

The background is a dark blue gradient with a subtle pattern of small white dots. Overlaid on this are several faint, light blue geometric elements: concentric circles, arcs, and dashed lines. Some of these elements have degree markings, such as 40, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, and 260, arranged in a circular fashion. There are also small white arrows pointing in various directions.

GIT

BY MOHAMMAD HASAN KHODDAMI

GIT

In this powerpoint we are going to learn GIT in 2 parts

First → learn GIT from the website and how to use Github.com

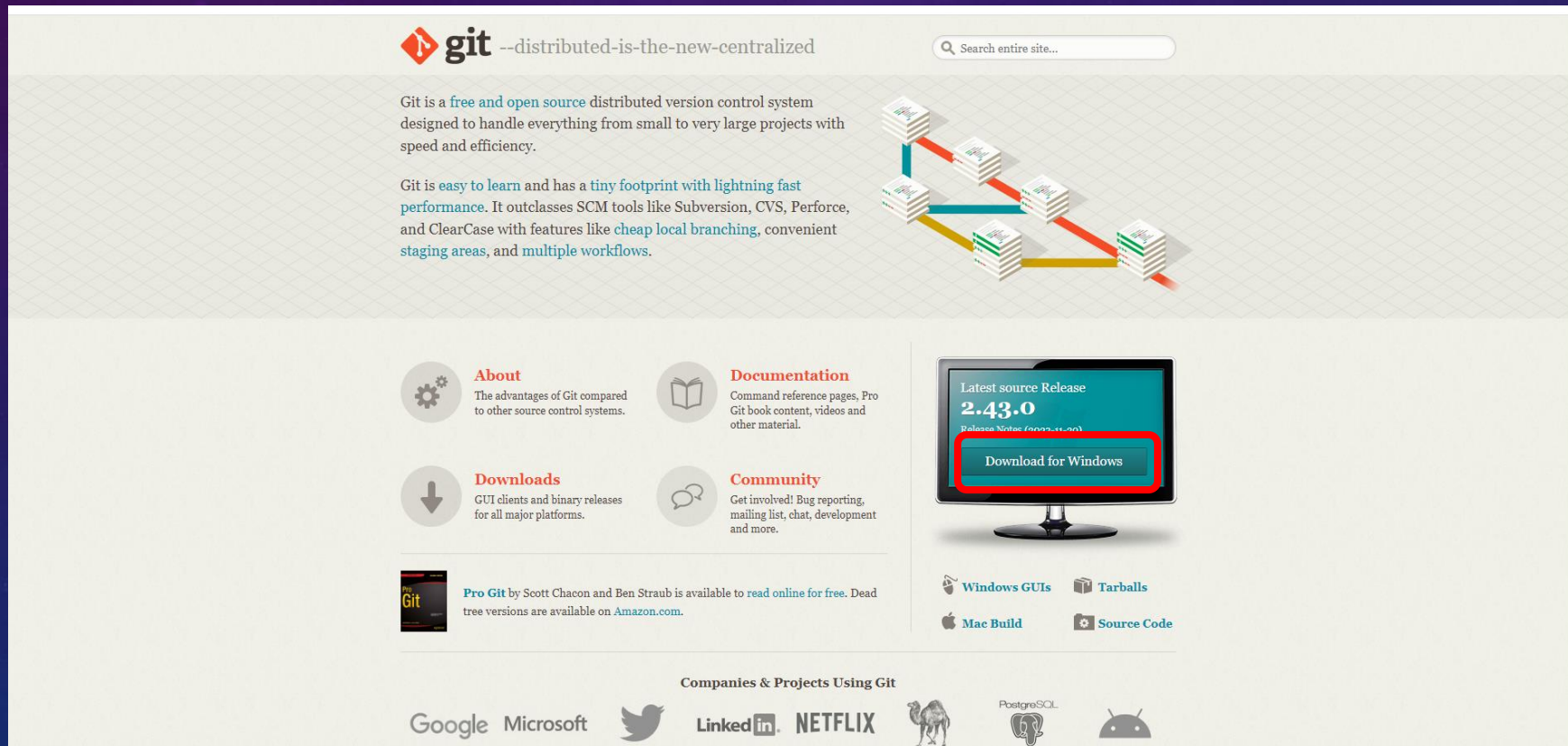
Second → Learn GIT-bash , using GIT in VSCode and Git commands

GIT VS GITHUB

- It's concept that you should understand, GIT is a source code that we can use it for GIT command !
 - Let explain it in an example : for using GitHub we have to know the GIT commands!
 - It will be more clear in continue of this session

DOWNLOADING GIT

You can download it from <https://www.git-scm.com/>



The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--distributed-is-the-new-centralized". A search bar is located on the right. The main content area describes Git as a free and open source distributed version control system. To the right of the text is a diagram illustrating Git's distributed nature with multiple repositories connected by lines. Below this, there are four sections: "About", "Documentation", "Downloads", and "Community", each with a brief description. On the right side, there is a section for the "Latest source Release" (2.43.0) with a "Download for Windows" button highlighted by a red rectangle. Below this, there are links for "Windows GUIs", "Tarballs", "Mac Build", and "Source Code". At the bottom, there is a section for "Companies & Projects Using Git" featuring logos for Google, Microsoft, Twitter, LinkedIn, Netflix, PostgreSQL, and Android.

