Web Applications I / Applicazioni Web I – 2020/2021

Instructions for management and submission of BigLabs through GitHub Classroom

1   Goal

The laboratories submissions will be done by pushing your group project to a specific repository, that will be created by *GitHub Classroom*.

Note: this is the same mechanism that will be used for the final exams.

2   Basic Requirements

The basic requirement is having a personal GitHub account and being familiar with Git commands.
If you are new to Git or GitHub, you can have a look to the GitHub starter course.
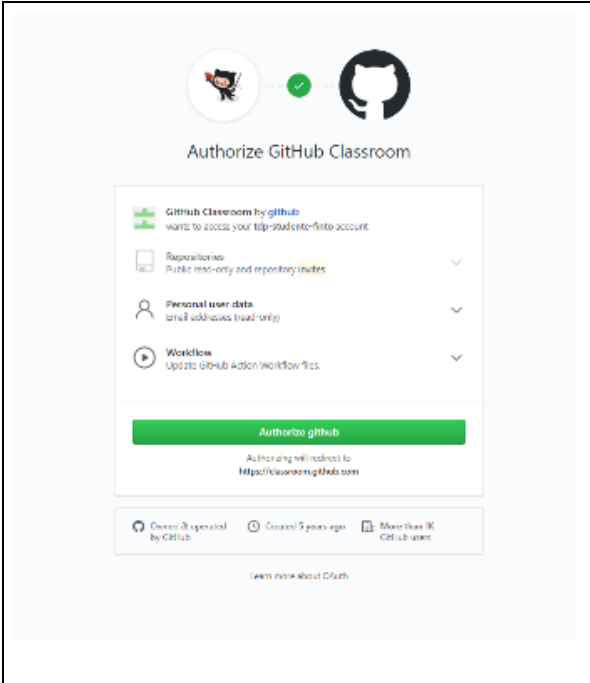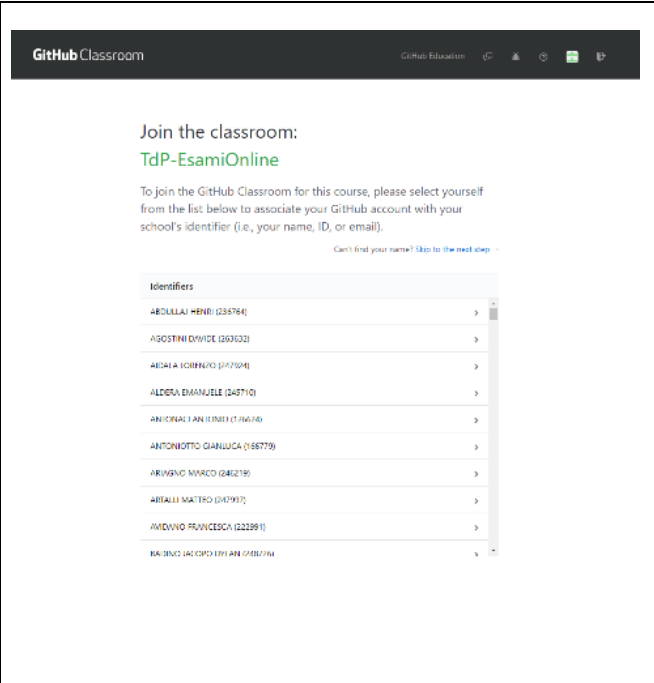
3   Accessing BigLabs

In the text of the BigLabs, you will find a link to access the GitHub Classroom's assignment. Following that link, you will have to login in your GitHub account.

4   First login

At the first login on GitHub Classroom, you must associate your GitHub user to your student's ID.
This is needed only once to join the 'Classroom'.

1. Authorize your GitHub user to access the Classroom platform.	2. Associate your account with your student identity, by selecting your name from the full list of students. If you are not on the list, please contact one of the teachers on Slack.

5   Accept the group assignment.

5.1   First BigLab: join (or create) your group.

Reach the assignment with the link provided to you in the text of the BigLab.

At your first login, you must check if one of your colleagues already created your team on GitHub Classroom. If yes, you must join your team clicking on the "join" button. In the example below, only a team called "*pippo*" was already created.

Otherwise, you must create your team using the form on the bottom of the page. It is mandatory to use the **same group name** specified in the Google Form filled at the beginning of the course. Please, **do not join** a wrong team.

