İTÜ İSTANBUL TECHNICAL UNIVERSITY

ASSIGNMENT 1

Department of Computer Engineering

BLG101E – Introduction to Information Systems – Fall 2020

Assignment 1

High-level Instructions

⚠ See page 3 for detailed instructions for each step ⚠

- **STEP 0 (by Nov. 5):**
  - **Create yourself a *GitHub* "repository" within our *GitHub* "classroom" with the "username" indicated below. Your username should have a specific format. Otherwise, we cannot link your GitHub account to you and record your grades.**
    - See the step-by-step instructions below for adding yourself to our *GitHub* "Classroom".

- **STEP 1 (by Nov. 10):**
  - **Collect information for your website. You will develop static a web site that promotes your high school.** Do not collect text verbatim. If your text is verbatim you must quote it and give a link or citation.
  - **Clone your *GitHub* repository to your computer using *Git* and play with editing the files and *committing* them.**

- **STEP 2 (by Nov. 20):**
  - **Create the (multiple, linked) *HTML* files for your website, so that you can view them on your local computer. Do not use any HTML editor, but type HTML file yourself. Also, you must not use any Javascript or similar code. In particular, your web site should feature the following:**
    - **✓ At least 4 different pages linked to each other.**
    - **✓ At least one table with at least 2 columns and 2 rows.**
    - **✓ At least one bulleted item list with at least 2 items.**
    - **✓ At least one multi-level list with at least 4 items.**
    - **✓ Basic structuring tags like body, heading, paragraph, etc.**
    - **✓ At least one image with an alternative text and size specified.**
    - **✓ Several hyperlinks.**
    - **✓ At least 4 different web pages with hyperlinks between.**
    - **✓ Text styles (e.g., font familes, font sizes, bold/italic, alignment, color, etc.).**
    - **✓ Non-text styles (background changes, borders, table or list formatting, etc.).**
  - **Using *Git, commit* these files and *push* this website to *GitHub* so that it can be viewed on the web.**

- **STEP 3 (by Nov. 25):**
  - **Add *CSS* styles to your website to give it an advanced look. Do not use any styling tool or framework. Type your CSS files yourself. Your web site should feature the following:**
    - **✓ CSS "classes" to style elements.**

        - ✓ CSS included both as an external style sheet and via a style attribute.
        - ✓ Web pages structured and styled using grouping elements.
    -
    - Using *Git, commit* the website files and *push* them to *GitHub* so that it can be viewed on the web, and validate it with a validator.

- STEP 4 (deadline: Nov. 30 at exactly 17:00):
    - Create a logo for your website from scratch using *Inkscape*. Incorporate it into the website so that it can be viewed from a browser.
    - Take (find) an existing photo of your *fictional* self and process it in *GIMP* to show off your *GIMP* skills. Put on the website both the original photo and the photo processed with *GIMP* for comparison. You will have an “About” page, and you may include these two photos on this page along with some information about yourself. Your web site should feature the following:
        - ✓ Images that have been spatially transformed (flipping, rotating, cropping, etc.) using GIMP or Inkscape.
        - ✓ Images that have been enhanced or color-transformed by GIMP.
        - ✓ Images that have had drawing applied to them using GIMP.
        - ✓ Images that have been manipulated using selections in GIMP.
        - ✓ Images that have had basic shapes or paths added by Inkscape.
        - ✓ Images that have had formats added to shapes or paths in Inkscape.
        - ✓ Images that have had paths edited by manipulating their nodes in Inkscape.
        - ✓ Images that have had objects edited with Inkscape’s transformation/align/distribution tools.

        For each of the above items please include both original and changed image in the repository (if the image was made from scratch, the original is not required).

    - *Push* this all to *GitHub* by the due date and time so that it can be viewed from the web.
        - ✓ Ensure the files are in the correct place so that your submission can be marked.

Submission Notes

- All steps of this assignment will be evaluated only after the final deadline has passed. However, it is recommended that you get the earlier steps done by their respective due dates.
- By the final deadline, you need to push your files to your GitHub classroom repository so that will be made available for marking.
- Also, submit to Ninova your grade sheet that you can find among Ninova class files Assignment1EvaluationCriteriaBLG101E.xlsx by the deadline. *You will evaluate your own assignment but this evaluation will be checked by the TA during demo.*
    - Use all of the document & image processing techniques shown in the evaluation form.
    - *If the lecture did not cover a particular technique listed on the evaluation form, it is your responsibility to learn how to do it.*

- **In order for your assignment to be considered for grading, attend your demonstration session, which will be announced in a separate schedule and will be after the due date of the assignment.**
- **Have fun.**

Detailed Instructions

Step 0: Create your GitHub Repository

Click on this link:
https://classroom.github.com/a/r5cmcNCm

A GitHub "repository" (a place to keep your web and programming files) will be created on the GitHub website under the "itu-itis" organization. You will have access to the repository.

