



CHIANG MAI UNIVERSITY

Bachelor of Science (Software Engineering)

College of Arts, Media and Technology

2nd Semester / Academic Year 2020

SE 234 Basic Development and Operations

Lab I – Basic GIT

Objective In this session, the students will learn how to use Git and Git basic operations. At the end of this class, student should be able to apply version control for their task executed locally in their computers.

0. Setting

0.1. Download today's lab resource

- A simple responsive WWW template – mod_iamx.zip from the MS Teams.

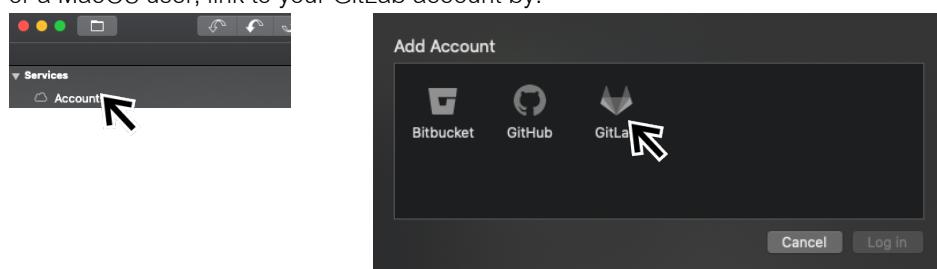
0.2. Register an account in Gitlab.com and login to your account

0.3. Find the latest version of the following applications and install them in the PC you are going to use.

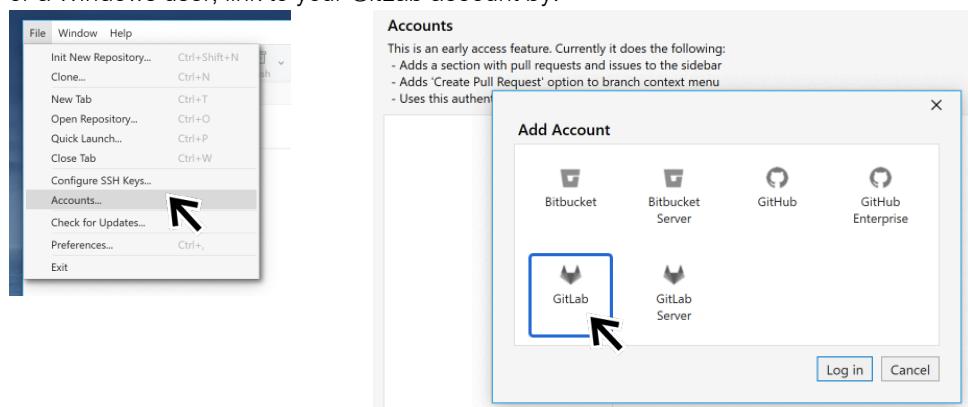
1. A GUI tool

- Install Fork - <https://git-fork.com/home>

- For a MacOS user, link to your GitLab account by:



- For a Windows user, link to your GitLab account by:



Then, login to your account.

- Fork will ask for a personal access token. You will have to create it in the GitLab website.

- Under user – setting select Access Tokens

The screenshot shows the GitLab User Settings page. At the top right, there are links for Projects, Groups, and User Settings. Below that is a sidebar with options like Profile, Account, Billing, Applications, Chat, and Access Tokens (which is highlighted with a red arrow). The main content area shows a message about CI/CD and a 'Settings' link.

- Fill the following information and click create.

Add a personal access token

Enter the name of your application, and we'll return a unique personal access token.

Name	<input type="text" value="SE234"/>
Expires at	<input type="text" value="2021-12-02"/>
Scopes	<input checked="" type="checkbox"/> api <small>Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.</small> <input type="checkbox"/> read_user <small>Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.</small> <input type="checkbox"/> read_api <small>Grants read access to the API, including all groups and projects, the container registry, and the package registry.</small> <input checked="" type="checkbox"/> read_repository <small>Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.</small> <input checked="" type="checkbox"/> write_repository <small>Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).</small>

- After clicking the form submission button, you will get you access token. Copy it and paste in the Fork application. Now, you should be able to log in via Fork and see your online status after that.

The image shows two side-by-side sign-in dialogs. On the left is the Mac OS version, which has a dark theme and a 'Sign In' button highlighted with a red arrow. On the right is the Windows version, which has a light theme and also has a 'Sign In' button highlighted with a red arrow.

