# МІНІСТЕРСВО ОСВІТИ І НАУКИ УКРАЇНИ

## НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Звіт з лабораторної роботи №1,2,3

З курсу “**ТЕОРІЯ ПРОЕКТУВАННЯ КОМП’ЮТЕРНИХ СИСТЕМ**

”

На тему: " Initiate GIT repository”

Виконав: ст. групи КІ-31з

Дмитришин Д. О.

Перевірив: викладач

Шпіцер А.С

1. Create a github repo with the name “**csdt*<YY1YY2><group><student's full name><student's number>***”, main branch develop.

*YY1* – Start studying year *YY2* – End studding year

For example: 2019-2020 -> **1920**Full example **csdt1920ki47spitzeras03**

1. Add access for the author.
2. Create a branch **feature/develop/<task number>**. For example

**feature/develop/task1.**

1. Create **README** file with:
   1. detailsaboutrepo;
   2. task details;
   3. studentnumberanddetailsfrom**Table1.Tasksbystudents**;
   4. add details about technology, program language, and HW that will be

used in next tasks.

1. Create **GIT TAG**: *<PROJECT NAME>\_<VERSION>\_WW<YYWWD>* 
   1. ∙  YY – current year;
   2. ∙  WW – work week;
   3. ∙  D – current day number of weak.
2. Create pull request with name **task1** and submit **lecturer** as reviewer.
3. After the reviewer approved – merge request into **develop** branch.

A screenshot of a computer

Description automatically generated

dmitro@MacBook-Air-Dmitro TikTac % echo "# Python game TikTactoe v1" >> README.md

dmitro@MacBook-Air-Dmitro TikTac % git comit

git: 'comit' is not a git command. See 'git --help'.

The most similar command is

commit

dmitro@MacBook-Air-Dmitro TikTac % git commit -m "second update"

On branch main

Your branch is up to date with 'origin/main'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: README.md

Untracked files:

(use "git add <file>..." to include in what will be committed)

.DS\_Store

.idea/

tiktactoe.py

no changes added to commit (use "git add" and/or "git commit -a")

dmitro@MacBook-Air-Dmitro TikTac % git add README.md

dmitro@MacBook-Air-Dmitro TikTac % git commit -m "second update"

[main 6a45b83] second update

1 file changed, 1 insertion(+)

dmitro@MacBook-Air-Dmitro TikTac % git add tiktactoe.py

dmitro@MacBook-Air-Dmitro TikTac % git commit -m "Game v1"

[main 6c9d147] Game v1

1 file changed, 53 insertions(+)

create mode 100644 tiktactoe.py

dmitro@MacBook-Air-Dmitro TikTac % git help

usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

[--super-prefix=<path>] [--config-env=<name>=<envvar>]

<command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

dmitro@MacBook-Air-Dmitro TikTac % git branch developers

dmitro@MacBook-Air-Dmitro TikTac % git push origen main

fatal: 'origen' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

dmitro@MacBook-Air-Dmitro TikTac % git push origin main

Enumerating objects: 8, done.

Counting objects: 100% (8/8), done.

Delta compression using up to 8 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (6/6), 1.11 KiB | 1.11 MiB/s, done.

Total 6 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/Dmytr9o/csdt2124ki41zdmytryshyndo05.git

ae0b4b1..6c9d147 main -> main

TASK 2:

1. Develop SW game.

2. Required steps.

def print\_board(board):

for row in board:

print(" | ".join(row))

print("-" \* 9)

def check\_winner(board, player):

for row in board:

if all(cell == player for cell in row):

return True

for col in range(3):

if all(row[col] == player for row in board):

return True

if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i] == player for i in range(3)):

return True

return False

def main():

board = [[" " for \_ in range(3)] for \_ in range(3)]

current\_player = "X"

moves = 0

print("Гра 'Хрестики-нолики'!")

print\_board(board)

while True:

row = int(input(f"Гравець {current\_player}, введіть номер рядка (0, 1, або 2): "))

col = int(input(f"Гравець {current\_player}, введіть номер стовпця (0, 1, або 2): "))

if row < 0 or row > 2 or col < 0 or col > 2 or board[row][col] != " ":

print("Недійсний хід. Спробуйте ще раз.")

continue

board[row][col] = current\_player

moves += 1

print\_board(board)

if check\_winner(board, current\_player):

print(f"Гравець {current\_player} виграв!")

break

elif moves == 9:

print("Нічия!")

break

current\_player = "O" if current\_player == "X" else "X"

if \_\_name\_\_ == "\_\_main\_\_":

main()