git --distributed-is-the-new-centralized

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.43.0
Released: 2023-11-09

Download for Windows

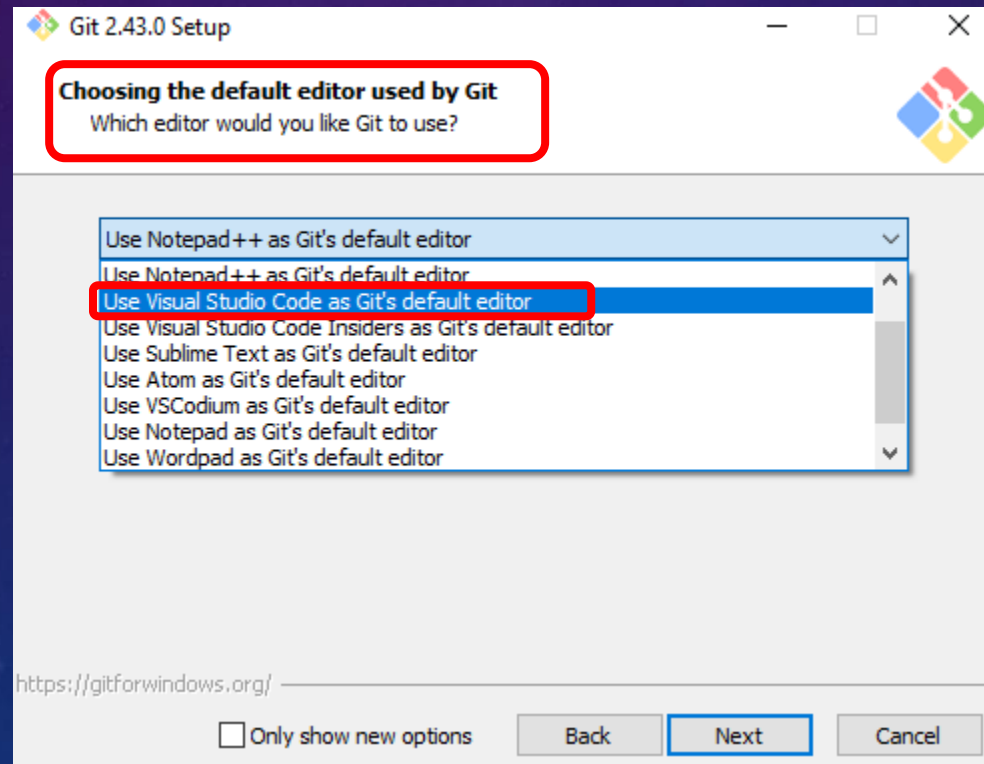
Pro Git
Pro Git by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Windows GUIs **Tarballs**
Mac Build **Source Code**

Companies & Projects Using Git
Google Microsoft Twitter LinkedIn NETFLIX PostgreSQL Android

INSTALLING GIT

- Installing GIT is pretty easy just press on next but there is just one note that you should pay attention
- In install setting GIT asking you for a your text editor you have to set it on VS Code



GITHUB.COM

The screenshot shows the GitHub homepage with several annotations:

- Repositories list:** A red box highlights the user's repositories: "ShakibSanatgar/MHSA-WEBSITE" and "mohammadkhoddami/mohammadkh...".
- Start a new repository:** A red box highlights the "Public" and "Private" options. The "Public" option is selected, with the description "Anyone on the internet can see this repository".
- Create a new repository:** A green button labeled "Create a new repository" is highlighted.
- For start project:** A purple arrow points from the "Create a new repository" button to the "Latest changes" section, which lists recent updates and includes a "View changelog" link.

Other visible elements include the GitHub logo, navigation links (Home, Issues, Pull requests, Projects, Discussions, Codespaces, Explore, Marketplace), a search bar, and a sidebar with a "Start writing code" button.

Your repos:
means your
projects

If your project is public and
anyone can have access to your
codes chose public in the other
hand chose private

For
start
project

WHY DO WE USE GITHUB?

- A. For version control :
 - If we update our code and the employer want`s to the last version and ask you to take back code to last changes there is no need to code it again ! We use GIT, it can save the versions of our code
- B. Teamwork:
 - Imagine that you are working on office so you need teamwork ! You have to see what updates are added to code so that`s why you need to use GIT
- C. present your skill
 - GitHub is a big resource of codes and you can share your code with others and challenge your skill.

VERSION CONTROL

- In version control there is some note that we have to now it
- 0.0.0 → this is version from the app that doesn't exist ! But I need to explain it:

0.0.0

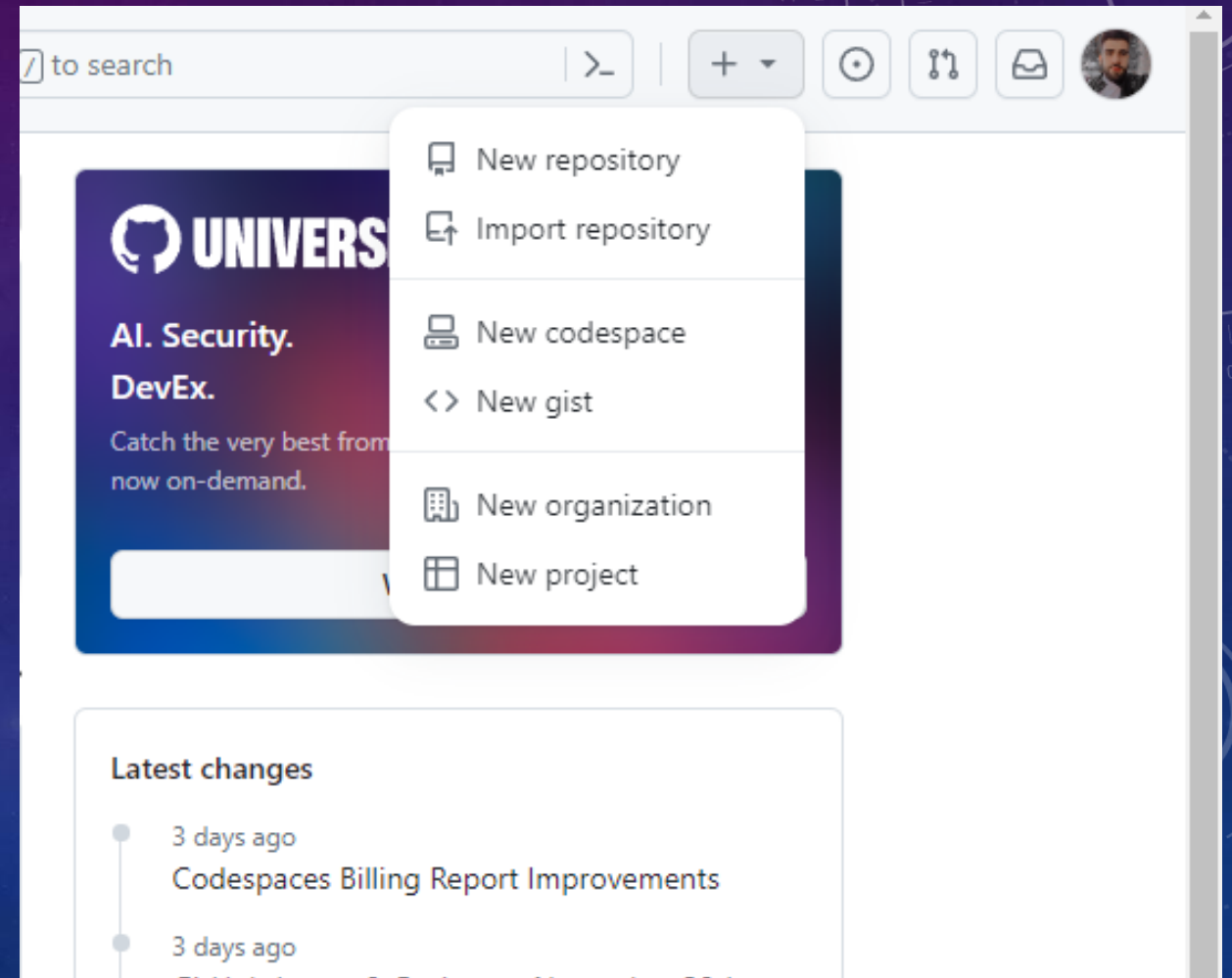
Fixing the bug

Adding a feature

Changing whole app that my
contains some bugs!