PoliTo-AW1-MZ-2021

Accept the group assignment —
BigLab1

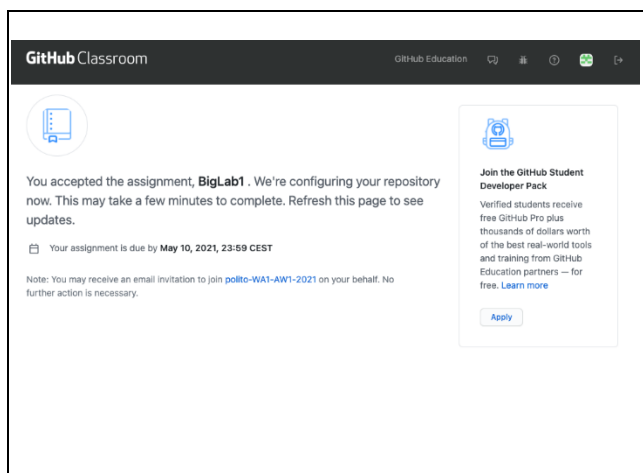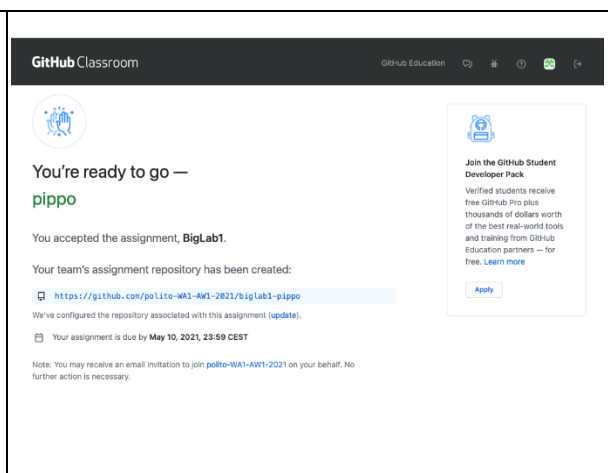Before you can accept this assignment, you must create or join a team. Be sure to select the correct team as you won't be able to change this later.

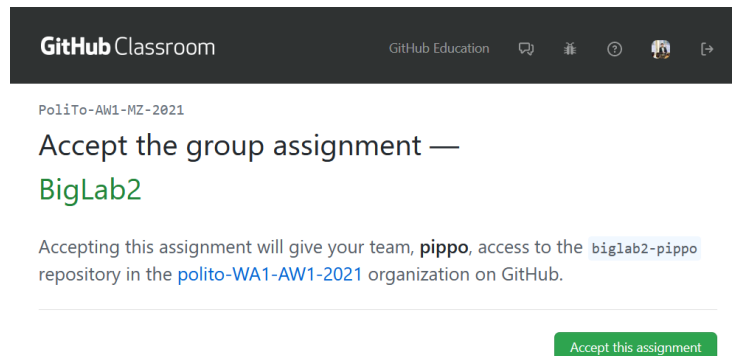**Join an existing team**

pippo 1   Join
student

OR Create a new team

Create a new team   + Create team

A) If you are the one that creates the team, you will see a message that your repository is under creation.	B) If you join an already created team, you are ready to go.

5.2   Second BigLab: accept the assignment.

For the second BigLab, you just have to "accept the assignment" to get a copy of the skeleton repository.

6   Your repositories

After creating or joining a team for each BigLab, you will get a link to the repository in which you have to work. The link contains a common part (the BigLab info), concatenated with your Team name.

Now you have your repository and may freely work on it. The repository is private, and only your team and the teachers may access it.

7   Developing the project

All the development of the project **must** take place inside this repository.

We suggest you create a **new branch** for each part of the BigLab, week by week, to keep track of your progress.

For instance, when you start working, you can create a branch called "*week1*" and commit your work on that branch. When you finish working on the first week, you merge everything into the main branch and move to a newly created branch "*week2*".

Please, notice that this is not mandatory; it is just a suggested developing methodology.

8   Submitting the project

- The final project must lie in the **master/main branch** (the content of other branches will not be considered or evaluated).
- Ensure that **all** the needed packages will be retrieved with `npm install`.
- The last commit (i.e., when you want to submit the whole BigLab), **must be tagged with 'final'** (see below for instructions).

We assume that, if your **master/main branch is empty**, or you did not push a commit tagged with **final**, you decided **not to submit** the BigLab for evaluation.

No further commits are considered after the BigLab deadline.

Note: for tagging a commit, you may use (from the terminal) the following commands:

```
# ensure the latest version is committed
git commit -m "...comment..."
git push

# add the 'final' tag and push it
git tag final
git push origin --tags
```

Alternatively, you may insert a tag from the GitHub web interface (click on "tags" and follow the link "Create a new release").