**Практичне завдання**

**Завдання №1. Початок роботи з Git. Файл README.md, .gitignore. Commit. Push. GitHub.**

1. Зенеруйте SSH-ключ та налаштуйте доступ до вашого акаунту за допомогою згенерованого SSH-ключа.

Для того, щоб згенерувати SSH ключ необхідно:

* 1. Ввійти у свій акаунт GitHub.
  2. Нажати на свою іконку в правому верхньому куті та вибрати **Settings**

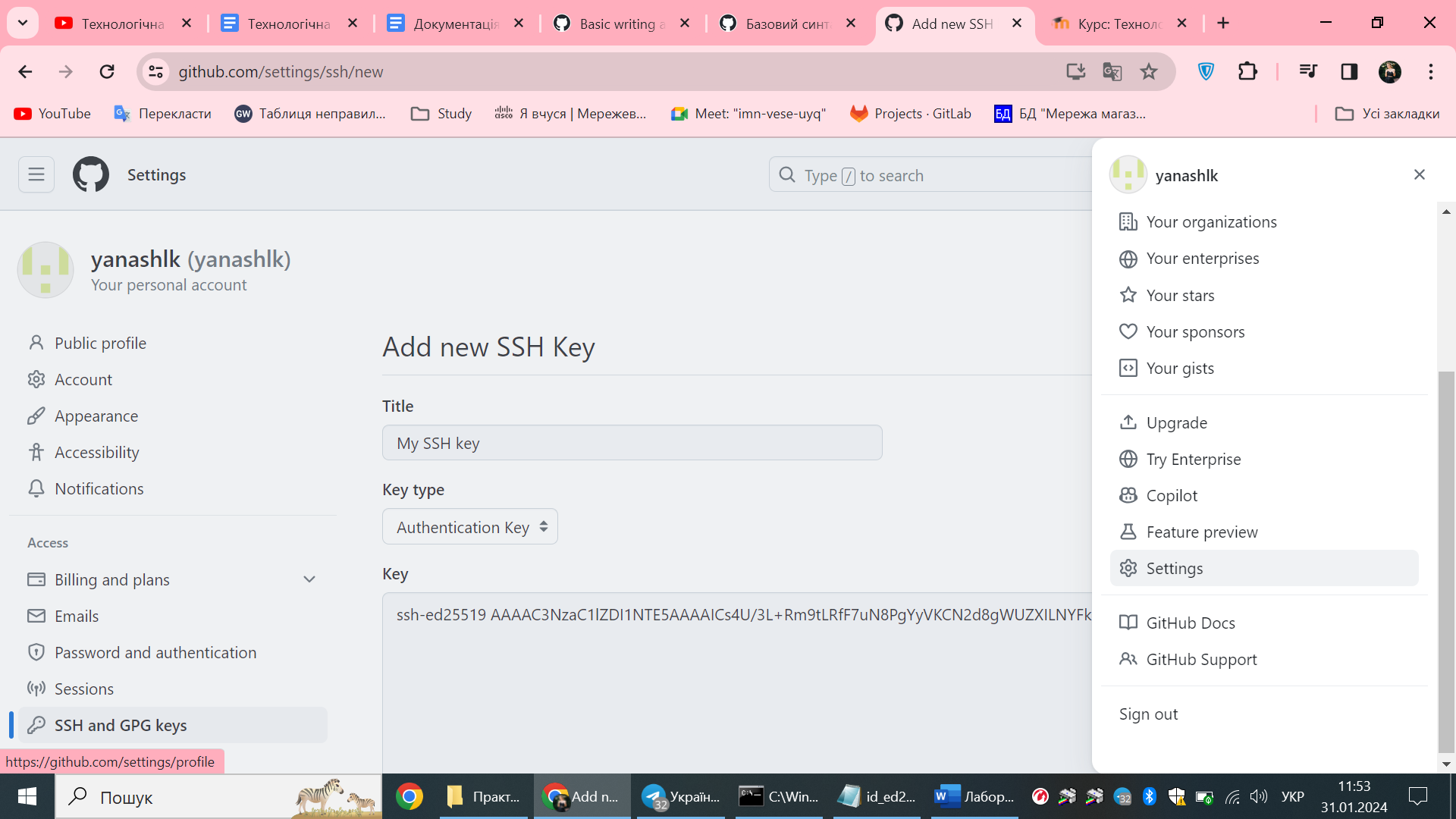


Рис. 1.1.

* 1. Далі переходимо до пункту **SSH and GPG keys** та натискаємо на кнопку **New SSH key**

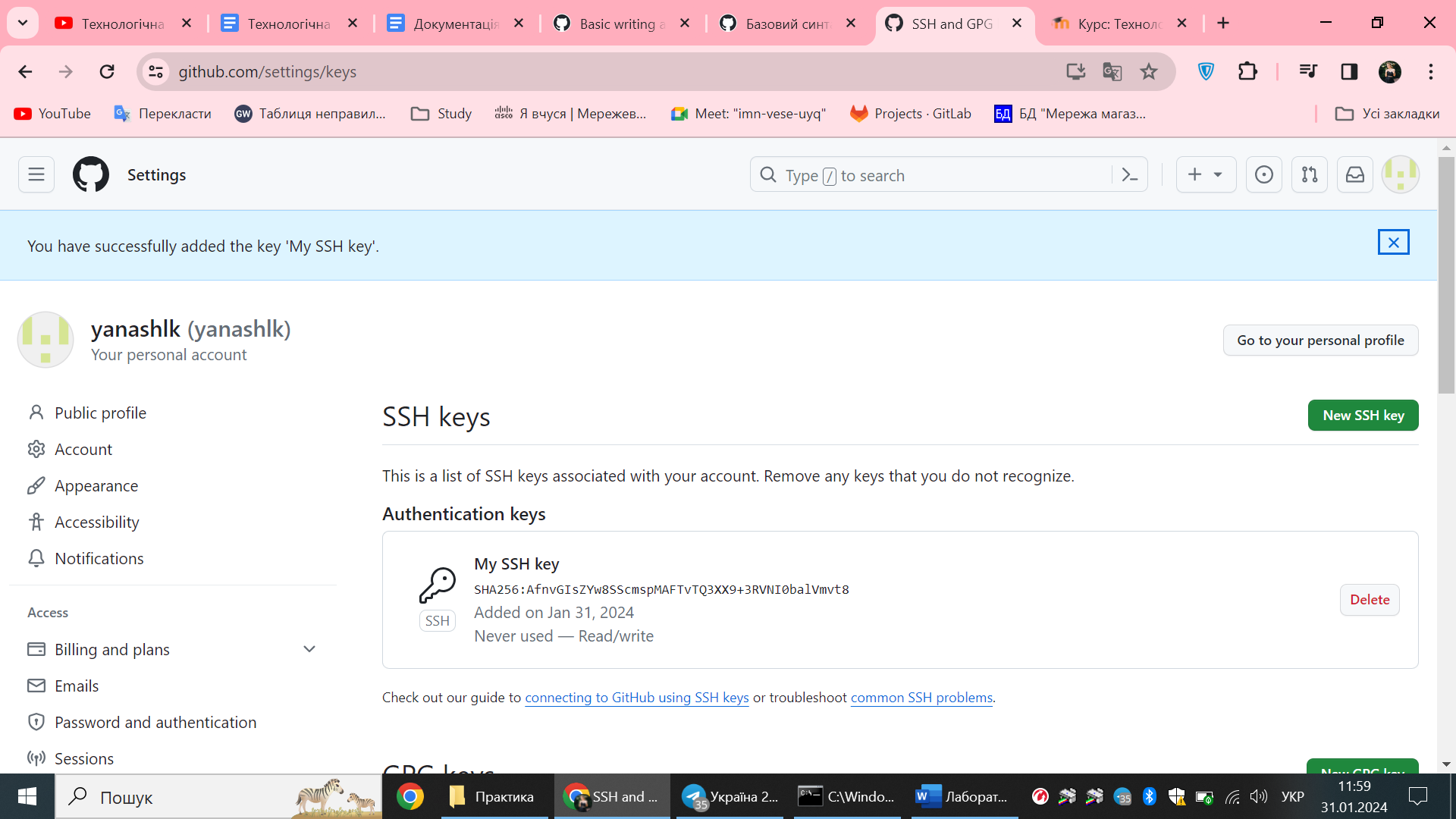


Рис.1.2

* 1. Наступним кроком відкриваємо термінал та вводимо відповідні команти:

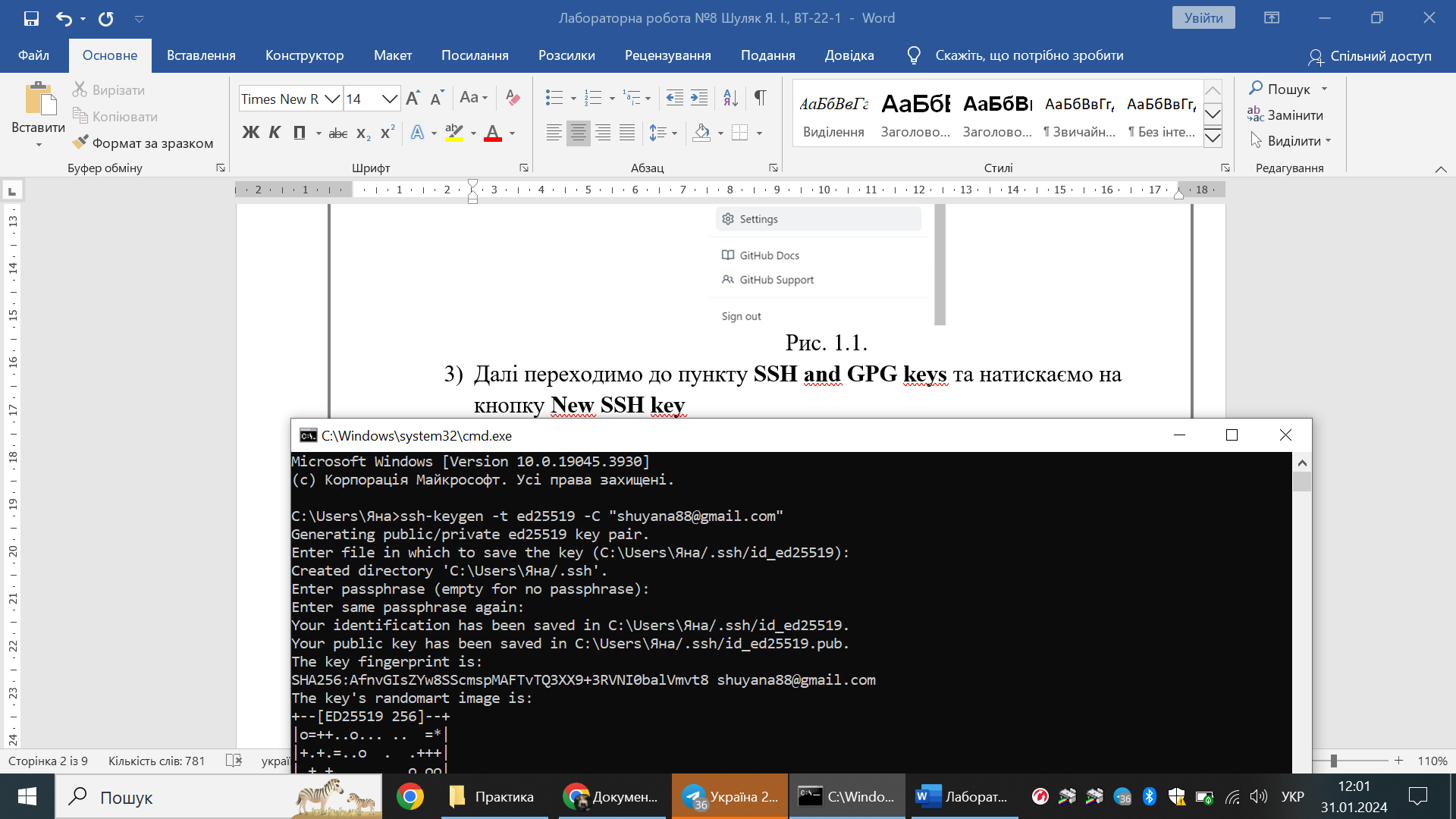


Рис.1.3. Виконуємо генераці. SSH ключа для свого акаунта

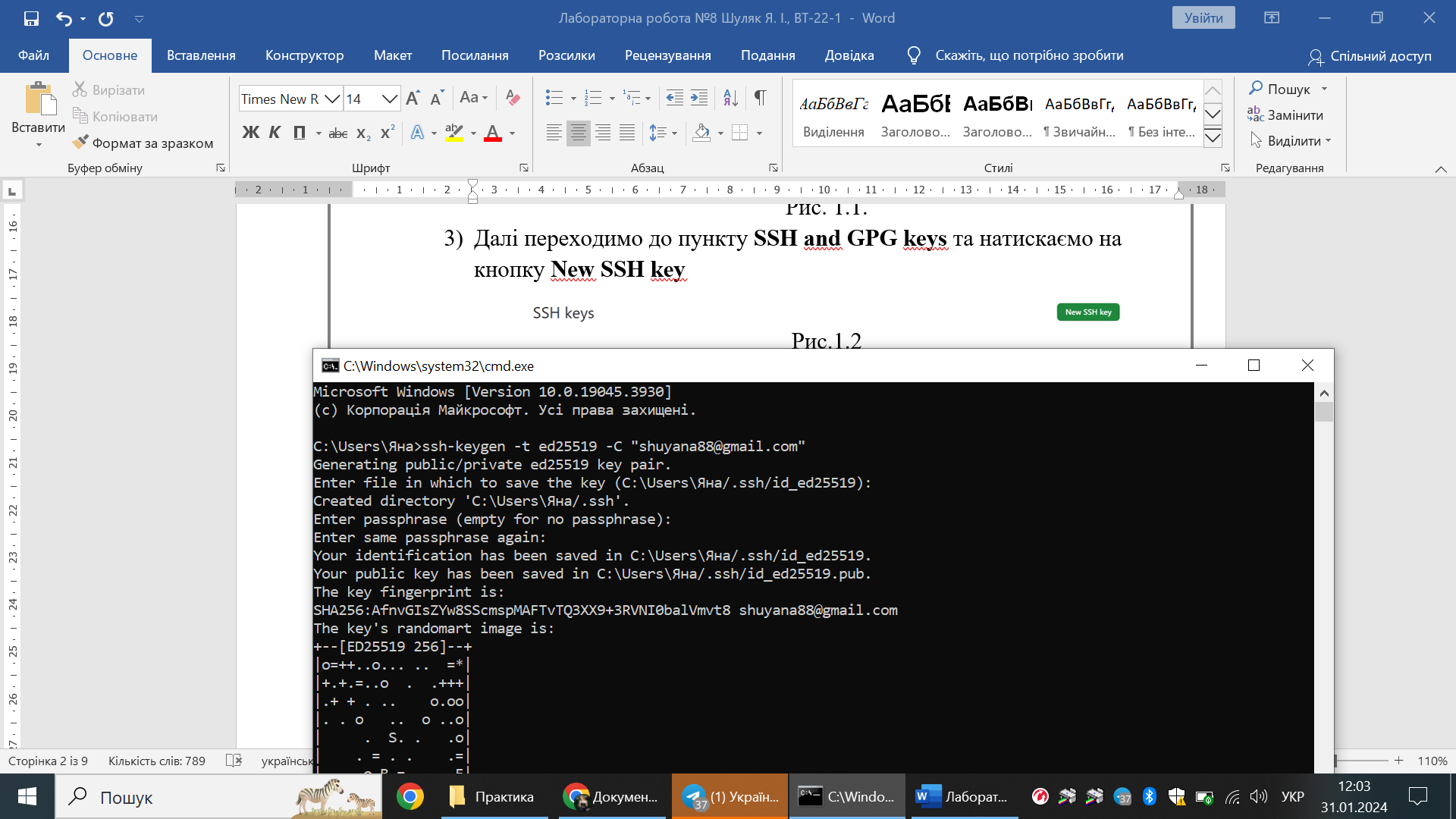


Рис.1.4. Тут нам запропонувало вибрати місце, куди буде зберігатись ключ (погодились)