CREATE NEW REPOSITORY

Click on + and then create new repository to create a new project



CREATE NEW REPOSITORY

Repository name!

Description of your project

Private or public

Sure that fill this square
cause readme file one of
the most important files of
project

I will explain GIT ignore later

CREATE !

The screenshot shows the GitHub 'Create a new repository' page. It includes fields for 'Owner' (set to 'mohammadkhodami'), 'Repository name', 'Description (optional)', and 'Public/Private' options. There are checkboxes for 'Add a README file' and 'Add .gitignore', and a 'Choose a license' section. A green 'Create repository' button is at the bottom right. Purple arrows from the text on the left point to these specific elements: 'Repository name!' points to the repository name field; 'Description of your project' points to the description text area; 'Private or public' points to the radio buttons; 'Sure that fill this square cause readme file one of the most important files of project' points to the 'Add a README file' checkbox; 'I will explain GIT ignore later' points to the 'Add .gitignore' section; and 'CREATE !' points to the 'Create repository' button.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * mohammadkhodami / Repository name *

Great repository names are short and memorable. Need inspiration? How about [shiny-octo-fishstick](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

① You are creating a public repository in your personal account.

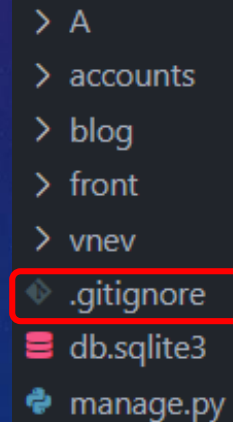
[Create repository](#)

GIT INIT

- If we have the files of our project and want to initialize it git we have to use git init command in vs code
- When you use git init command your files has to change their color to green.
- Test it by your self ! (then you have to push it and etc.)

.GITIGNORE

- GITIGNORE is for the files that we don't want to share it in GIT for example pycache , password files or etc.
- One of the most important files that have to put in gitignore is venv and database
- For creating Gitignore first we have to create a file which called .gitignore



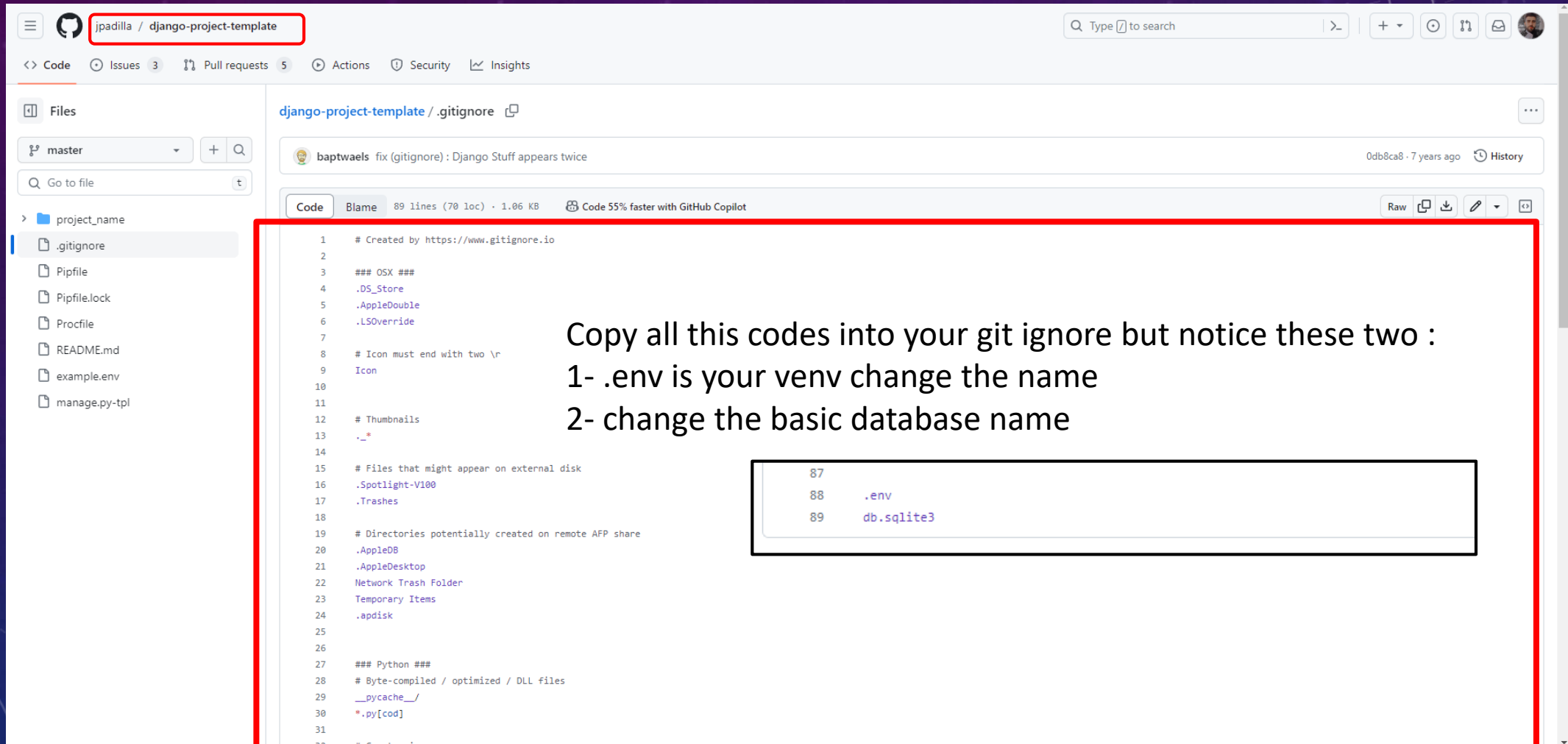
A screenshot of a file explorer window showing a directory structure. The files listed are: > A, > accounts, > blog, > front, > vnev, .gitignore, db.sqlite3, and manage.py. The .gitignore file is highlighted with a red rectangular box.

```
> A
> accounts
> blog
> front
> vnev
♦ .gitignore
▢ db.sqlite3
+ manage.py
```


HOW TO WRITING GIT IGNORE

- For using git ignore as I said we have to put our database, venv, pycache and etc. but how ?
- There is 2 ways for inserting a file to .gitignore
- A.
- Search your language program on google + gitignore template then open the Github and copy paste it into your .gitignore file

DJANGO EXAMPLE



The screenshot shows the GitHub interface for the repository `jpadilla / django-project-template`. The `.gitignore` file is selected, showing its content. A red box highlights the file content, and a text overlay provides instructions on how to use it.