The screenshot shows the Fork application interface. At the top, it displays a user profile picture and the URL <https://gitlab.com/Passakorn.p>. Below that is a navigation bar with 'Account' and 'SSH Keys' tabs. The 'Account' tab is selected. It shows 'Account Information' with 'Server Type: GitLab', 'Authentication Type: Personal Access Token', 'Username: Passakorn.p', and 'Status: Online'. To the right, another section shows 'Server Type: GitLab', 'Username: kong-se', and 'Status: Online'.

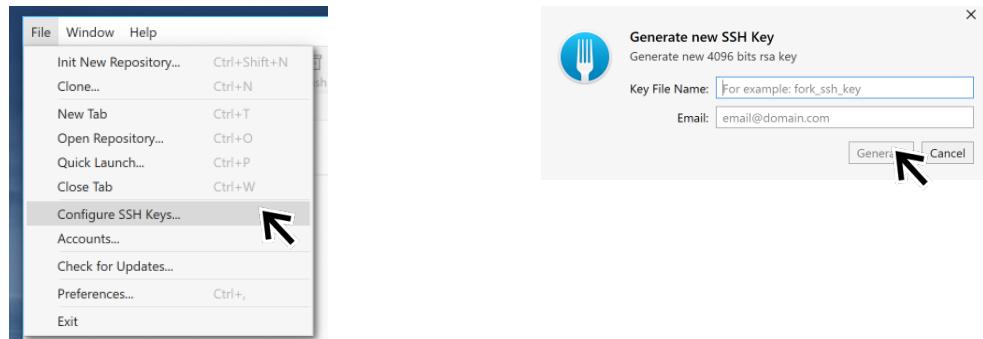
- Next is about security matter, we have to register the SSH keys of the specific computer we want to synchronize with GitLab.

- For a MacOS user, click Register under your account and generate a new key. Make sure that you use the same email as that you registered the GitLab account.



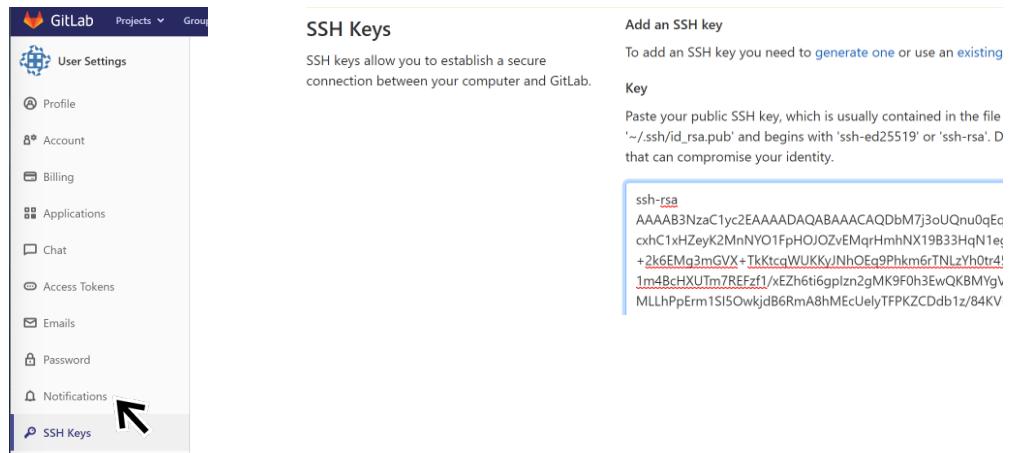
After the key is generated, copy the key content to clipboard and prepare to paste it in the GitLab website.

- For a Windows user, click Register under your account and generate a new key. Make sure that you use the same email as that you registered the GitLab account.



After the key is generated, copy the key content to clipboard and prepare to paste it in the GitLab website.

- Click the SSH Keys menu item, paste the copied key content in the largest textbox, choose an expired date (maybe a year from now), and click the Add key button



- The setup for Gitlab and the Fork application is now completed.