Ensure that you join our *classroom* with your *classroom prefix* (i.e., ituitis20) and your *ITU account name*, that we can connect your GitHub account with your ITU account to assign your grades correctly. It is shown below how your username MUST be:

ituitis20-[youraccount]

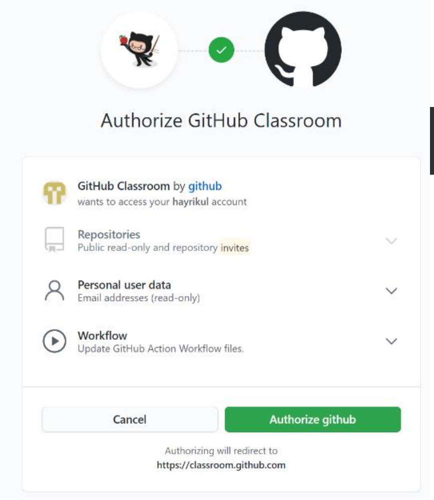
for example, if your name is "Mehmet Yılmaz" and your ITU account name is *yilmazmeh19*, then your github account name MUST be:

ituitis20-yilmazmeh19

Be careful. The prefix is always "ituitis20-". There is a single hyphen character at the end. After that, your exact ITU username that you are using to login to your email account.

If you make a mistake here or run into trouble choosing your Github account name mistakenly, you can create a new account from scratch. Then, inform us about the incorrect username so that we can remove it.

After creating your GitHub account, you may need to authorize GitHub Classroom to access your account and accept the assignment – the dialogues may be similar to the screenshots below (though the date may change according to the year of the course):



Authorize GitHub Classroom

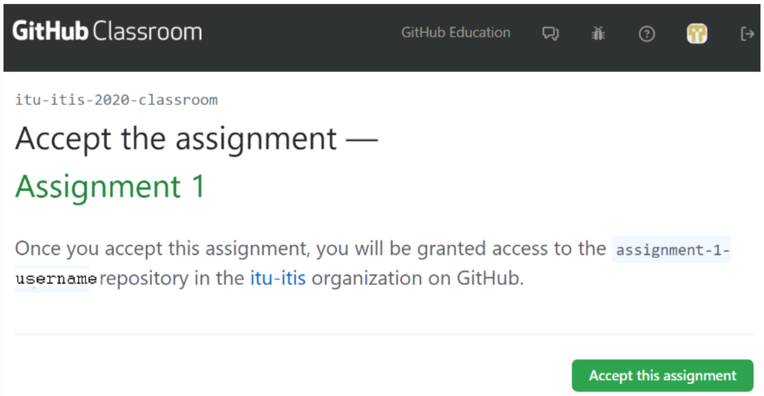
GitHub Classroom by github
wants to access your hayrikul account

Repositories
Public read-only and repository invites

Personal user data
Email addresses (read-only)

Workflow
Update GitHub Action Workflow files.

Authorizing will redirect to
https://classroom.github.com



itu-itis-2020-classroom

Accept the assignment — Assignment 1

Once you accept this assignment, you will be granted access to the `assignment-1-username` repository in the itu-itis organization on GitHub.

Step 1:

Collect information

Collect information for your website. The website will present information about your high school. The information will be in the form of text and pictures. DO NOT COLLECT TEXT VERBATIM. IF YOUR TEXT IS VERBATIM YOU MUST QUOTE IT AND PROVIDE A

LINK OR CITATION. (It is recommended not to use copy-paste at all and if you do, clearly state in your notes the text's origin so that you can quote it if you include it in your website).

Clone your repository

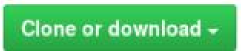
You need to “clone” (copy) your repository from GitHub to your own computer so that you can edit the files in it there. When you clone a repository, you copy all the files in it and you also have the ability to later “push” the files that you “add” and “commit” back to the repository you cloned it from.

In order to do this on Linux (for other systems, the approach may differ), the procedure is as follows:

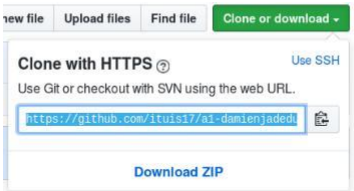
1. **You need to go to your repository page. For example, if my username were ituitis20-yilmazmeh19**, I would go to:

   https://github.com/itu-itis/assignment-1-ituitis20-yilmazmeh19

2. There you will be able to find a link to give to Git by looking for the “Clone or download” button:

3. **Clicking that button will reveal a link which you can copy (the icon on the right will copy it your clipboard):**

   Use Git or checkout with SVN using the web URL.