Copy all this codes into your git ignore but notice these two :

- 1- `.env` is your venv change the name
- 2- change the basic database name

```
1 # Created by https://www.gitignore.io
2
3 ### OSX ###
4 .DS_Store
5 .AppleDouble
6 .LSOVERRIDE
7
8 # Icon must end with two \r
9 Icon
10
11 # Thumbnails
12 .*
13
14 # Files that might appear on external disk
15 .Spotlight-V100
16 .Trashes
17
18 # Directories potentially created on remote AFP share
19 .AppleDB
20 .AppleDesktop
21 Network Trash Folder
22 Temporary Items
23 .apdisk
24
25
26
27 ### Python ###
28 # Byte-compiled / optimized / DLL files
29 __pycache__/
30 *.py[cod]
31
32 # C extensions
```

Lines 87-89 are highlighted in the original image:

```
87
88 .env
89 db.sqlite3
```

2ND WAY

- Writing by yourself (not recommended):
- For put I file in git ignore use the following command
- `**/*.css` → this means every css files must not be upload on git
- `**/name_of_file.type_of_file` → put the specific file in git ignore
- For knowing more use following link
 - <https://git-scm.com/docs/gitignore>

SYNC GITHUB TO VSCODE

First click on
code then copy
it and we are
going to make
local repos in
our C drive

The screenshot shows the GitHub interface for a repository named 'DND1.3.5' by user 'mohammadkhoddami'. The repository is public and has 1 branch and 0 tags. The 'Code' button is highlighted with a red box, and the dropdown menu is open, showing the 'Clone' option with the HTTPS URL 'https://github.com/mohammadkhoddami/DND1.3.5.' highlighted with a red box. The repository description is 'This is my python final project which called DND , this is adaptatic'. The footer shows the GitHub logo and copyright information: '© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact'.

mohammadkhoddami / DND1.3.5

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

DND1.3.5 Public

Pin Unwatch 1 Fork 0 Star 0

main 1 branch 0 tags

mohammadkhoddami Initial commit

README.md Initial commit

README.md

DND1.3.5

This is my python final project which called DND , this is adaptatic

Go to file Add file <> Code

Local Codespaces

Clone

HTTPS SSH GitHub CLI

https://github.com/mohammadkhoddami/DND1.3.5.

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Code 55% faster with AI pair programming.

Start my free trial Don't show again

About

This is my python final project which called DND , this is adaptation from Dungeon And Dragon boardgame

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published
[Create a new release](#)

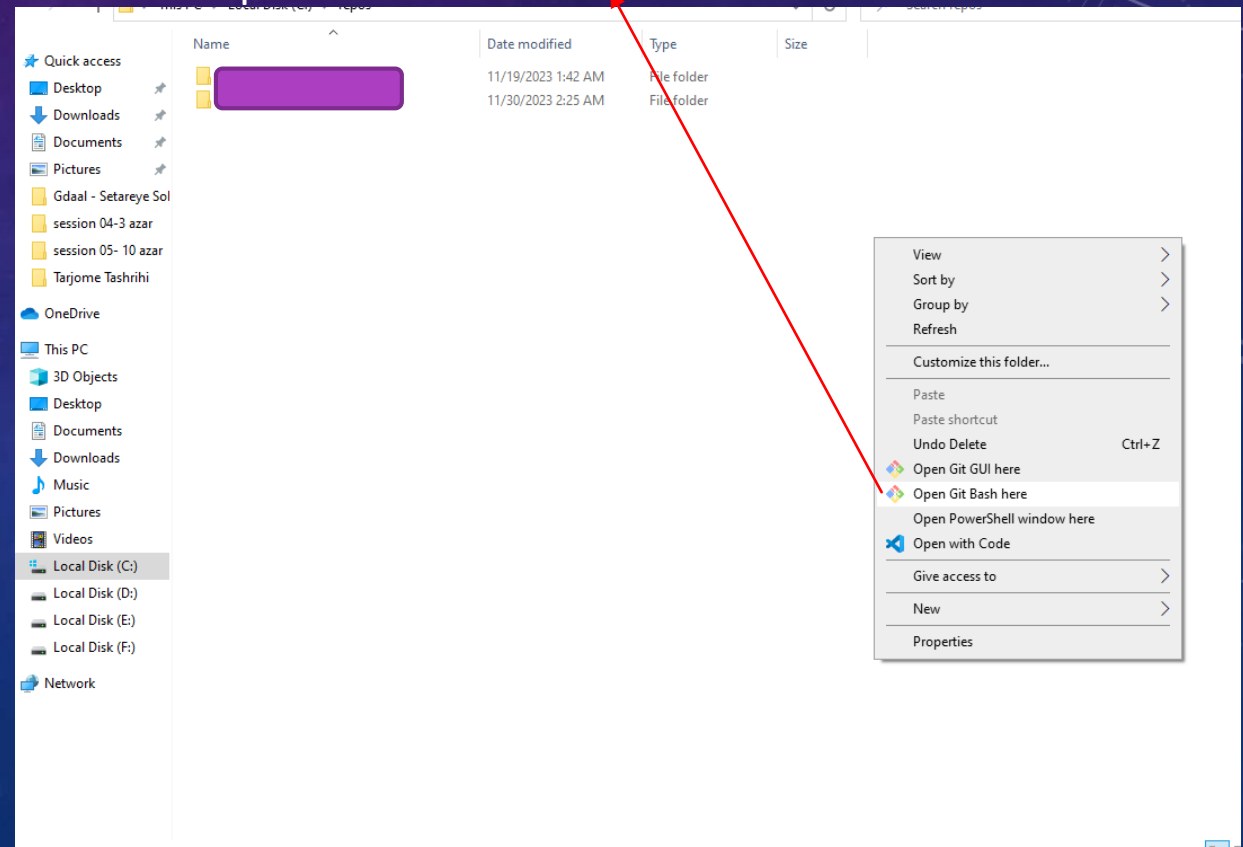
Packages

No packages published
[Publish your first package](#)