2. The Git engine.

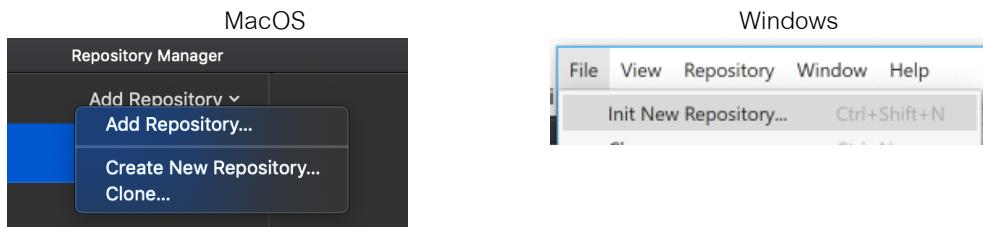
- For a MacOS or an Ubuntu user, Git is already preinstalled with the OS, so you do not have to do anything for this step.
- For a windows user, you need to download a git engine for Windows and install it.
 - <https://git-scm.com/download/win>

3. Text editor

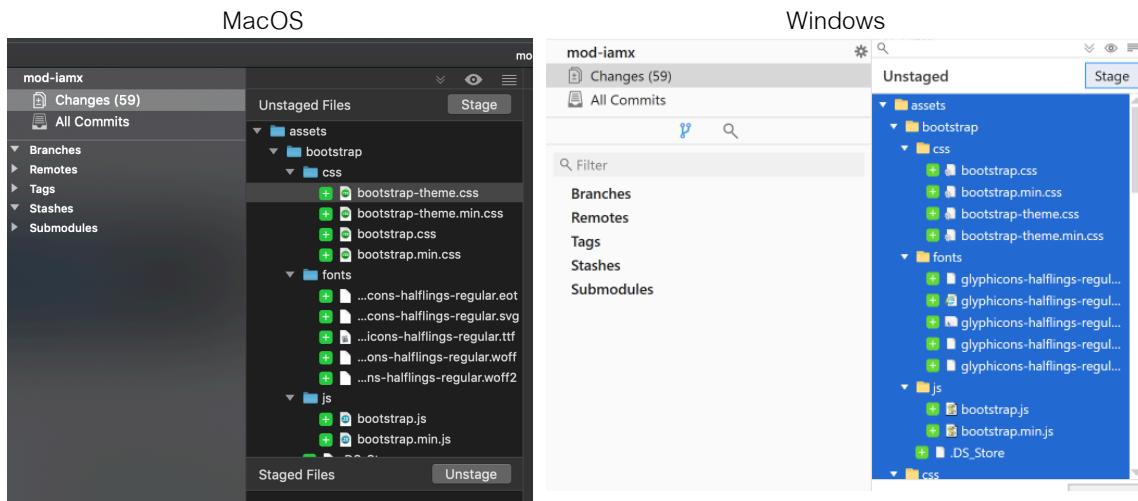
- Use anything you are familiar with :)

1. Repository initialization

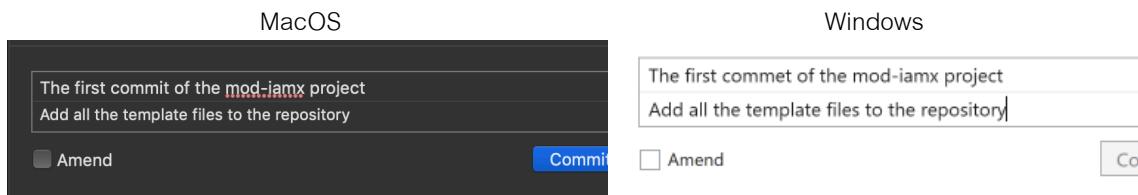
- 1.1. Extract the mod_iamx.zip file to a folder name mod_iamx under your project home folder.
- 1.2. Create a new repository and select the path you extract the mod_iamx project.



- 1.3. Click Changes. Then, drag all the files under the Unstaged files tab, and click stages. This will prepare the list of file we are going to store in this very first revision of our repository.

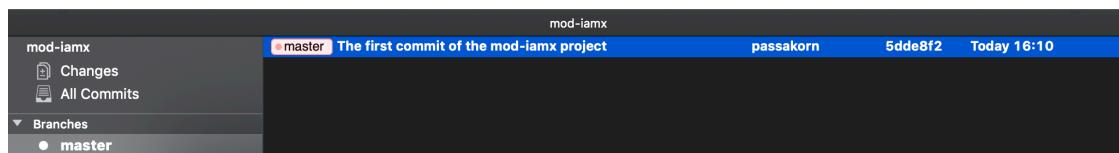


- 1.4. Make the first commit. Note that, the commit message must be added to every single commit. Ensure that all the files have been staged.



