**Work-case 1**

**GitHub** — один з найбільших веб-сервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc (раніше Logical Awesome).

Git є набором утиліт командного рядка, які дозволяють відстежувати і записувати зміни в файлах (найчастіше коду, але ви можете відстежувати будь-що). З її допомогою ви можете відновити старі версії вашого проекту, порівнювати, аналізувати, об'єднувати зміни та багато іншого. Цей процес називається управлінням версіями*.* Є багато подібних систем контролю версій. Можливо, ви вже чули деякі з них: SVN, Mercurial, Perforce, CVS, Bitkeeper та ін.

Git працює децентралізовано, що означає, тобто не залежить від центрального сервера. Дані зберігаються локально в папці на вашому жорсткому диску, яка називається репозиторій. Проте ви також можете зберегти копію вашого репозиторія онлайн, що дає можливість команді людей працювати одночасно над одним кодом. Для цього використовуються такі сайти, як GitHub і BitBucket.

GitHub це система управління проектами і версіями коду, а також платформа соціальних мереж, створена для розробників. Але для чого використовується GitHub? Ну, серед іншого, це дозволяє вам працювати спільно з іншими людьми по всьому світу, планувати свої проекти і відстежувати свою роботу.

GitHub також є одним з найбільших онлайн-сховищ спільної роботи по всьому світу.

**Базові команди git:**

git clone адреса сховища - клонувати репозиторій на локальний компютер

git commit -m 'initial commit' комит змін в локальний репозиторій

git push -u origin master - тільки перший раз відправка змін в віддалений репозиторій

git push - всі наступні рази відправка змін в віддалений репозиторій

git pull - скачується актуальна версія віддаленого сховища та всі зміни застосовуються до локального сховища

checkout - перейти в іншу гілку

discard - не відправляти в репозиторій ті зміни які нам не подобаються

revert - відкотити існуючі Коміто

hard reset -откатіть цілу пачку комита

merge - спочатку підтягуємо в свою гілку з master всі зміни

merge - потім тестуємо і робимо злиття переходимо (checkout) в master і застосовуємо (merge) всі зміни з нашої гілки

git clone - скачування сховища

git status - перегляд поточних змін

git add - додати файл в локальний репозиторій

git commit - Комміт в локальний репозиторій

git push - Комміт в віддалений репозиторій

Робота з гілками

git branch -a подивитися в якій гілці ми знаходимося і показати все гілки ключ -а all

git branch newbranch - створити нову гілку з ім'ям newbranch

git checkout newbranch - перейти в нову гілку

git push origin newbranch - (спочатку переключитися на майстер) відправити в віддалений репозиторій гілку newbranch

git branch -d newbranch - видалити гілку newbranch з локального сховища

git push origin --delete newbranch - видалити гілку newbranch в віддаленому репозиторії