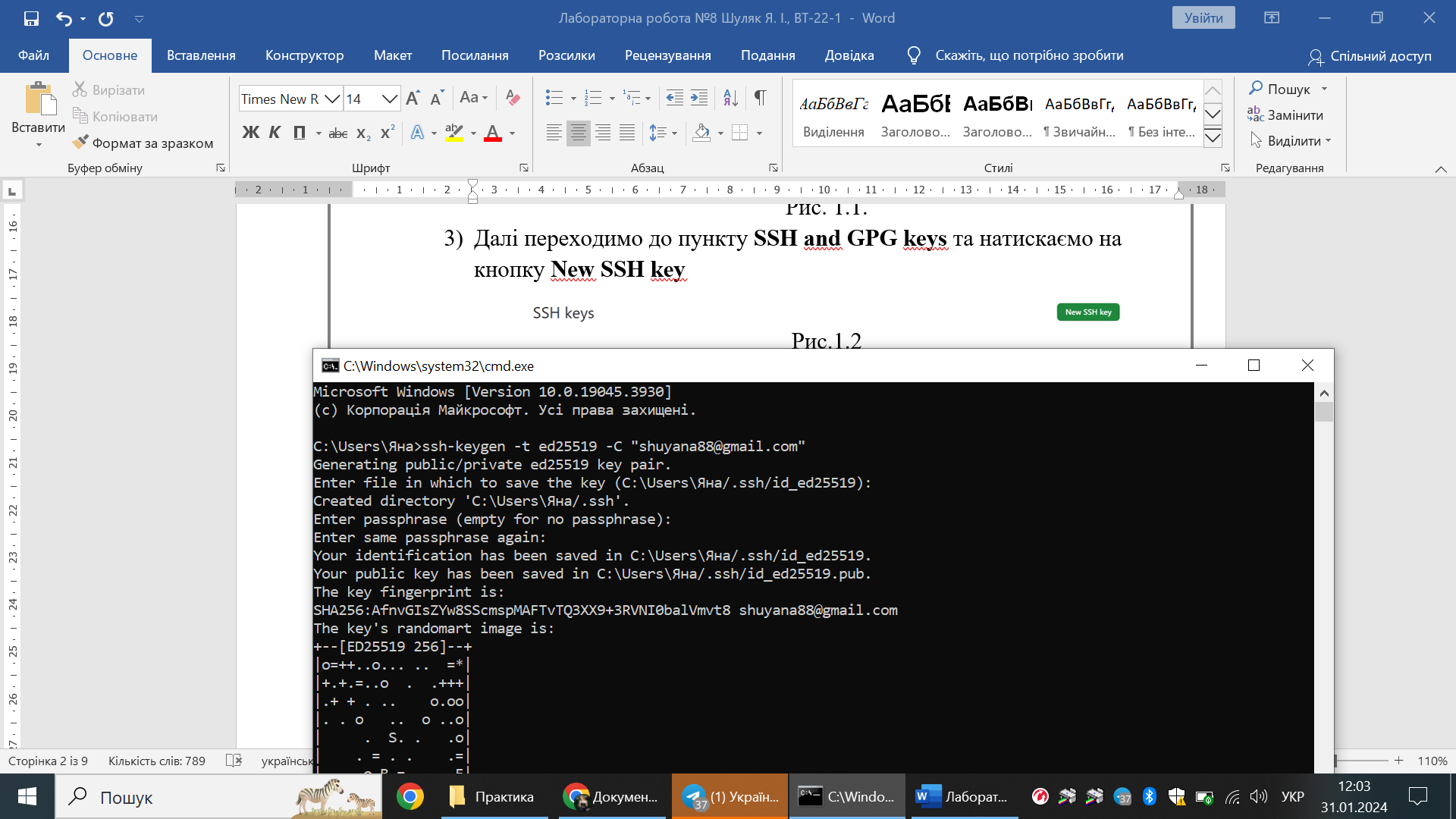


Рис.1.5. Наступним кроком потрібно придумати та ввести пароль (він не відображається задля безпеки)

* 1. Далі знаходимо за тим посиланням, яке нам пропонувалось вище та шукаємо папку .ssh, в якій буде знаходитись файл id\_ed25519.pub і саме його вміс потрібно вставити в поле **Key** на GitHub

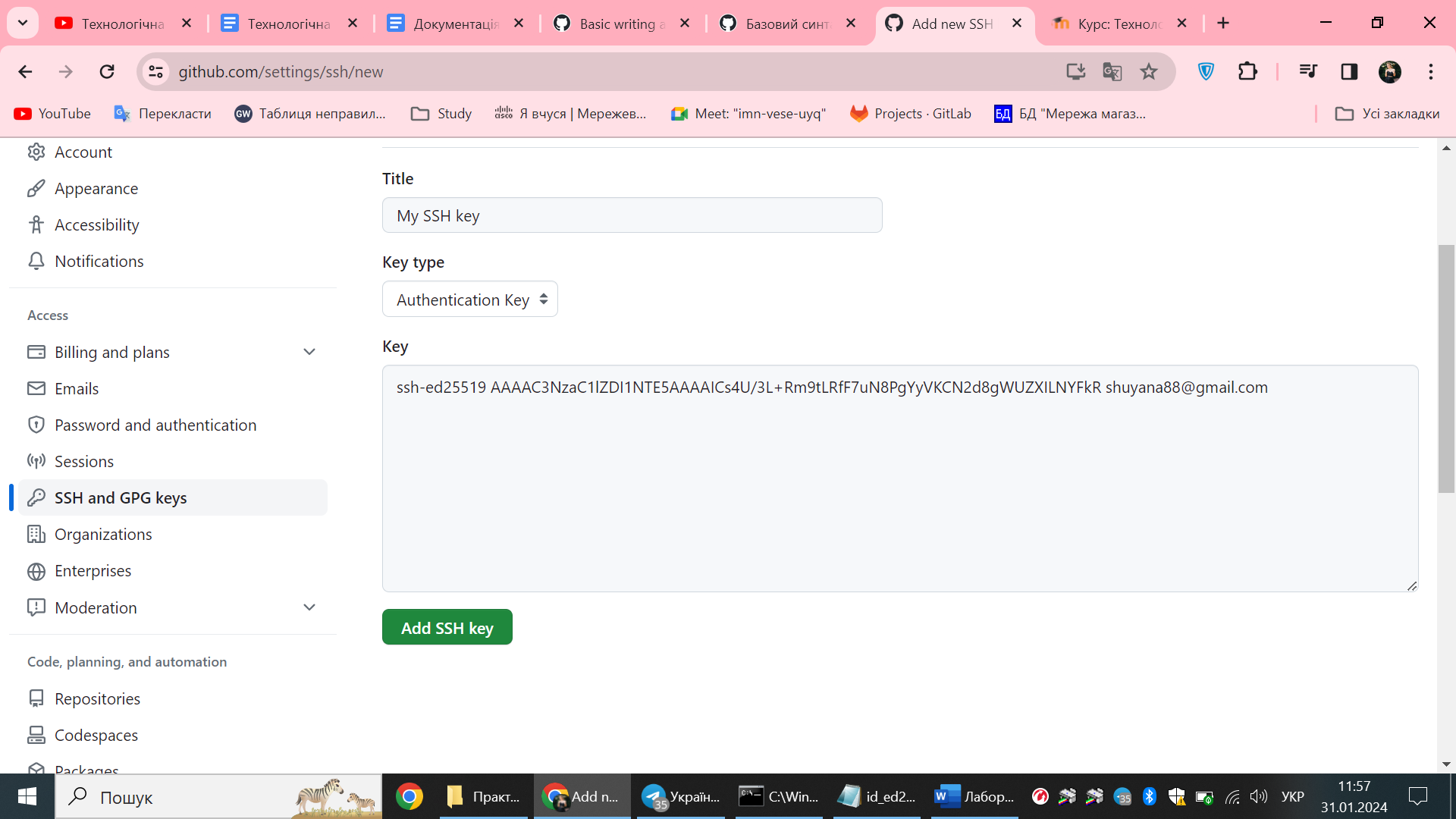


Рис. 1.6

Придумуємо Title та натискаємо Add SSH key

Після усіх цих маніпуляцій ми успішно створили SSH ключ.

1. Створіть файл **README.md** та внесіть у нього інформацію.

Файл можна створити власноруч, наприклад, в блокноті та зберегти з розширенням **.md** або ж створити його відразу при створенні репозиторію в самому GitHub

Я використала другий варіант і ось результат:

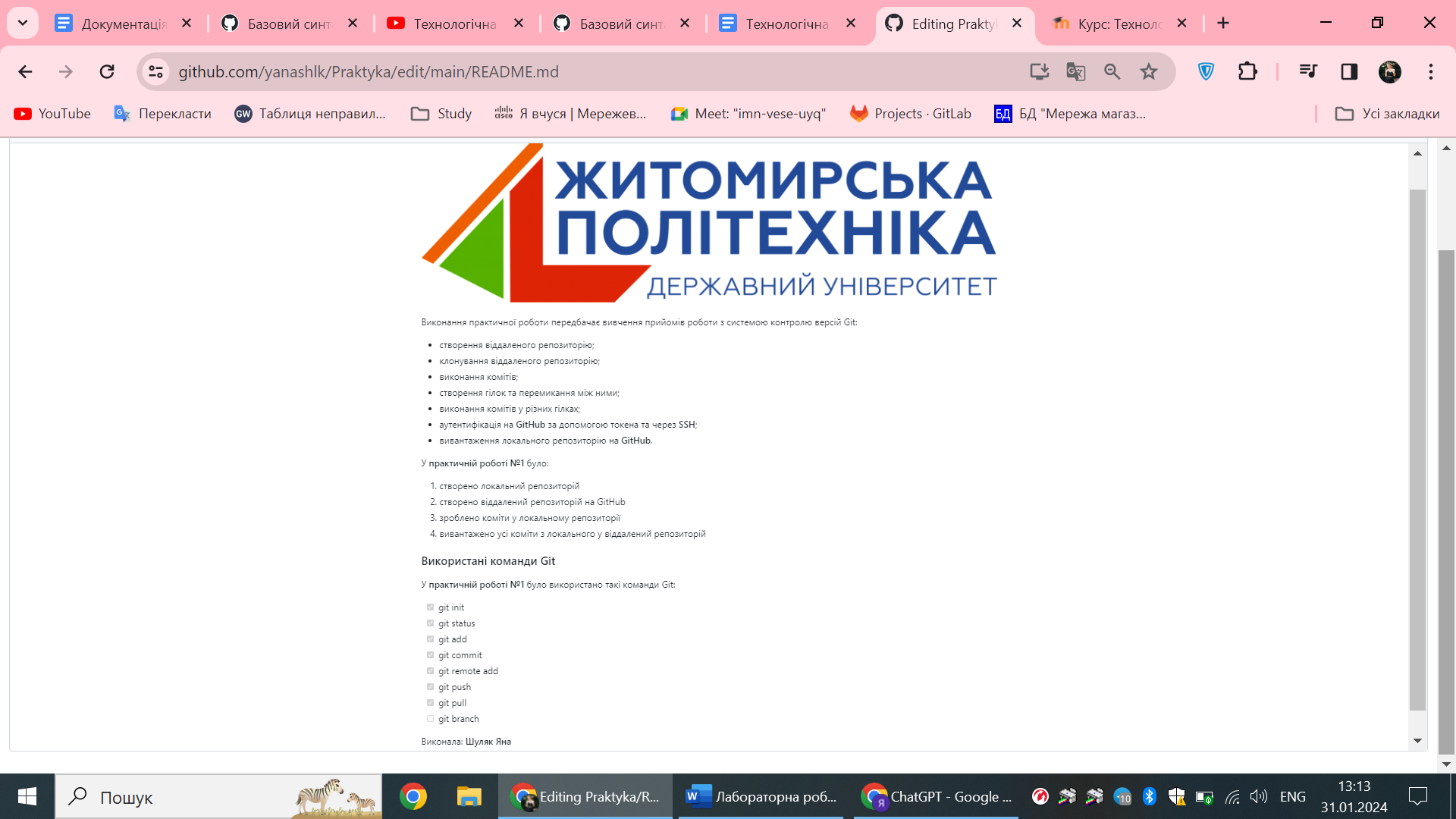


Рис.1.7

Текст, який необхідно ввести, щоб вийшов такий результат:

|  |
| --- |
| ![Зображення ЖДТУ](https://media.ztu.edu.ua/wp-content/uploads/2020/02/Group-6-1-1536x465.png)  Виконання практичної роботи передбачає вивчення прийомів роботи з системою контролю версій Git:  - створення віддаленого репозиторію;  - клонування віддаленого репозиторію;  - виконання комітів;  - створення гілок та перемикання між ними;  - виконання комітів у різних гілках;  - аутентифікація на \*\*GitHub\*\* за допомогою токена та через \*\*SSH\*\*;  - вивантаження локального репозиторію на \*\*GitHub\*\*.  У \*\*практичній роботі №1\*\* було:  1. створено локальний репозиторій  1. створено віддалений репозиторій на GitHub  1. зроблено коміти у локальному репозиторії  1. вивантажено усі коміти з локального у віддалений репозиторій  ### \*\*Використані команди Git\*\*  У \*\*практичній роботі №1\*\* було використано такі команди Git:  - [x] git init  - [x] git status  - [x] git add  - [x] git commit  - [x] git remote add  - [x] git push  - [x] git pull  - [ ] git branch  Виконала: \*\*Шуляк Яна\*\* |

Пояснення символів:

1. Щоб додати зображення потрібно ввести -> ! (альтернативний текст) [посилання на зображення]
2. Щоб створити маркований список, необхідно перед кожним розділом маркованого списку поставити – або +, або \*.
3. Щоб зробити текст жирним, потрібно огорнути текст, який необхідно зробити жирним в \*\*
4. Щоб створити послідовний список, то варто перед кожним розділом поставити 1. , а далі воно продовжиться автоматично.
5. Щоб застосувати рівні, то перед абзацом необхідно поставити #, ## або ### знаки, відповідно якого рівня потрібно зробити текст.
6. Щоб створити список завдань, перед елементами списку ставиться дефіс і пробіл, а потім [ ]. Щоб позначити завдання як виконане, треба написати [x].
7. Зробіть коміт.

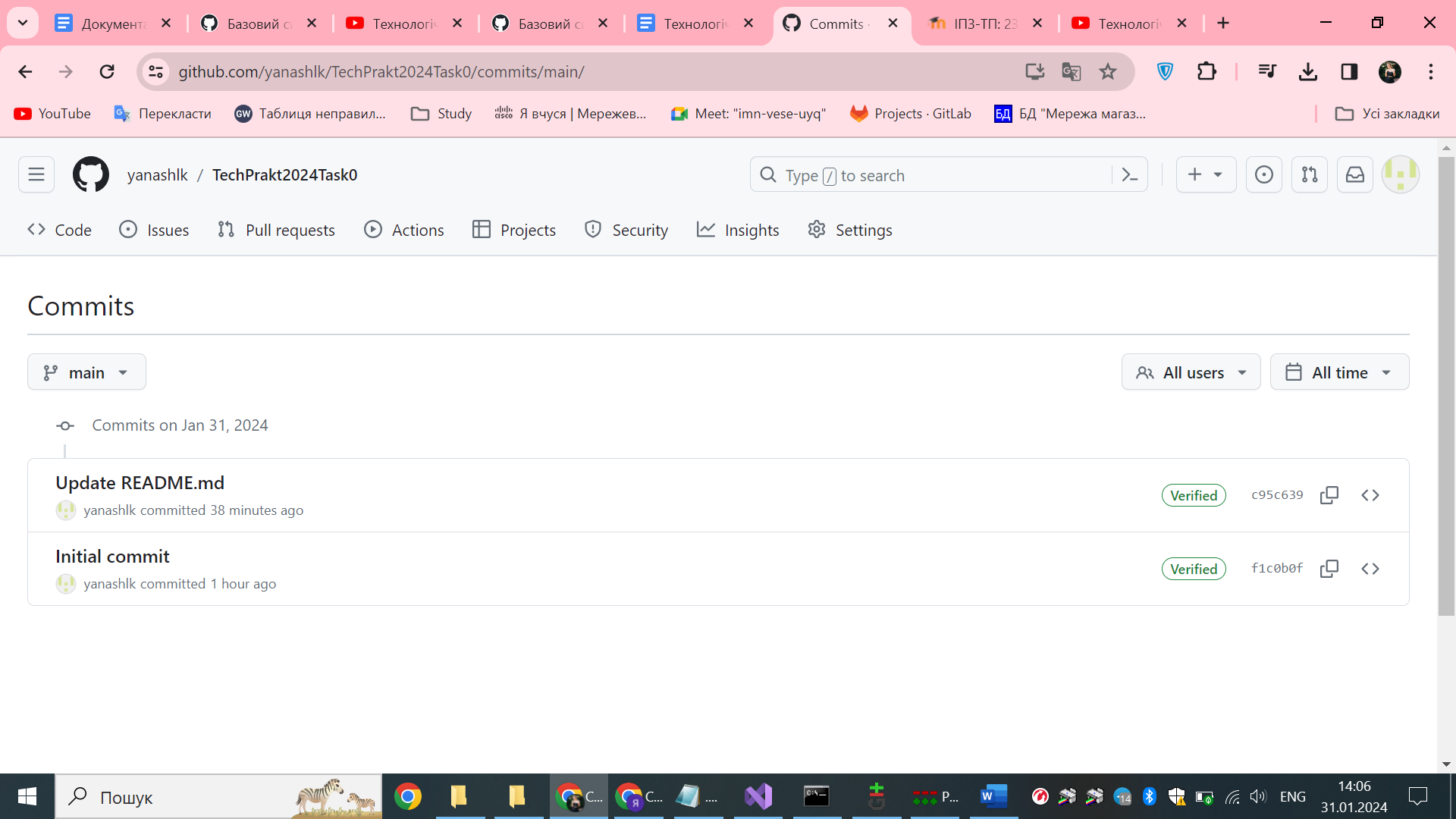


Рис.1.8. Коміт був успішно створений

1. Створіть файл файл **.gitignore**, у якому будуть ігноруватися:

4.1. директорії:

bin

obj

.vs

.idea

4.2. файл:

temporary.txt

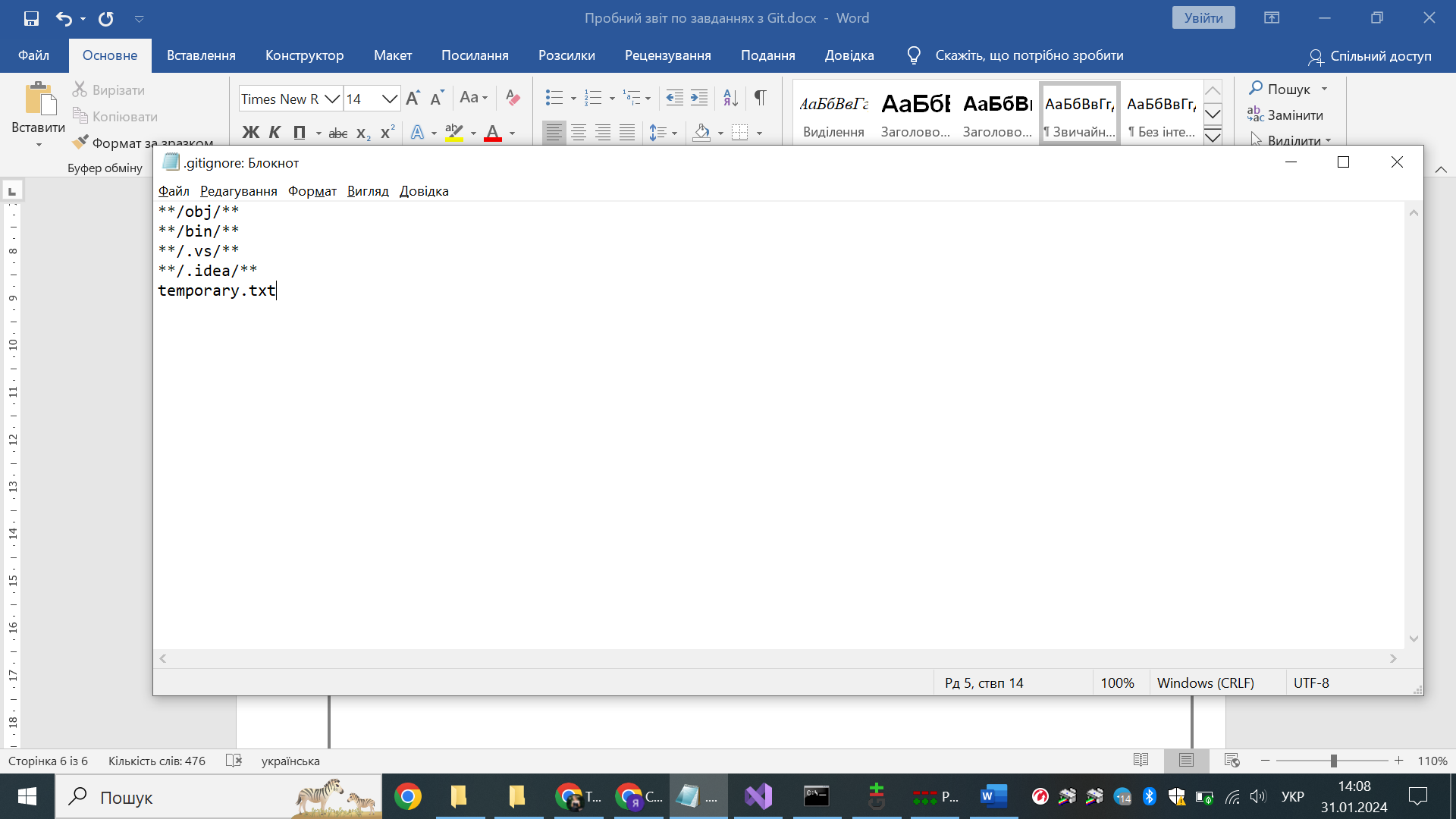


Рис.1.9. Файл .gitignore був успішно створений та необхідні данні, які не мають відображатись були додані

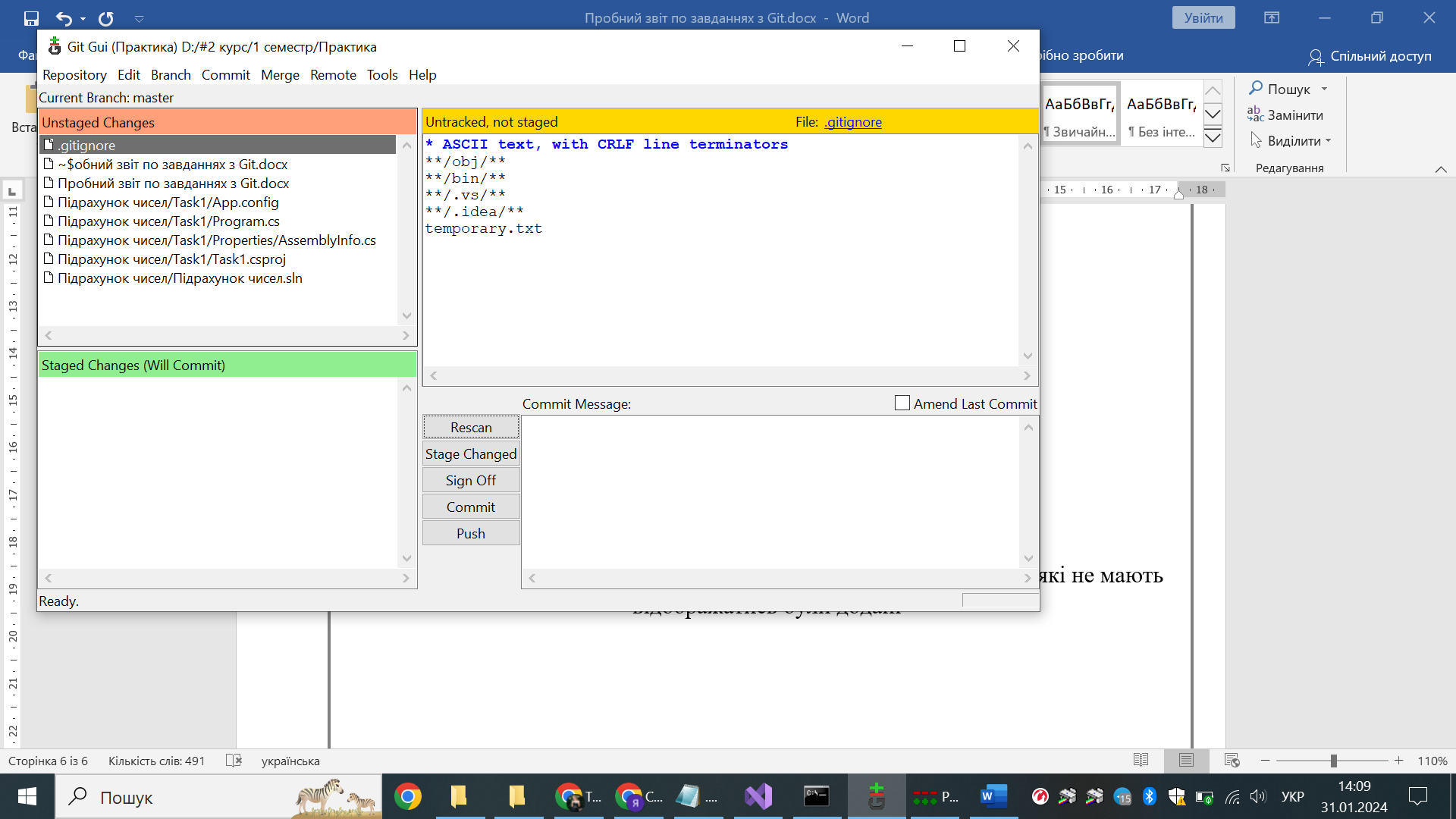


Рис.1.10.

Як бачимо, непотрібні файли не відображаються, а отже файл gitignore працює.

**Завдання 2. Fork та Pull Requests.**

1. Зробіть fork відкритого репозиторію до свого профілю:  
   <https://github.com/ZTUTeaching/TechPrakt2024Task1>
2. Клонуйте у себе на комп’ютері репозиторій **TechPrakt2024Task1** з вашого GitHub-аккаунту.
3. Створіть папку з назвою, яка відповідає вашому акаунту в університеті (наприклад: ipz228\_aaa).   
   У папці розмістіть файли **index.html** та **style.css** та **images**.
4. Зверстайте просту сторінку, що містить декілька новин з текстом та картинками. По мірі верстки зробіть 5 або більше комітів із зрозумілими описами.
5. Окремим комітом зробіть додавання рядка з вашим ПІБ та номером групи у файл **README.md**, що зберігається в кореневому каталозі репозиторію.
6. Внесіть зміни у будь-який файл в папці **jsdemo**. Зробіть коміт.
7. Вивантажіть усі ваші коміти у ваш репозиторій **TechPrakt2024Task1** на **GitHub**.
8. Зробіть **Pull Request** до репозиторію [**https://github.com/ZTUTeaching/TechPrakt2024Task1**](https://github.com/ZTUTeaching/TechPrakt2024Task1).

Після натискання fork публічний репозиторій перенесеться у власний акаунт, звідки можна вже буде зробити клонування до себе на пк.

Щоб зробити клонування, необхідно в cmd ввести команду -> git clone посилання на репозиторій.

Після створення відповідної папки там ми розміщуємо файли з кодом нашої веб-сторінки.