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact

CREATE LOCAL REPOS

- First create a new folder and name it whatever you want I chose repos
- Then open the folder and press Shift + right click and chose open GIT-BASH here



USE GIT-BASH

On GIT-BASH use following command
git clone + (the address that we copy that)
For paste the address use
CTRL + SHIFT + INSERT
Then you have to have a folder with the
project (repository) name

```
MINGW64:/c/repos
--no-tags                don't clone any tags, and make later fetches not to fo
low them
--shallow-submodules     any cloned submodules will be shallow
--separate-git-dir <gitdir>
                        separate git dir from working tree
-c, --config <key=value>
                        set config inside the new repository
--server-option <server-specific>
                        option to transmit
-4, --ipv4               use IPv4 addresses only
-6, --ipv6               use IPv6 addresses only
--filter <args>          object filtering
--also-filter-submodules
                        apply partial clone filters to submodules
--remote-submodules     any cloned submodules will use their remote-tracking b
anch
--sparse                 initialize sparse-checkout file to include only files
at root
--bundle-uri <uri>       a URI for downloading bundles before fetching from ori
in remote

niklaus@DESKTOP-V4GPV4N MINGW64 /c/repos
git clone https://github.com/mohammadkhoddami/DND1.3.5.git
```

GIT PUSH

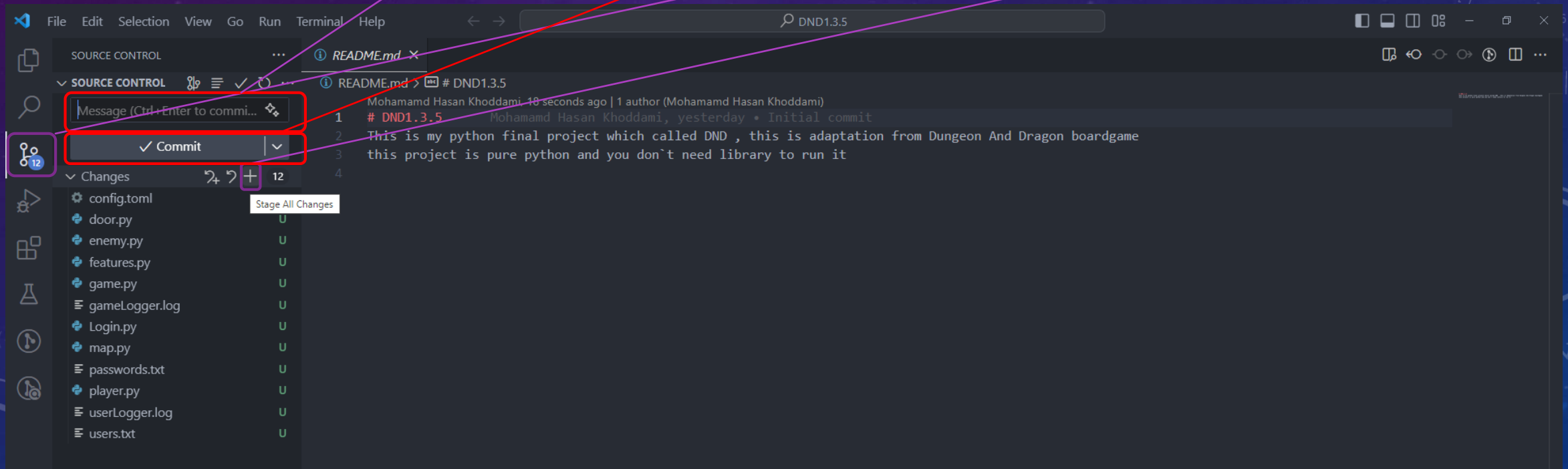
- Now I import my project files on the folder and use git push command

Don't forget to write message

Then press on commit to upload all files into your GIT-HUB

Click on source control

Click on (+) for confirm your changes



AFTER COMMITTING

- You have to code : git push to update your main branch

result

The screenshot shows a GitHub repository page for 'DND1.3.5'. The repository is owned by 'mohammadkhoddami' and has 3 commits. The file list on the left includes: Login.py, README.md, config.toml, door.py, enemy.py, features.py, game.py, gameLogger.log, map.py, passwords.txt, player.py, userLogger.log, and users.txt. The README.md file is selected, showing the title 'DND1.3.5' and a description: 'This is my python final project which called DND, this is adaptation from Dungeon And Dragon boardgame this project is pure python and you don't need library to run it'.

