

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



Факультет інформаційних технологій  
**Кафедра системного аналізу та управління**

**Звіт**  
з практичної роботи №2 з дисципліни  
**«Аналіз програмного забезпечення»**

Виконала:  
студентка групи 122-22-2

Калмикова О. К.

Перевірили:  
доц. Мінеєв О.С.  
ас. Шевченко Ю.О.

**Дніпро**  
**2025**

**Тема:** Створення і налаштування профілю у системі Git.

**Мета:** Набування навичок при реєстрації та налаштуванню облікового запису (account) на хостінгу GitHub.

**Очікувані результати навчання:** уміння створити, налаштувати та підтримувати власний профіль на найбільшому хостингу для сховищ GitHub.

**Хід роботи:**  
Першочергово необхідно перейти на сайт на сайт <https://github.com/> (рис 1.1) та розпочати реєстрацію.

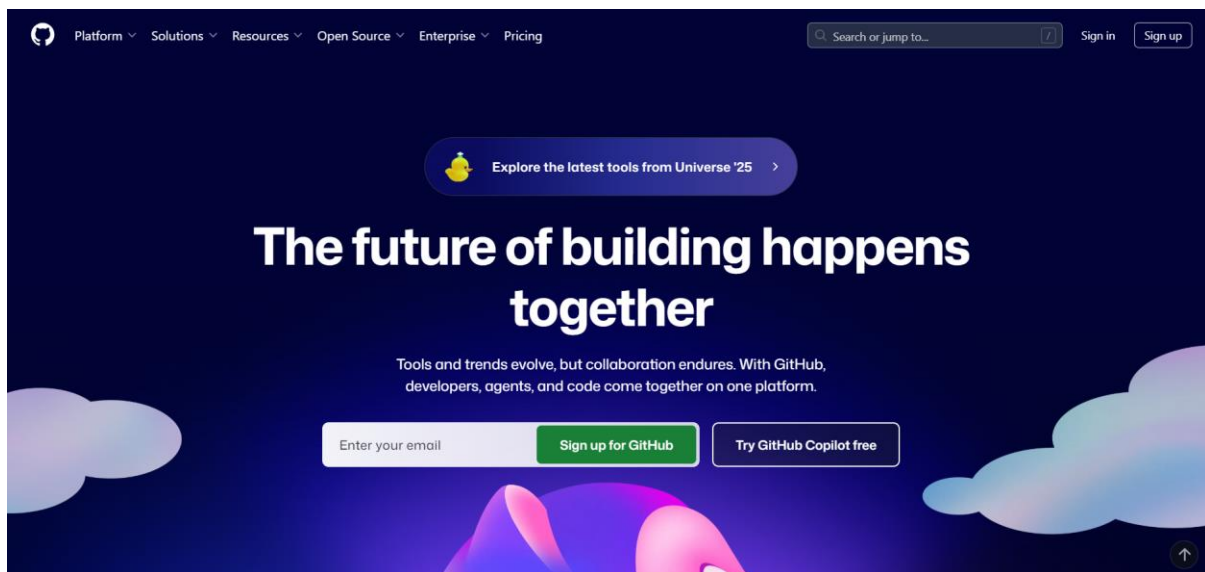


Рис 1.1 перехід на сайт

Для реєстрації було введено всі необхідні данні після чого акаунт був успішно створений.

The image shows the GitHub account creation interface. At the top, there are two buttons: "Continue with Google" and "Continue with Apple". Below these is a horizontal line with the word "or" in the center. The form consists of several fields: "Email" with the value "Kalmykova.O.K@nmu.one", "Password" with masked characters, "Username" with the value "S-Sasha-122", and "Your Country/Region" with a dropdown menu set to "Ukraine". Below the country field is a checkbox for "Email preferences" which is checked, with the text "Receive occasional product updates and announcements". At the bottom of the form is a large dark button labeled "Create account >". Below the button, there is a line of text: "By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll

Рис 1.2 реєстрація

Далі необхідно створити нового проєкт (репозиторій) для розміщення всіх практичних робіт в ньому.

*Репозиторій* — це місце може містити файли коду, документацію, зображення і будь-які інші файли, які стосуються проєкту.

Для початку треба увійти в новостворений акаунт для початку роботи(рис 1.3)

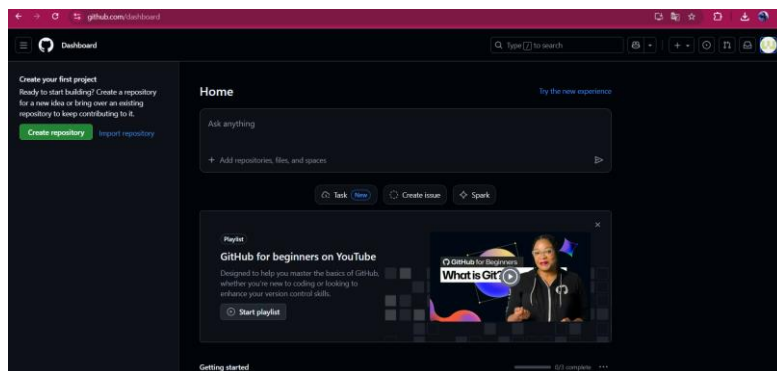


Рис 1.3 вхід до GetHub

Натиснувши “Create repository” переходимо до створення

публічного репозиторію і всі файли в ньому зможе побачити будь-яка людина в інтернеті.