   https://github.com/ituis17/a1-damienjadedu

⚠ Don’t click “Download ZIP”! You need to “clone” your repository so you can later “push” to it. ⚠

4. **You will now use the “git clone” command in a terminal to clone the repository from that link. Open up a terminal console (for example, by clicking on the programs menu and searching for “terminal” and clicking to run the program that comes up – e.g. *xfce4-terminal*). If the link that you retrieved were https://github.com/itu-itis/assignment1-sandikkaya.git** then you would type into the terminal the following command:

```
git clone https://github.com/itu-itis/assignment-1-sandikkaya.git
```

Git will now create a directory called **assignment-1-yourusername** where you “yourusername” would be replaced with your GitHub username. For example, if your Git username is ituitis20-yilmazmeh19, it would create a directory called **assignment-1-ituitis20-yilmazmeh19**. In that directory is just one file called “index.html”.

⚠ You can move this directory into other directories on your computer’s file system without breaking anything. All the configuration files relating to the repository are stored in a hidden folder called “.git”. ⚠

Step 2

Create the HTML files for your website

Using the browser, text editor and file manager of your choice, create the HTML files for your high school website and commit them to your repository.

Important points:

➤ The website should include a banner on the very top of the page that states that the name of your high school and the website is prepared as a demonstration for a homework.

➤ The assignment may be done on any operating system with any browser but the website will be viewed by the markers using the software tools specified for the course.

➤ The work of creating the website should be done on your own computer, not on *GitHub*. The website should be designed so that it can be viewed both on your computer ("locally") and via *GitHub*. So use "relative links" when making hyperlinks between your web pages.

➤ The repository you create should contain at least a file called index.html, which is the home page for your website.

➤ Make sure to create multiple HTML files (at least 4) with links between them. When you click on a link, the correct file should be loaded, whether the web site is hosted on your computer or on GitHub.

➤ You can view your website while it is on your computer ("locally") by loading it into your browser.

> **For this purpose you can use the "File... Open File..." capability of your browser, navigating to the index.html file and selecting it.**

➤ Check the evaluation criteria for this assignment so that you know at least the minimum HTML capabilities that you need to use for this assignment, and be prepared to explain to the demonstrator where you used these in your website.

➤ Extra information about HTML and websites can be found from the tutorial website http://w3schools.com/.

➤ The following textbook is the suggested reference. The section "HTML Basics" is relevant to this step:

> **http://interactivepython.org/runestone/static/webfundamentals/index.html**

➤ The website should be written in valid HTML according to the "HTML 5" specification.

Upload to GitHub and GitHub pages

Using the method described in Step 1, commit your changes and new files to your repository, and push them to your GitHub repository. We have enabled a capability of GitHub on your GitHub repository called "GitHub Pages" that will allow anyone on the Internet to view the website in your repository from a browser if they know the address of the website.

If your GitHub username is ituitis20-yilmazmeh19 then you can find the website being served from your repository at the address:

> **https://itu-itis.github.io/assignment-1-ituitis20-yilmazmeh19/**

Validate your website

Navigate to https://html5.validator.nu/ and in the "Address" textbox section type in the address to your website as above.

Click "Check". If there are any errors, fix them on your computer, add the files that were changed to the Git staging area, commit them and push the repository.

Step 3

Use the knowledge you have learned during lectures or from http://w3schools.com/ to style your web site to make it more visually appealing. Use *external* stylesheets (.css files and <link rel ...> tags).

As described in the previous step, push the result to GitHub and validate the HTML as HTML 5 using https://html5.validator.nu/.

Further points:

➤ **In the Fundamentals of Web Programming textbook, "Cascading Style Sheets" is the relevant section:**

http://interactivepython.org/runestone/static/webfundamentals/index.html

➤ **Fill out the evaluation form and check the evaluation criteria for this assignment as you are preparing it so that you know at least the minimum CSS capabilities that you need to use for this assignment, and be prepared to explain to the demonstrator where you used these in your website and how they work.**

Step 4

This step has a deadline.

1. **Use the *Inkscape* program to create a logo for your website. Ensure that you check and fill out the evaluation document so that you use at least the capabilities of *Inkscape* described there.**
2. **Add the logo so that it is visible on the website.**
3. **Take (find) an existing photo of your *fictional* self and process it in *GIMP* to show off your *GIMP* skills. Put on the website both** the original photo and the photo processed with *GIMP* for comparison. You will have an "About" page, and you may include these two photos on this page along with some information about yourself.
4. **Add both the original photo and the processed photo to your repository. Ensure they are linked within your website.**
5. **Again commit all the files, and push the website to GitHub and validate it.**

⚠ If you get the website pushed to GitHub in the correct location by the due date and time, it will be automatically saved for marking.