The screenshot shows a VS Code editor with a Python file named 'door.py' open. The code defines a 'Door' class with methods for initialization and creating a door on a grid. Below the editor, the terminal window shows the output of a 'git push' command, which successfully pushed the changes to the main branch. The terminal output is highlighted with a red box.

```
class Door:
    def __init__(self) -> None:
        self.door_x = random.randint(0, 9)
        self.door_y = random.randint(0, 9)

    def create_door(self, grid):
        grid[self.door_x][self.door_y] = "■"
        return self.door_x, self.door_y

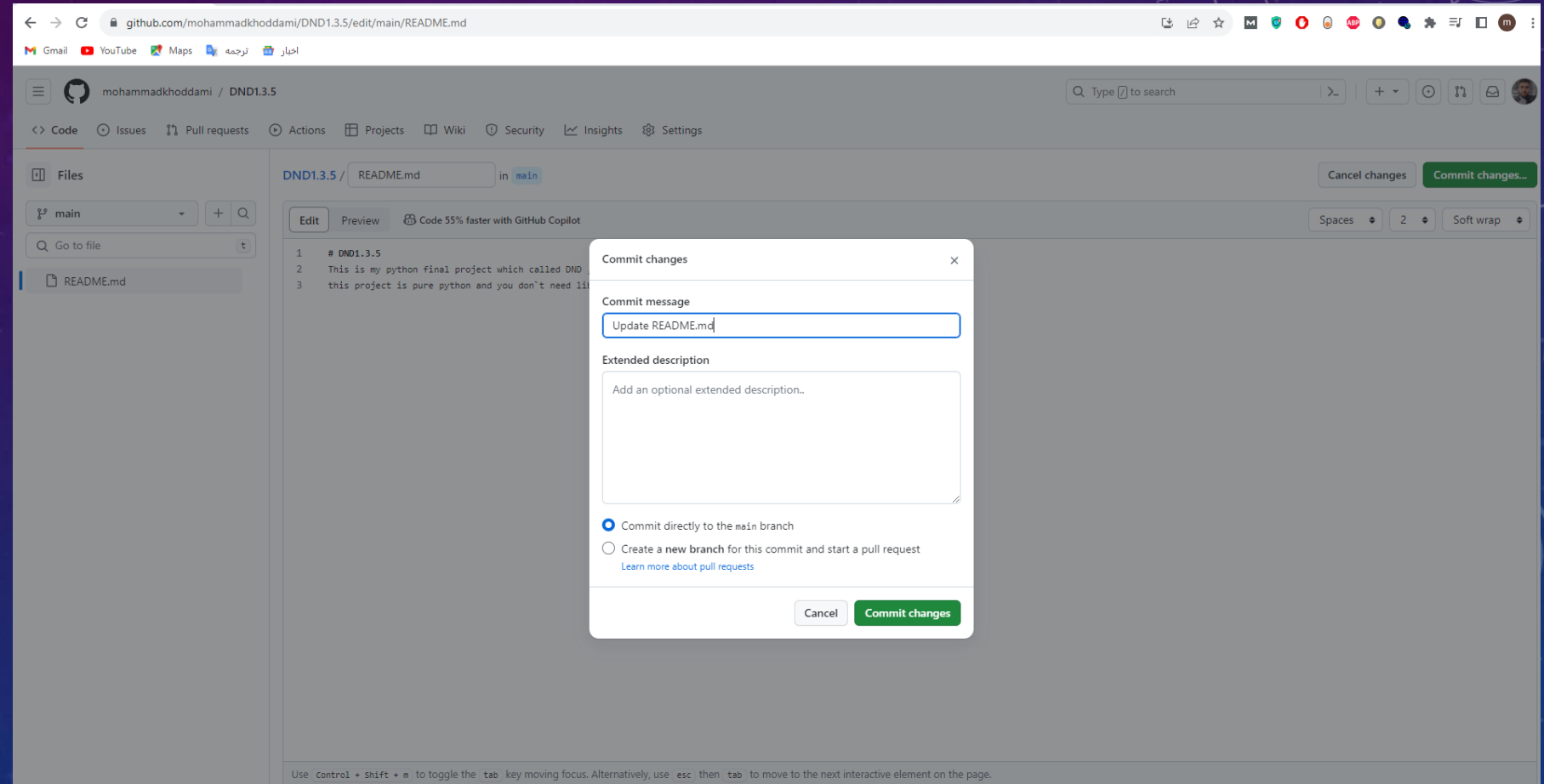
"""
note :
write a funtion for the time that door and dragon being in same position.
should be used after dragon , player , dungeon and before draw map
it's need a file at least!
"""
```

```
Niklaus@DESKTOP-V4GPV4N MINGW64 /c/repos/DND1.3.5 (main)
$ git push
Everything up-to-date

Niklaus@DESKTOP-V4GPV4N MINGW64 /c/repos/DND1.3.5 (main)
$ git push
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 4.21 KiB | 2.10 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/mohammadkhoddami/DND1.3.5.git
   cbf6a58..0e0a85b  main -> main
```