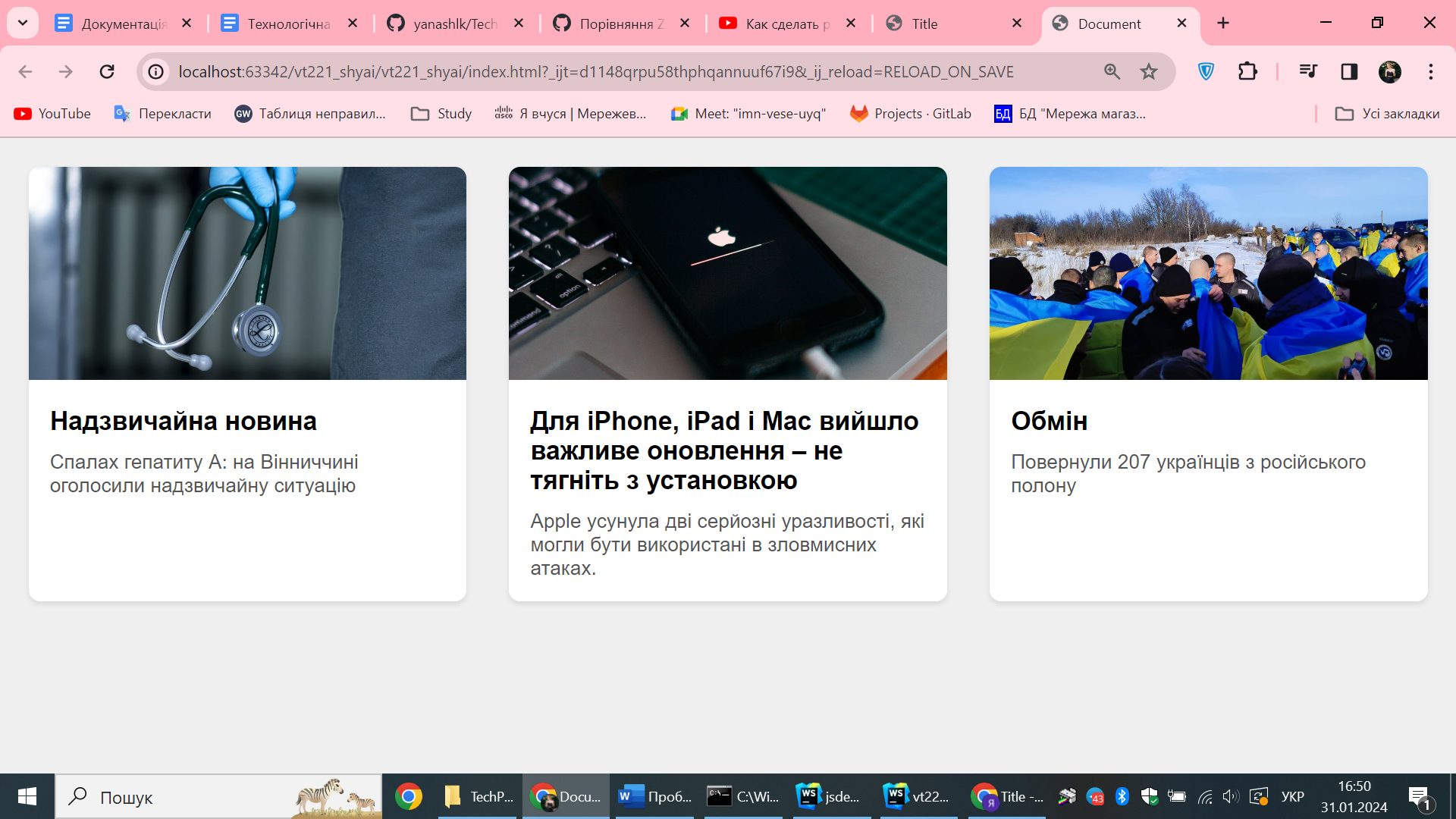


Рис.1.11. Приклад моєї простої веб-сторінки з новинами

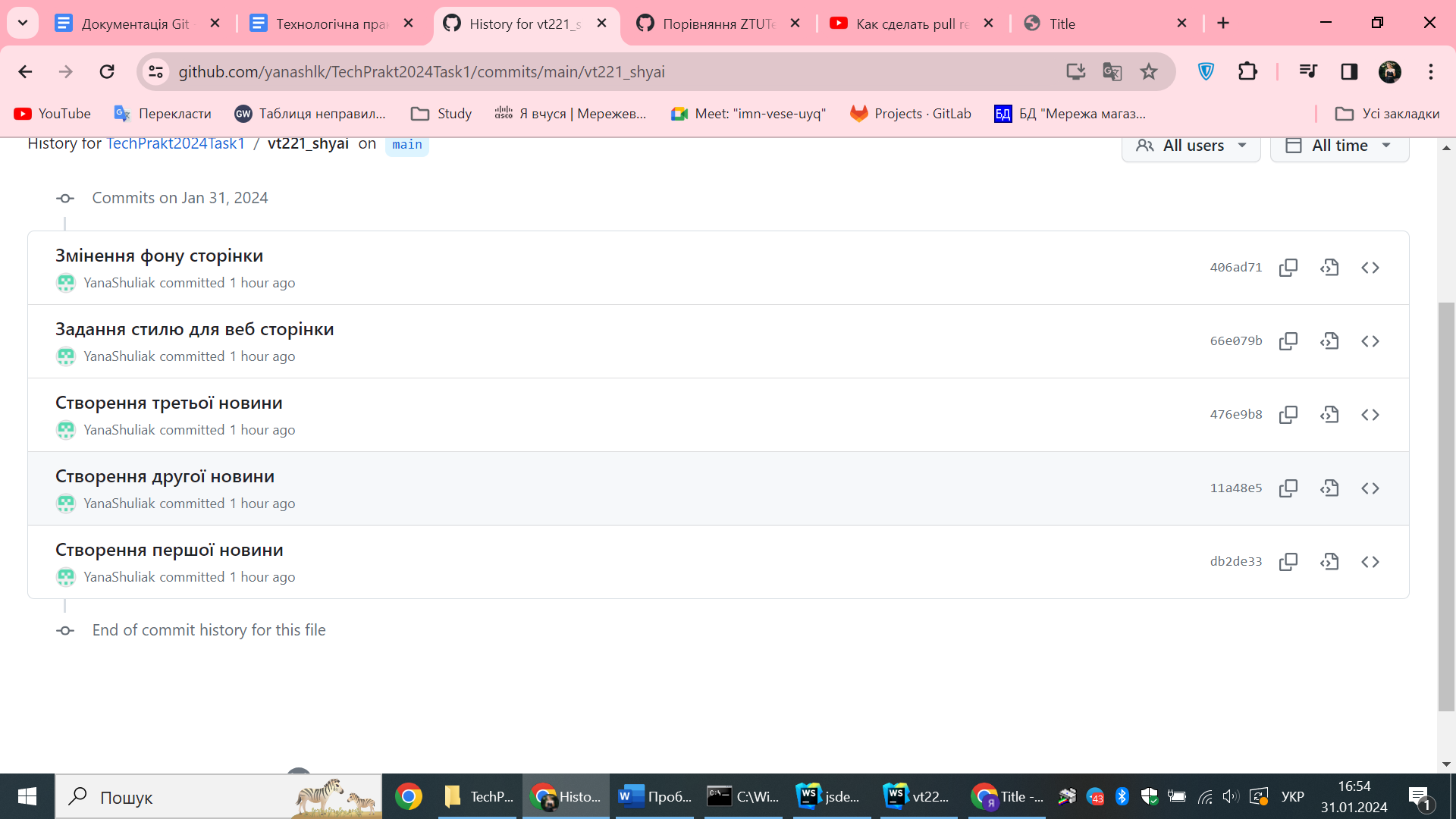


Рис.1.12. Створення 5 комітів під час верстки

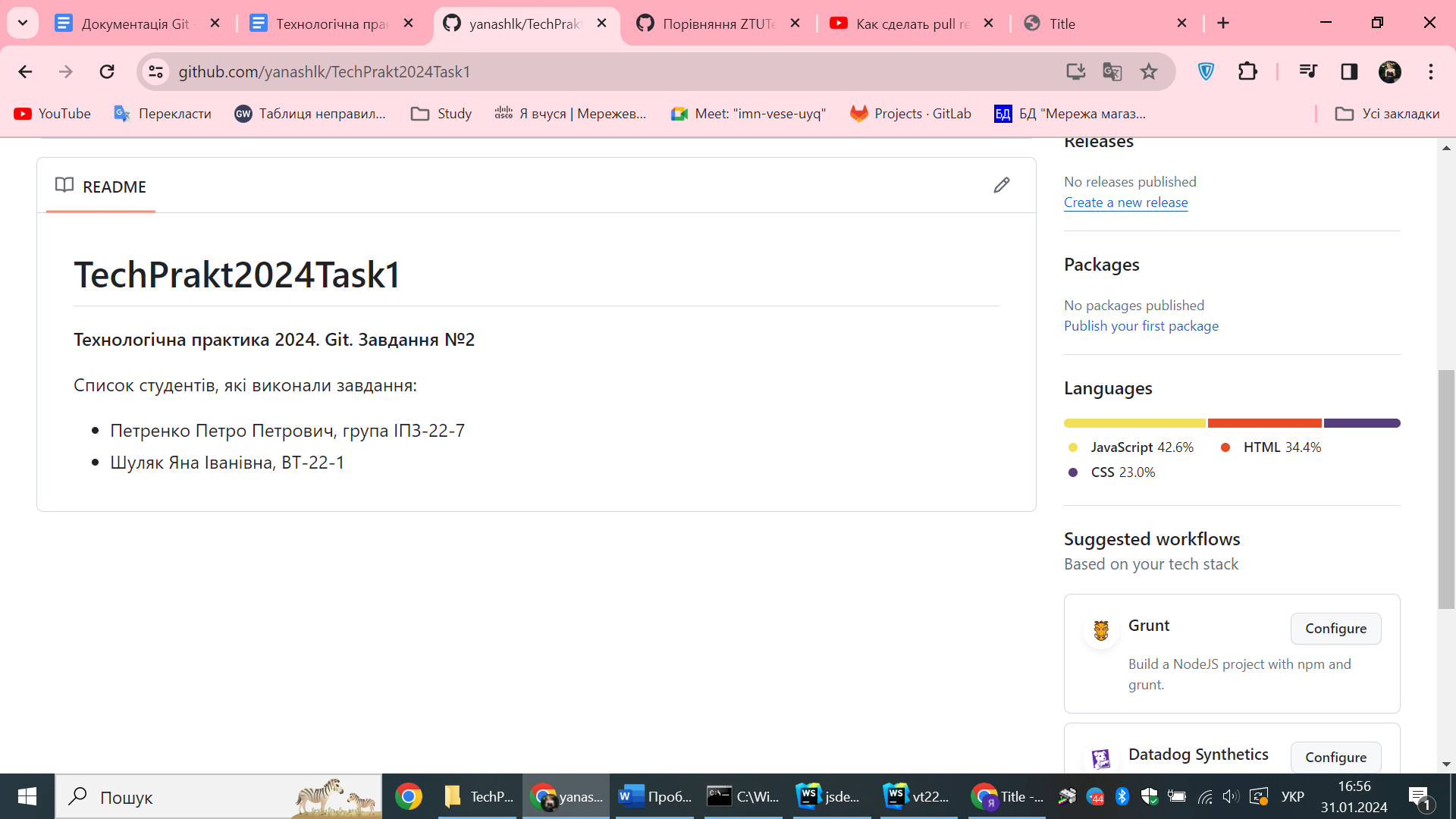