- 1.5. Click Branches->master on the right, you will see the committed log.

- MacOS



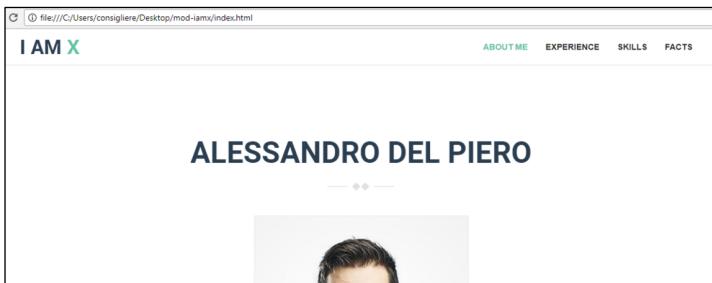
- Windows



- 1.6. Edit the index.html file in the mod-iamx project using a text editor by changing the name to one of your favorites or known soccer player. After that, see the result on the web browser.

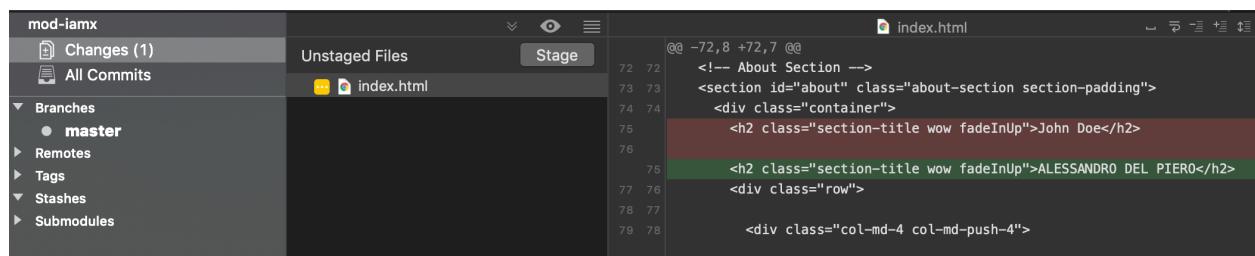
For example,

```
72 <!-- About Section -->
73 <section id="about" class="about-section section-padding">
74 <div class="container">
75 <h2 class="section-title wow fadeInUp">ALESSANDRO DEL PIERO</h2>
76
77 <div class="row">
```

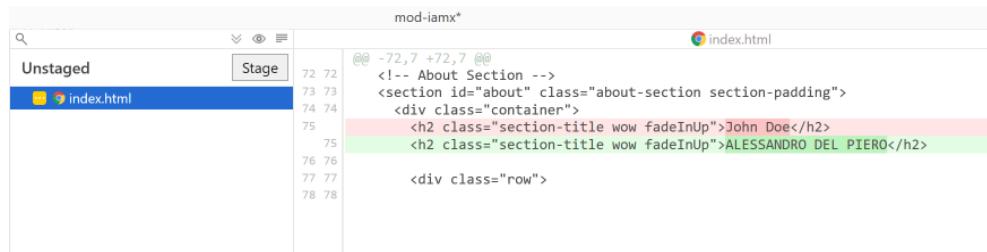


- 1.7. At this step, if we observe the #changes in the Fork application, we should be able to see that there is a change made in the index.html file, along with the details about modified lines of code.

