**Merge Requests -> New**) ,from its branch to the master branch.

- Once the merge request is open, the merging can be done by himself or asks the other team members to look inside the code and they can merge it if they think everything is ok (peer-review).
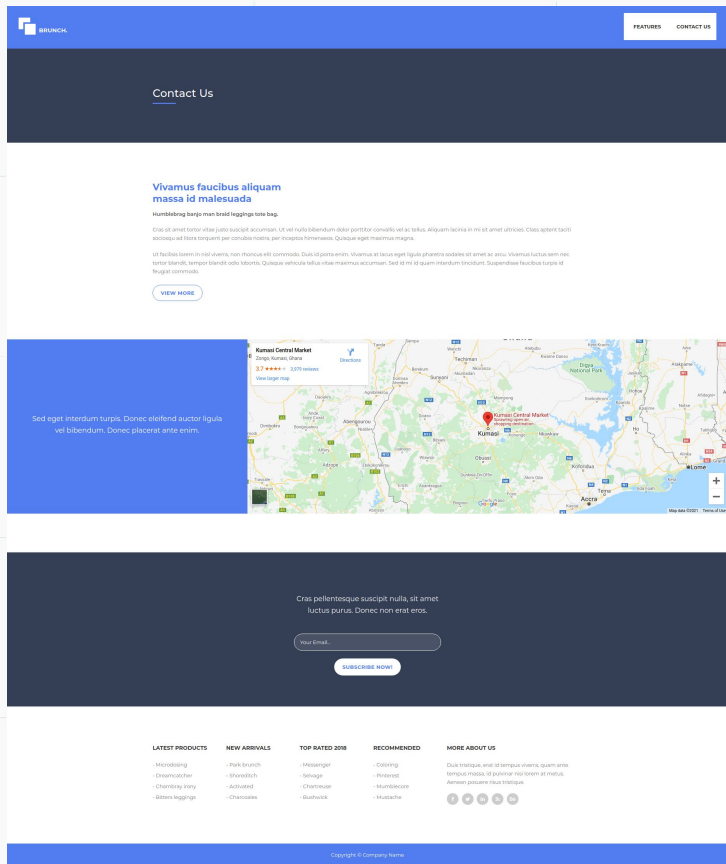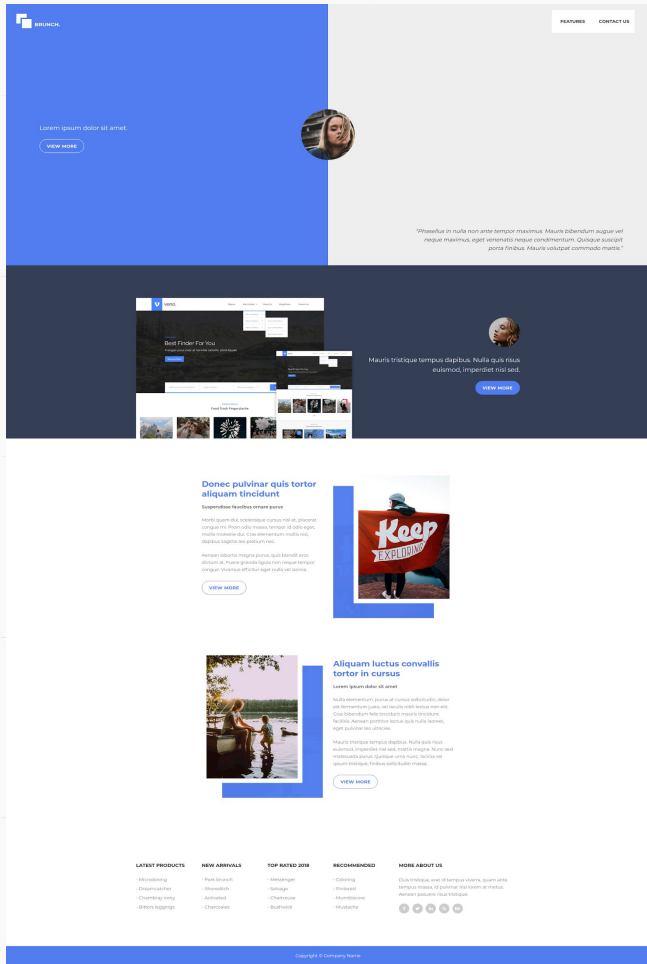
HINT:

❏   At the master branch, no one from the team ever commits directly.

❏   The merging on every branch to the master is making on GitLab.

01

❏ Conflicts are resolved on Gitlab if possible, or locally if not on Gitlab (if Gitlab cannot resolve a given conflict it will display the option - "Can not merge. Merge locally", and there will be instructions).

❏ The branches that are created for each task, after their merging to the master, delete them (both locally and on Gitlab).

01

4

Break a leg!

Please use the screenshots folder to see how the pages should look. If there is not enough time, try to finish the exercises at home. Feel free to use different images for the exercises if not provided.