Рис 1.3 створення репозиторію

Спробуємо створити тестовий файл “lr1” у репозиторії через кнопку “create new file” (рис 1.4)

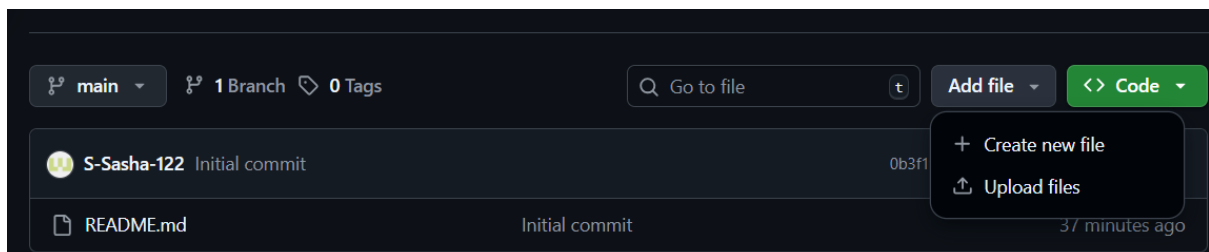


Рис 1.4



У репозиторії Study-122-22-2, розташовано папку arz, а у ній тестовий файл lr1, та звіт у PDF з першого практичного заняття.

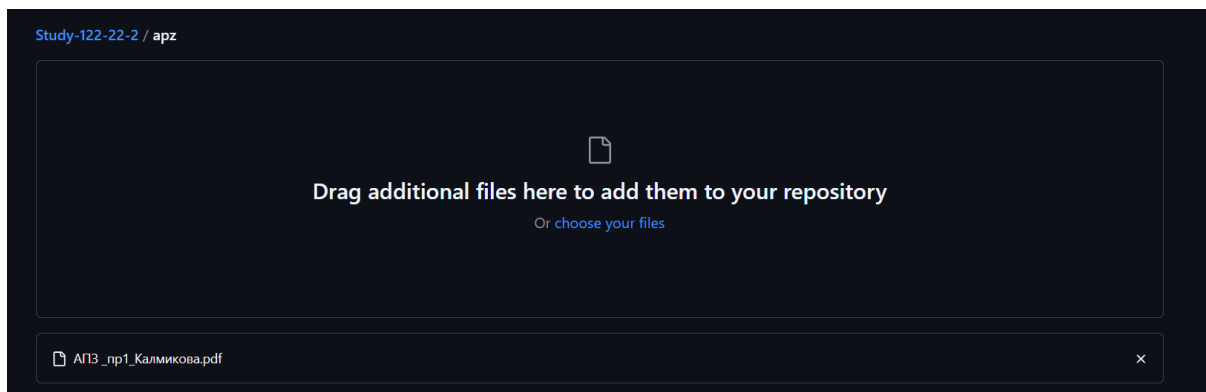


Рис 1.5 додавання PDF файлу

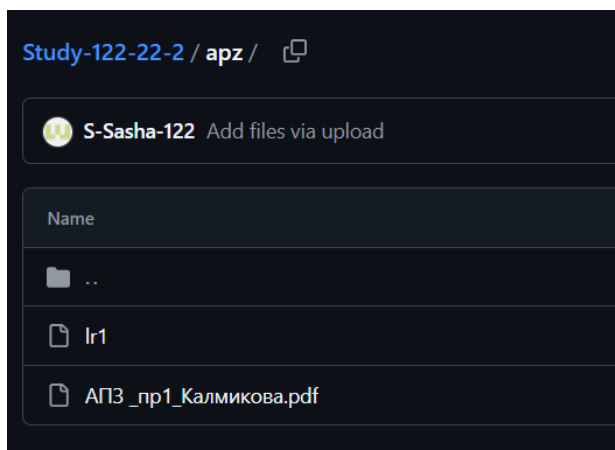


Рис 1.6 кінцевий результат

**Висновок:** у ході виконання роботи було опановано основні кроки зі створення та налаштування облікового запису на GitHub — найбільшому хостингу для зберігання та спільної розробки проєктів, що використовують Git. Було розглянуто процес реєстрації, базових налаштувань профілю, створення репозиторію, файлів та папок.

### Контрольні питання:

#### 1. Що таке GIT?

Git — це система контролю версій, яка дозволяє відстежувати зміни в проєкті та працювати над ним у команді.

#### 2. Що таке репозиторій у GIT?

Репозиторій — це місце, де зберігаються файли проєкту та історія їх змін.

### **3. Які переваги використання GIT?**

Швидкість, робота офлайн, зручні гілки, командна робота, збереження історії, захист від втрати даних.

### **4. Яка мова використовується в GIT?**

Git використовує власні команди в терміналі (наприклад: `git init`, `git add`, `git commit`).

### **5. Як можна створити репозиторій у Git?**

Створити у теці новий — команда: `git init`.

Або скопіювати готовий — команда: `git clone <посилання>`.

### **6. Яка команда використовується для видалення гілки?**

Для видалення: `git branch -d <назва>`.

Примусове видалення: `git branch -D <назва>`.

### **7. Що таке контроль версій GIT?**

Це спосіб керування змінами у файлах з можливістю повернення до попередніх версій.

### **8. Як можна виправити несправний комміт?**

Оновити останній комміт: `git commit --amend`.

Скасувати комміт: `git revert <хеш>`.

Повністю відкотити: `git reset --hard <хеш>`.

### **9. Як дізнатись, чи гілку вже об'єднано з master?**

Подивитись злиті гілки: `git branch --merged master`.