Рис.1.13. Додавання додаткової інформації в файл README

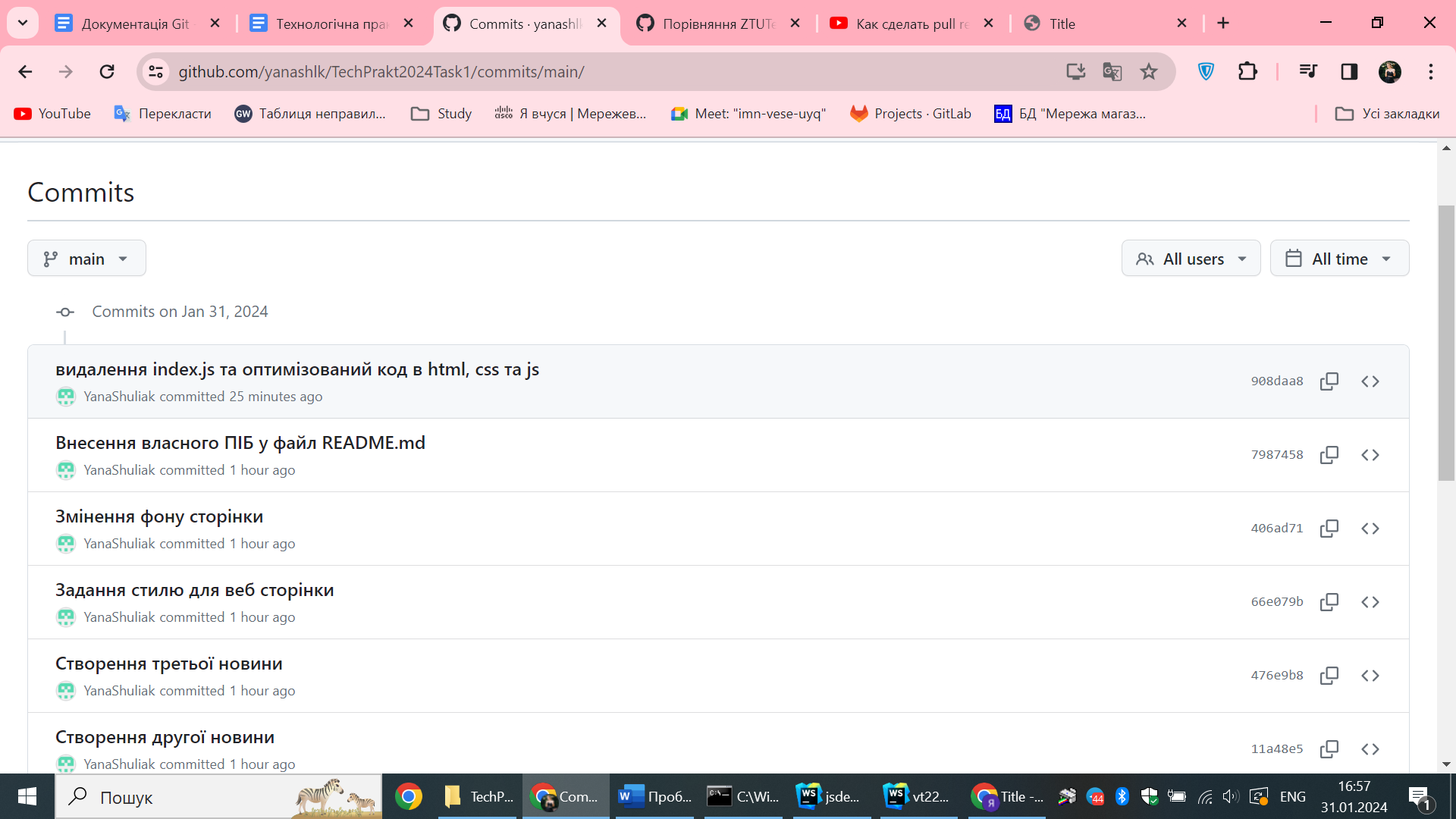


Рис.1.14. Закомічення змін

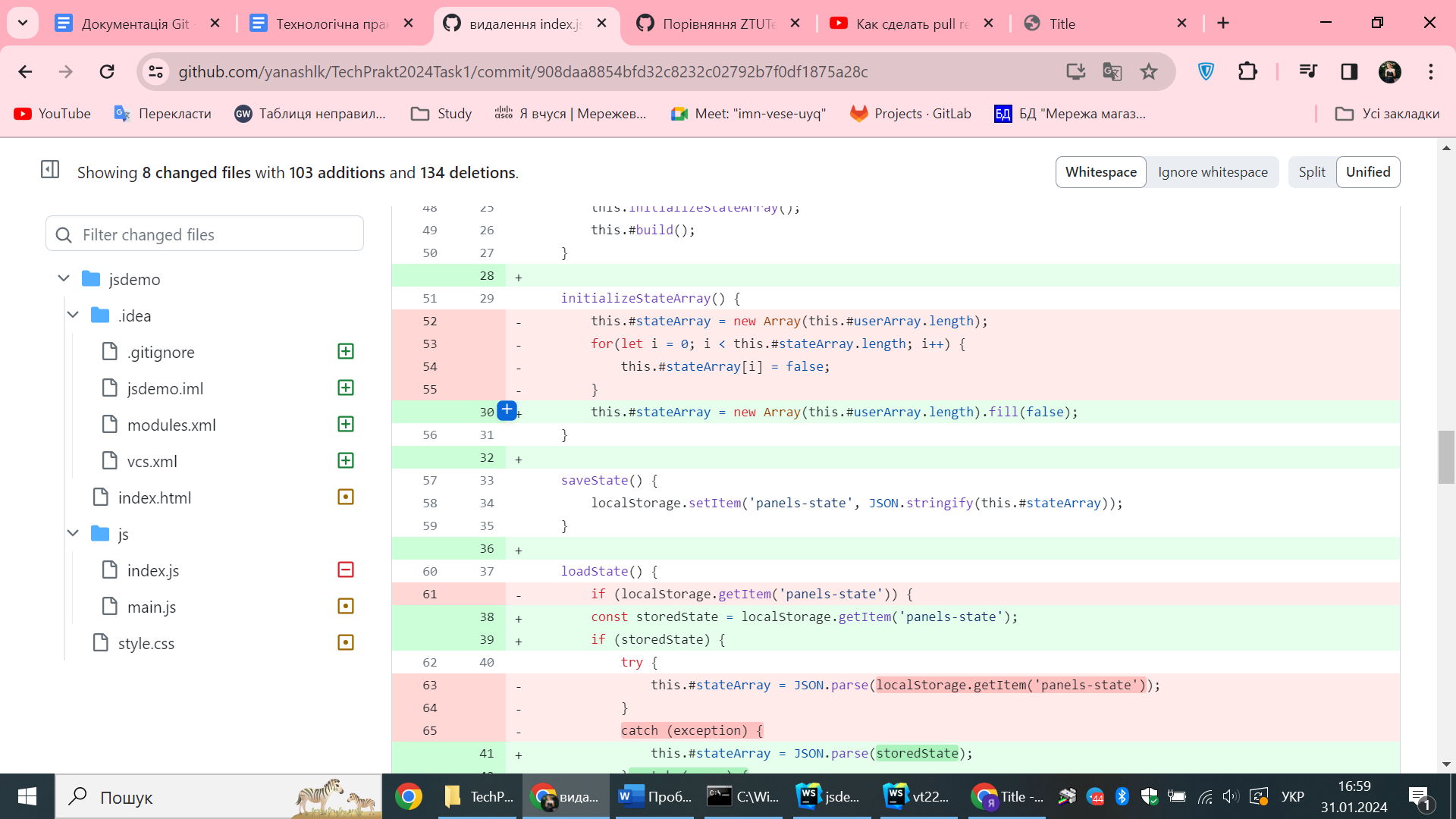


Рис.1.15. Внесення змін у файли в папці **jsdemo**