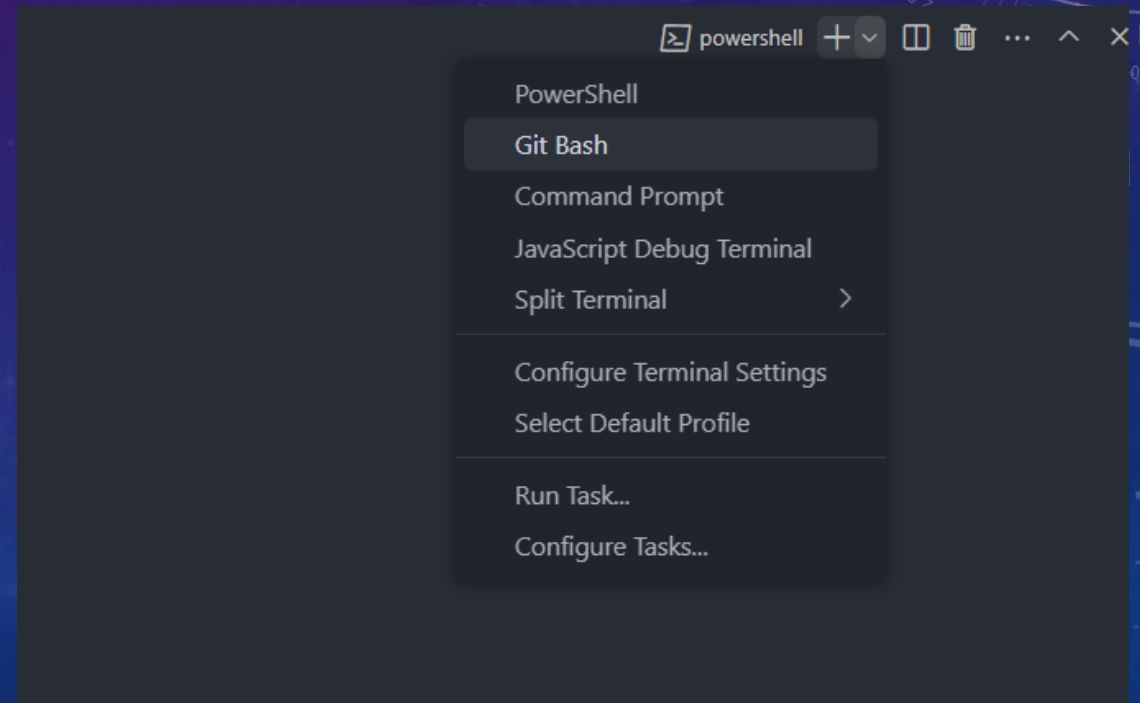

UPDATE FROM WEBSITE

Press commit change...
then name it update
And then commit
changes



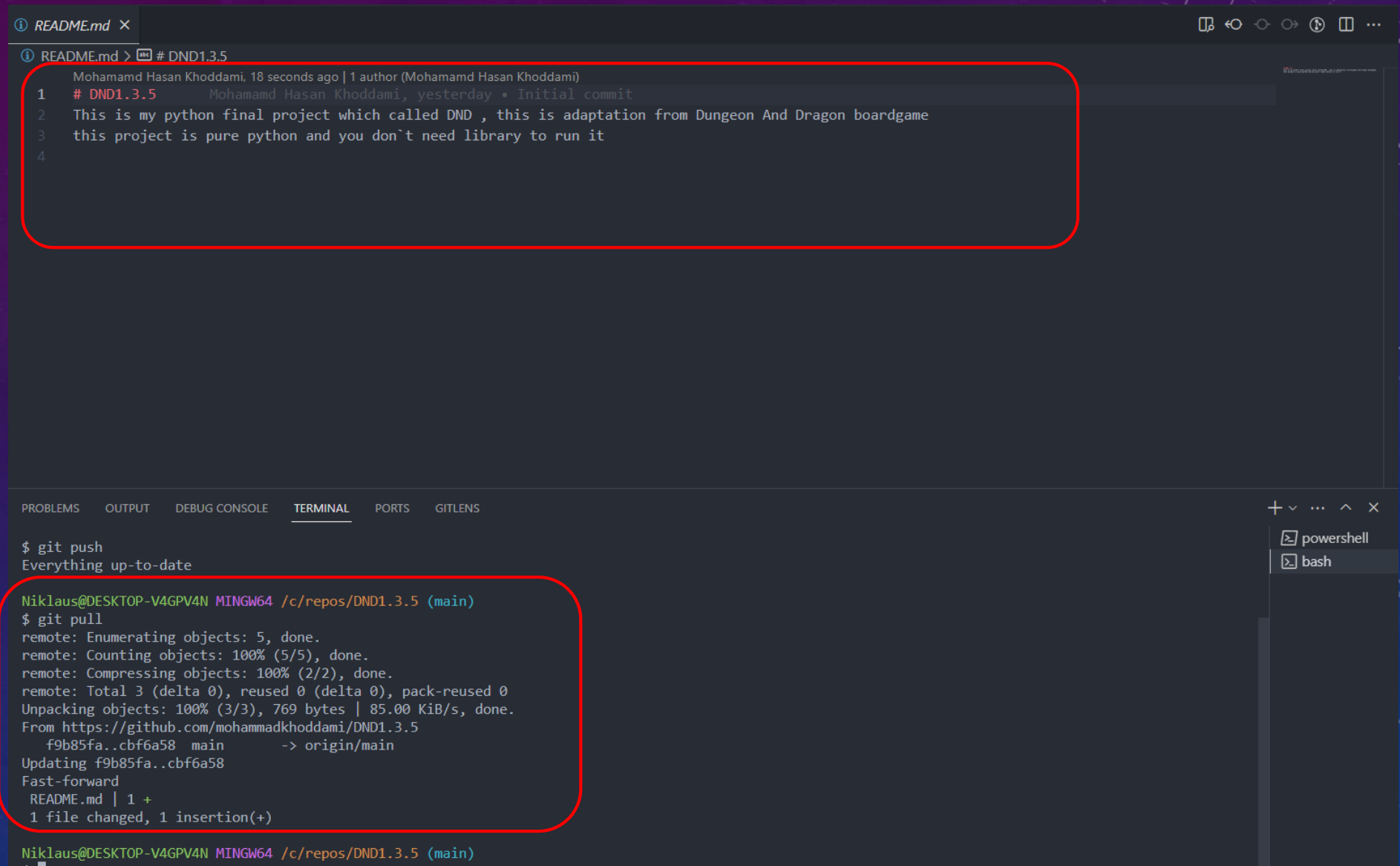
PULL FROM WEBSITE

- When someone (your teammates or even your self) update project from the website we have to pull it in to our local repos to can work on it
 - Open the Git Bash
 - Then code : `git pull` on Git Bash Terminal



RESULT

As you can see
everything is updated



```
README.md X
README.md > # DND1.3.5
Mohamamd Hasan Khoddami, 18 seconds ago | 1 author (Mohamamd Hasan Khoddami)
1 # DND1.3.5 Mohamamd Hasan Khoddami, yesterday • Initial commit
2 This is my python final project which called DND , this is adaptation from Dungeon And Dragon boardgame
3 this project is pure python and you don't need library to run it
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

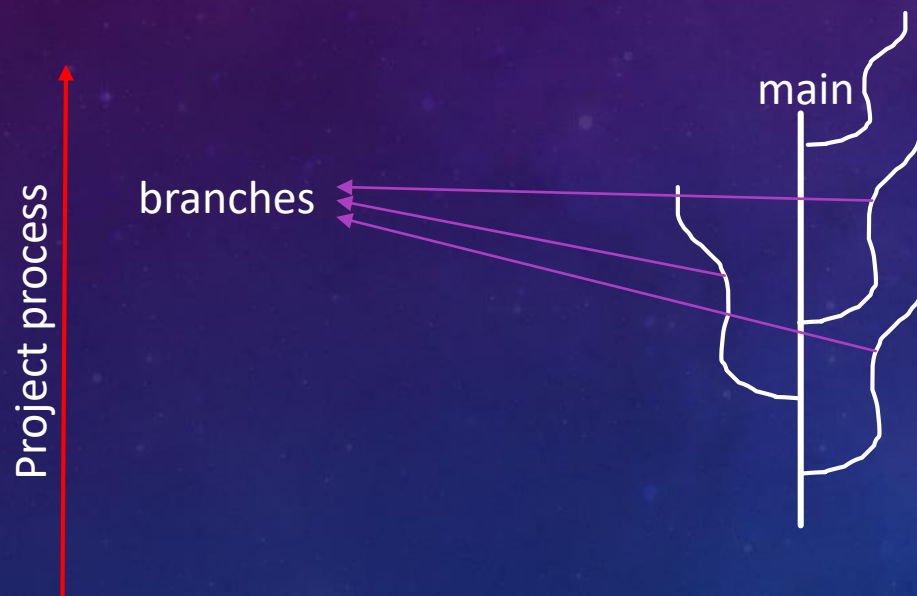
$ git push
Everything up-to-date

Niklaus@DESKTOP-V4GPV4N MINGW64 /c/repos/DND1.3.5 (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 769 bytes | 85.00 KiB/s, done.
From https://github.com/mohammadkhoddami/DND1.3.5
 f9b85fa..cbf6a58  main      -> origin/main
Updating f9b85fa..cbf6a58
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)

Niklaus@DESKTOP-V4GPV4N MINGW64 /c/repos/DND1.3.5 (main)
```

