**Markdown for 07/13 Moldir Polat**

**(1) Learning Github and gitignore**

Git Workflow

1. Create repository
2. Fork it
3. In cmd input: git clone link

cd repository name

git checkout –b update-readme

<changing readme file>

git status

git commit –a –m “message”

git remote

<output origin>

git push origin update-readme

1. Refresh browser (new message in github)
2. Click compare&pull requests
3. Write -> create pull request
4. Go to File changed 🡪 can comment each line
5. In conversation Merge
6. Cmd: git checkout master

git branch –D update-readme

git remote add upstream link (central)

git fetch upstream

git merge –ff –only upstream/master

Gitignore

Git sees every file in your working copy as one of three things:

* tracked - a file which has been previously staged or committed;
* untracked - a file which has not been staged or committed; or
* ignored - a file which Git has been explicitly told to ignore.

Ignored files are usually build artifacts and machine generated files that can be derived from your repository source or should otherwise not be committed. Some common examples are:

* dependency caches, such as the contents of /node\_modules or /packages
* compiled code, such as .o, .pyc, and .class files
* build output directories, such as /bin, /out, or /target
* files generated at runtime, such as .log, .lock, or .tmp
* hidden system files, such as .DS\_Store or Thumbs.db
* personal IDE config files, such as .idea/workspace.xml

Ignored files are tracked in a special file named .gitignore that is checked in at the root of your repository. There is no explicit git ignore command: instead the .gitignore file must be edited and committed by hand when you have new files that you wish to ignore. .gitignore files contain patterns that are matched against file names in your repository to determine whether or not they should be ignored.

$ git status (check status)

$ touch .gitignore

# Input to .gitignore

## ignore all dir files

## ./dirname

## ignore the certain filr

## filename

## ignore the file wih the certain extension

## \*.extension

$ git add .gitignore

$ git status --ignored

**(2)** Crawled data from new website, because the link where all jobs are listed does not have job description; this part is in the job link itself, that’s why have to change the logic of the functions; not completed yet;

Problem: because one function should parse all jobs references from one website, and other parses respective job reference’s details, it is a little bit complicated; code has errors; need to be completed

**Some notes for database attributes**

Job\_name – Rec\_Title

Money – Rec\_Salary

Job-wel –

Company -

Position\_type- Rec\_Cate

Xueli-Rec\_Education

Jingyan-Rec\_Experience

Address-Rec\_Location

Description – Rec\_Desc