

## Proiect 1

### I. Cu ce vom lucra?

1. Docker desktop
2. Visual Studio Code
3. GitHub

### II. Cum instalam soft-urile necesare?

- Docker desktop (vezi Laborator 1-Baze de date anul universitar 2024-2025)  
Link: <https://www.docker.com/products/docker-desktop/>
- Visual Studio Code: Link: <https://code.visualstudio.com/download>
- Git: Link: <https://git-scm.com/downloads>
- GitHub: Link: <https://github.com/>

### III. Cum vom folosi GIT în proiectele noastre?

1. Crearea contului pe GitHub
2. Crearea unui Repository

Un repository (sau „repo”) pe GitHub este un spațiu de stocare unde se păstrează toate fișierele și istoricul unui proiect. Practic este „dosarul” proiectului tău online: conține codul sursă, documentația, dar și informații despre toate modificările făcute de-a lungul timpului.

Un repository poate fi public (vizibil pentru oricine) sau privat (vizibil doar pentru tine și colaboratorii tăi). Prin repo-uri, GitHub permite organizarea proiectelor, colaborarea între mai mulți programatori și partajarea codului cu comunitatea.

Repo-ul pe care voi îl creați poate fi privat sau public dar trebuie să aibă ca denumire **numele vostru!** (acces pentru cele 3 cadre didactice ale acestei materii)

Inițializare și încărcare	
git init	Inițializezi un repo local
git clone	Copiază conținutul de pe GitHub local în PC
Git add [numefisierului.extensia]	Adaugi fișiere în staging
Git commit -m ”denumirecommit”	Creezi un commit

git config --global user.email "danielacitu4@gmail.com"	Este folosit pentru auth
git remote add origin https://github.com/danielatunsoiu/CursTW.git	Setezi conexiunea cu GitHub
git push -u origin main	Trimiți codul online (pe GitHub)
<b>Branch-uri din CMD(Command Prompt)</b>	
git checkout -b feature-login	Creezi un branch nou
git switch -c feature-login	Creezi un branch nou
git checkout main	Te întorci pe main
git merge feature-login	Combină branch-ul cu main
git branch -d feature-login	Ștergi un branch local
<b>Conectare la BRANCH-uri create in GitHub</b>	
git remote -v	Verifică remote-ul
git fetch --all --prune	Aduce toate branch-urile remote local
git branch -a	Listează branch-urile
git switch -c test --track origin/test	Creează local un branch care urmărește branch-ul remote

**Pașii** pentru a retrage un proiect din Github în calculator:

1. Se creează proiectul (să conțină fișierul README)
2. Se creează un branch (un fel de copie de siguranță) cu numele L1, L2, L3, P1, P2, P3,...
3. Se verifică în CMD Git-ul (comanda *git*)
4. Se creează un folder în adresa din CMD (ex- C:/Users/..)
5. *cd numelefolderului* - pentru a intra în folder-ul creat
6. *git clone linkproiectdinGithub*
7. După ce proiectul a apărut în folder și se adaugă alte doc sau se fac modificările necesare, trebuie să trimitem doc înapoi în Github  
*git add numedoc.extensia* (ex:lab1.html)
8. Urmează commit  
*git commit -m "numele commitului"*
9. Se va cere să ne autentificăm și vom folosi e-mailul folosit la crearea contului în Github  
*git config --global user.email "adresa@gmail.com"*

10. Încercăm din nou realizarea commit-ului

***git commit -m "numele commitului"***

11. Realizăm comanda de push

***git push***

11. Regăsim în github documentele actualizate/adăugate

Realizăm îmbinarea documentelor cu Merge dacă suntem siguri că varianta pe care vrem să o adăugăm este corectă.

Cum facem conexiunea unui document php prin docker:

1. Se creează document text cu denumirea **docker-compose.yml** cu următorul cod:

```
version: '3.8'
services:
  apache:
    image: php:8.1-apache
    container_name: lamp_apache
    volumes:
      - ./studenti:/var/www/html
    ports:
      - "8080:80"
    depends_on:
      - mysql
  mysql:
    image: mysql:5.7
    container_name: lamp_mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: studenti
      MYSQL_USER: user
      MYSQL_PASSWORD: password
    ports:
      - "3307:3307"
    volumes:
      - db_data:/var/lib/mysql
volumes:
  db_data:
```

2. Se creeaza document text cu denumirea **index.php** cu următorul cod:

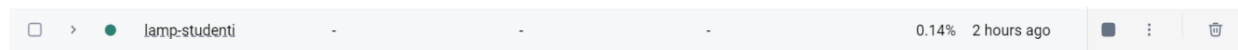
```
<?php echo "Prima ora de proiect"; ?>
```

3. Se creează două foldere - unul **lamp-studenti** care să aibă în interior documentul cu extensia yml de mai sus și al doilea folder în interior numit **student** care să aibă în el documentele text-php, html, etc.

4. În docker:

Se ruleaza ***mkdir lamp-studenti***

Se ruleaza ***docker compose up -d*** - aici se asteapta cateva secunde/minute



Va fi creat container-ul cu numele **lamp-studenti**

5. Se va accesa <http://localhost:8080>

6. Pentru a accesa proiectul se deschide visual studio code - open folder și se va selecta folderul lamp-studenti.