BRANCH

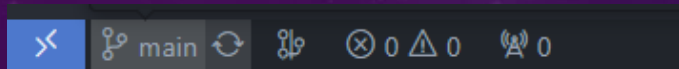
- Usually we don't change main branch! Probably you don't know what is branch so pay attention to this picture



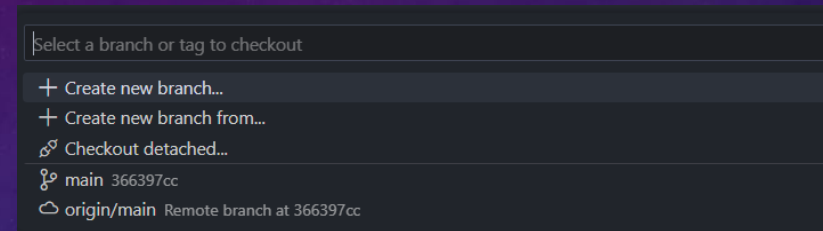
When we change a thing in project or debug it or even add a feature we shouldn't change the main branch of project we have to create a new branch add our codes (change the project) and then push it In new branch and after observer and optimize it by team it could be add to main branch or your branch going to be main branch!

CREATE NEW BRANCH

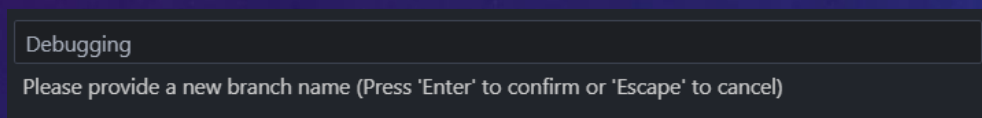
1- on left-bottom click on main



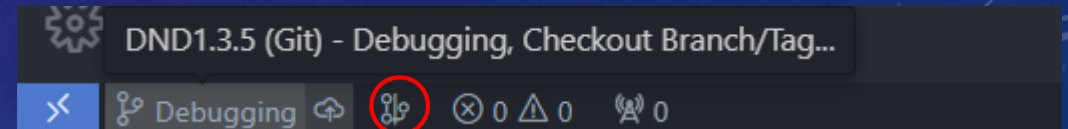
2- on new opened menu click on create new branch



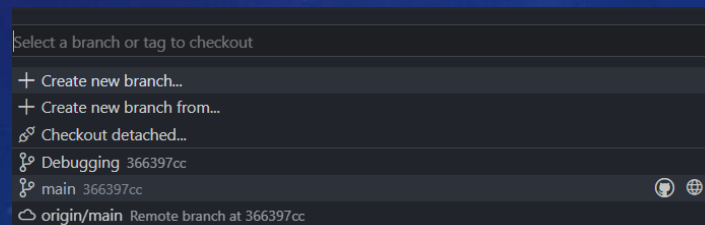
3- chose a appropriate name for your branch



4- and done! New your changes won't change the main branch



4.5- you can chose the branch on this menu too



WRITING README.MD5

- Readme is one of important files of project
- Readme support md5 language and you can write your readme by md5 language and design it by that
- Commands:
 - # → title
 - ### → subtitle
 - ##### → paragraph tittle
 - > → opens a quotation
 - - + space → create set
 - tab + - + space → create subset
 - [site_name] + (site_address) → give the text link , also you can redirect it to one your files
 - [![banner] + (banner_address)] +(site_address) → create a banner and redirect it to you page you want

HOW TO WRITE A GOOD README FILE

- **1. Project's Title**
- **2. Project Description**
- **3. Table of Contents (Optional (if readme file is too long))**
- **4. How to Install and Run the Project**
- **5. How to Use the Project**
- **6. Include Credits**

RESULT

☰ README.md ✎

Dunegon And Dragon

Version : 1.3.5

Explanation

- This is my python final project which called DND , this is adaptation from Dungeon And Dragon boardgame

How to install and run the project

- This project is pure python and you don't need library to run it

How to use project

- this project run in terminal and the only things that you need is a text editor and python.
- first you need to register your informations will save on users.txt and passwords.txt
- then you need to login!

How this game works ?

- you are a player in 10 x 10 map
 - there is a door , key , and dragon
- for opening the door you need key so first of all you need to find key
- if your position and dargon poistion getting equal your going to lose the game
 - if you are near the dragon, dragon will chase you (by Euclidean distance)
 - if your distance from dragon is 2 square by 30% of chance it will chase you
 - if your distance from dragon is 1 square by 90% of chance it will chase
- if your position and door position getting equal and you don't have a key you got a message that you find door but you do not have key
- and in the end if you have key and reach the door you can escape from dungeon and win the Game

Credit

- the whole project is coded by myself
- for contact me you can send me email
- mohammadh.khoddami@gmail.com

```
enemy.py  features.py  game.py  README.md x  README.md (Index)  Login.py  map.py  player.py  ...
D README.md > # Dunegon And Dragon > ## How this game works ?
You, 2 minutes ago | 1 author (You)
1  # Dunegon And Dragon
2  ### Version : 1.3.5
3
4  ### Explanation
5  - This is my python final project which called DND , this is adaptation from Dungeon And Dragon boardgame
6
7  ### How to install and run the project
8  - This project is pure python and you don't need library to run it
9
10 ### How to use project
11 - this project run in terminal and the only things that you need is a text editor and python.
12 - first you need to register your informations will save on users.txt and passwords.txt
13 - then you need to login!
14
15 ### How this game works ?
16 - you are a player in 10 x 10 map
17   - there is a door , key , and dragon
18 - for opening the door you need key so first of all you need to find key
19 - if your position and dargon poistion getting equal your going to lose the game
20   - if you are near the dragon, dragon will chase you (by Euclidean distance)
21     - if your distance from dragon is 2 square by 30% of chance it will chase you
22     - if your distance from dragon is 1 square by 90% of chance it will chase
23 - if your position and door position getting equal and you don't have a key you got a message that you find door but you do not have key
24 - and in the end if you have key and reach the door you can escape from dungeon and win the Game
25
26 ### Credit
27 - the whole project is coded by myself
28 - for contact me you can send me email
29 - mohammadh.khoddami@gmail.com
30
31
```

Just don't forget to push the changes!

EXTENSION

- It's better to download GIT Lens extension on VSCode it can help you!



THANKS FOR YOUR ATTENTION

MOHAMMADH.KHODDAMI@GMAIL.COM