- MacOS

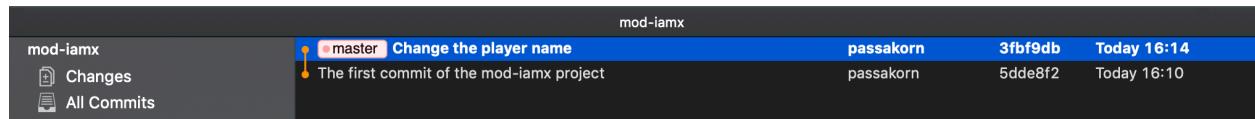


- Windows

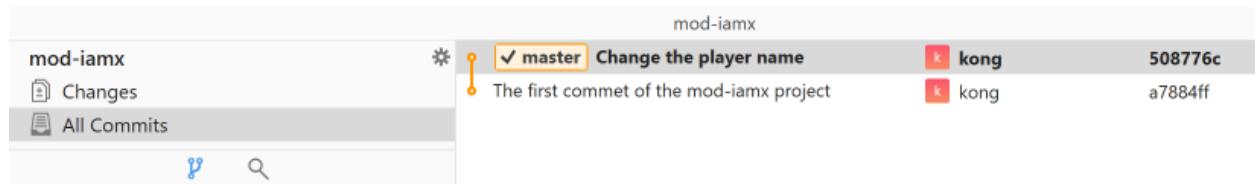


- 1.8. Stage the index.html, add the commit message, such as "change the player name", and commit it

- MacOS



- Windows



- 1.9. Explain what is meant by "master", "head", and "commit message" and show the current progress.

2. Workflow and branching

2.1. Edit the profile of your selected athlete based on the information available at <http://www.futhead.com/>

- Nationality
- Date of birth
- Height
- Club and league if exists

For example,

ALESSANDRO D		
	Alessandro Del Piero (CF)	90
	Alessandro Del Piero (CF)	87

2.2. Commit this version. Make sure that you provide some meaningful commit message.

2.3. Look up for the profile of your **selected athlete** in the Wikipedia, e.g.,

https://en.wikipedia.org/wiki/Alessandro_Del_Piero and add the profile associating senior career to the

Experience section of the template. The result will be like --



2.4. Commit this version with the commit log indicating the completion of the experience section. Show the staff that your git tree has already logged 4 commits.

Staff's Signature _____ (2)

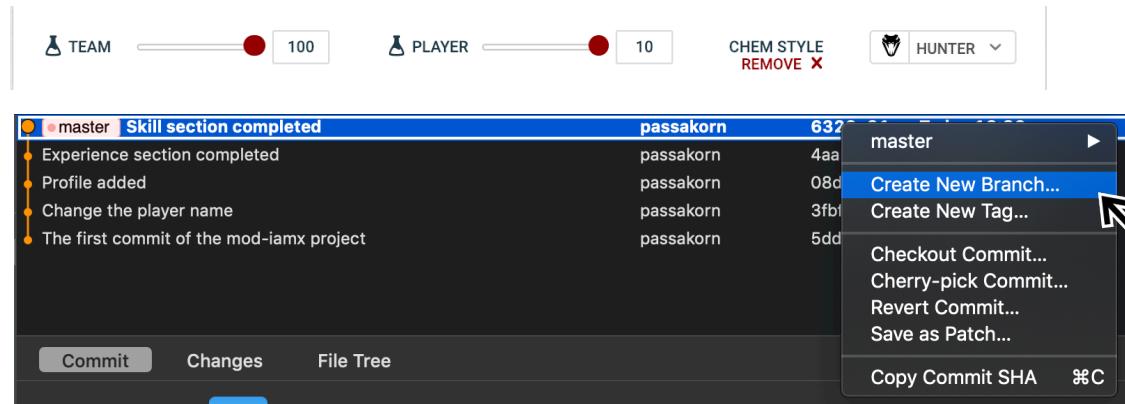
- 2.5. Edit the skill section based on the stats shown in <http://www.futhead.com/>, which are the values of Pace, Shooting, Passing, Dribbling, Defending, and Physical. The following is the example retrieved from Alessandro Del Piero.



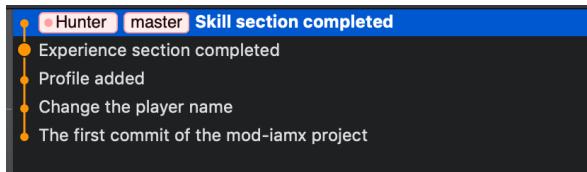
- 2.6. Commit this version with its corresponding commit log and show the staff everything you have done up to now.

Staff's Signature _____ (3)

- 2.7. Suppose that you feel that the basic stats may not appropriately represent your favorite player, so, you change the chem style parameter to Hunter. However, you are not so sure about the new stats. Since you have already studied about version control, you have a chance to simply create a new branch associating the Hunter stat. This will allow you to switch back to the Basic stat version in the future.