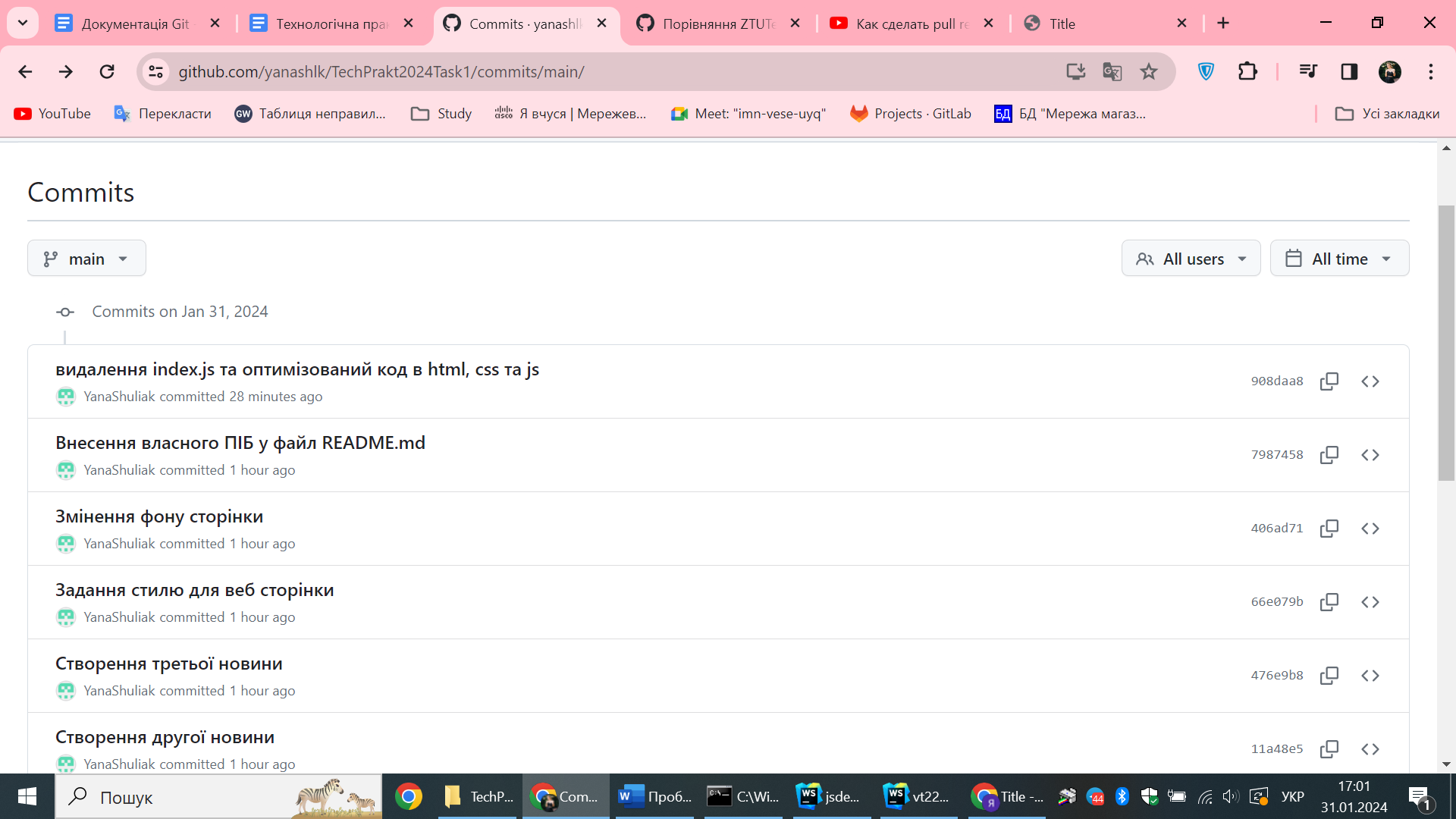


Рис.1.16. Закомічення змін та вивантаження всього на GitHub

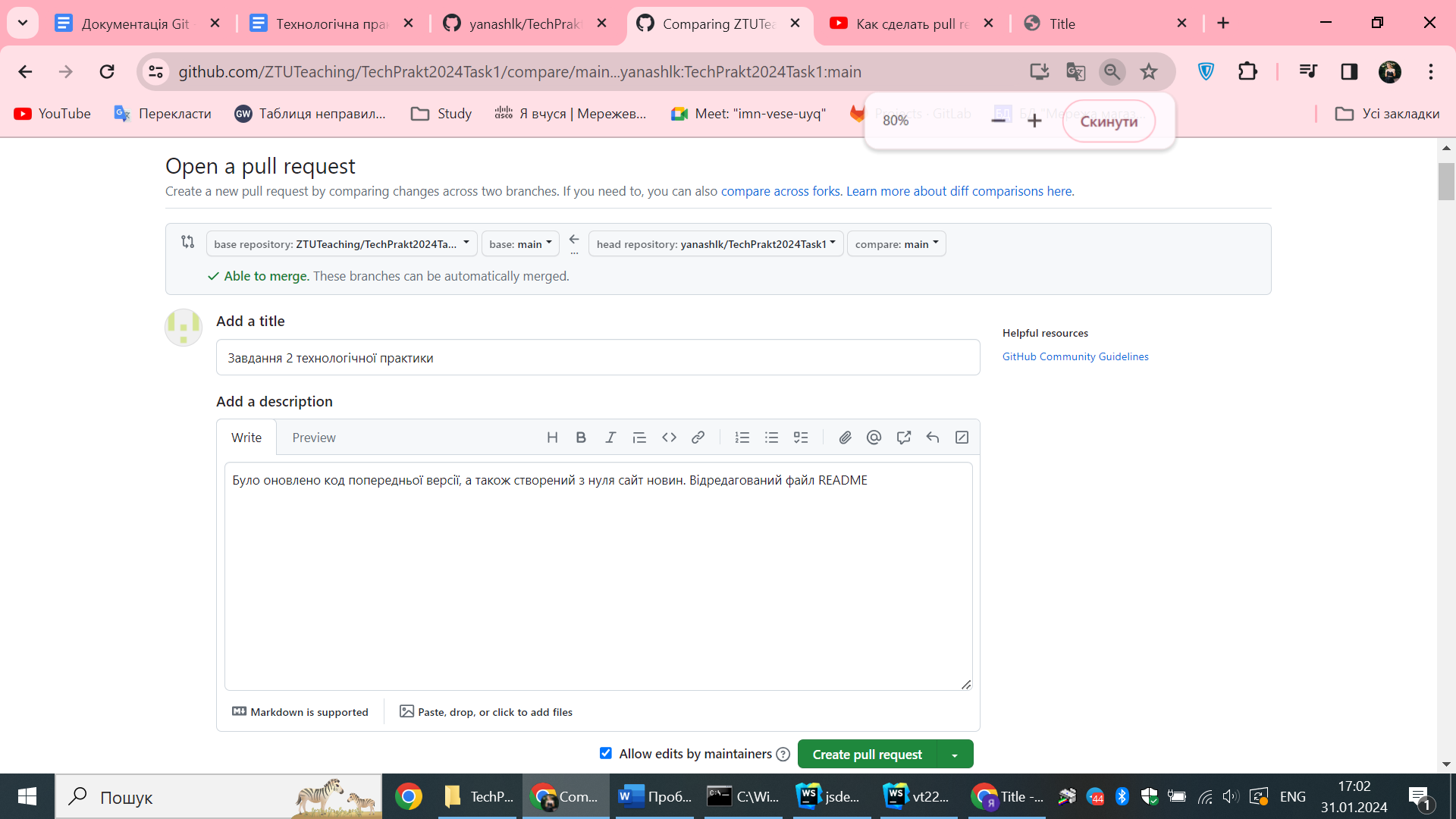


Рис. 1.17. Оформлення **Pull Request** до головного репозиторію