- 2.8. Name the new branch as Hunter. Now your git tree will be like:



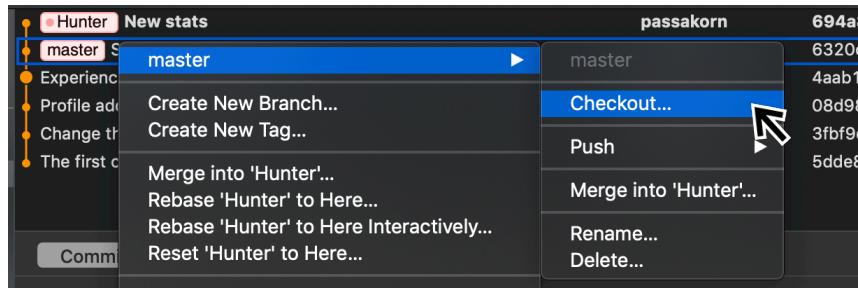
- 2.9. Edit the stats regarding the Hunter-type stat and commit the version. Explain the staff about your current git tree.

	passakorn	694a817	Today 16:33
Hunter New stats			
master Skill section completed	passakorn	6320c61	Today 16:30
Experience section completed	passakorn	4aab147	Today 16:30
Profile added	passakorn	08d9873	Today 16:29
Change the player name	passakorn	3fbf9db	Today 16:14
The first commit of the mod-iamx project	passakorn	5dde8f2	Today 16:10

Staff's Signature _____ (4)

2.10. See the page on a browser that the stat is currently the Hunter version. Now, let's simply switch back to the master version:

- Right click at master and then



- Refresh the browser and see the difference.

2.11. Now switch back to the Hunter branch

2.12. Add contents to the Fact section based on anything you like in the Wikipedia page.

2.13. Change the icons using the icon list from

https://www.w3schools.com/icons/fontawesome_icons_webapp.asp



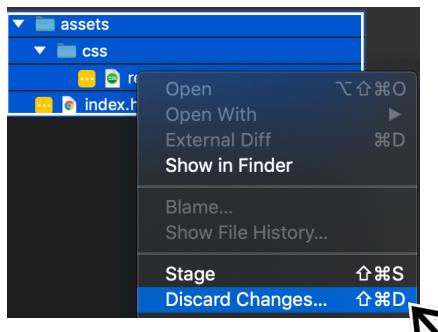
2.14. Commit the version and tell the staff whether the changes made to the New stats commit are also appeared in the master branch.

Hunter	Add facts	passakorn	0493fd8	Today 16:38
New stats		passakorn	694a817	Today 16:33
master	Skill section completed	passakorn	6320c61	Today 16:30
Experience section completed		passakorn	4aab147	Today 16:30
Profile added		passakorn	08d9873	Today 16:29
Change the player name		passakorn	3fbf9db	Today 16:14
The first commit of the mod-iamx project		passakorn	5dde8f2	Today 16:10

Staff's Signature _____ (5)

3. Rollback and cherry picking

- 3.1. If you feel that you should reduce the whitespace between all the section headers and any of its related contents.
 - Try modifying style.css file.
 - Hint: It will be simple if you remember the CSS box model?
- 3.2. If you unintentionally remove the last 20 lines from the code and save the file, ignore the change and your project will be the same as that of the latest commit.

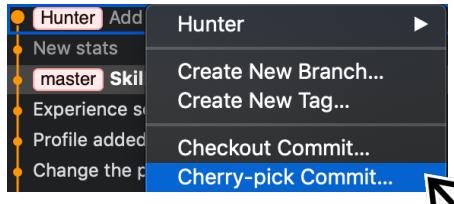


- 3.3. Try use the command once and show the staff your current git tree

Staff's Signature _____ (6)

- 3.4. Suppose that you show your page to your friend and everyone opposes the Hunter skill set. You have already made many updates to the Hunter branch. One of the easiest ways to apply all the changes made to the Hunter branch to the main branch for this situation is to use the Cherry pick command.

- First check out the master branch
- Right click to the latest commit of the Hunter branch and select Cherry pick commit



- Apply the changes and check the web browser if the 100% skill stat is shown alongside with the latest updated Facts section.

3.5. Commit the changes

- At the time you commit, observe the diff log. It should combine the changes from the previous version with the cherry-picked changes.

3.6. Check if your git tree looks like the one below. Then, explain the junction-like line to the staff

Staff's Signature _____ (7)