**Лабораторна робота №6**

**Тема:** система контролю версій.

**Мета:** отримати практичні навички роботи з графічними застосунками системи контролю версій.

**Теоретичні відомості:**

Системи керування версіями зазвичай використовуються при розробці програмного забезпечення для відстеження, документування та контролю над поступовими змінами в електронних документах: у сирцевому коді застосунків, кресленнях, електронних моделях та інших документах, над змінами яких одночасно працюють декілька людей.

Кожна версія позначається унікальною цифрою чи літерою, зміни документу занотовуються. Зазвичай також зберігаються дані про автора зробленої зміни та її час.

Інструменти для контролю версій входять до складу багатьох інтегрованих середовищ розробки.

Працюючи з системою контролю версій Git, користувач процює з локальним репозиторієм, для внесення змін до якого необхідно виконати команду «commit». Для внесення змін у віддалений репозиторій необхідно виконати «push». Зворотній процес називається «pull».

Копіювання існуючого віддаленого репозиторію здійснюється командою «clon».

**Алгоритм виконання лабораторної:**

1. спершу необхідно зареєструватись на github.com (реєстрація безкоштовна);
2. після реєстрації необхідно створити репозиторій (рис. 1);

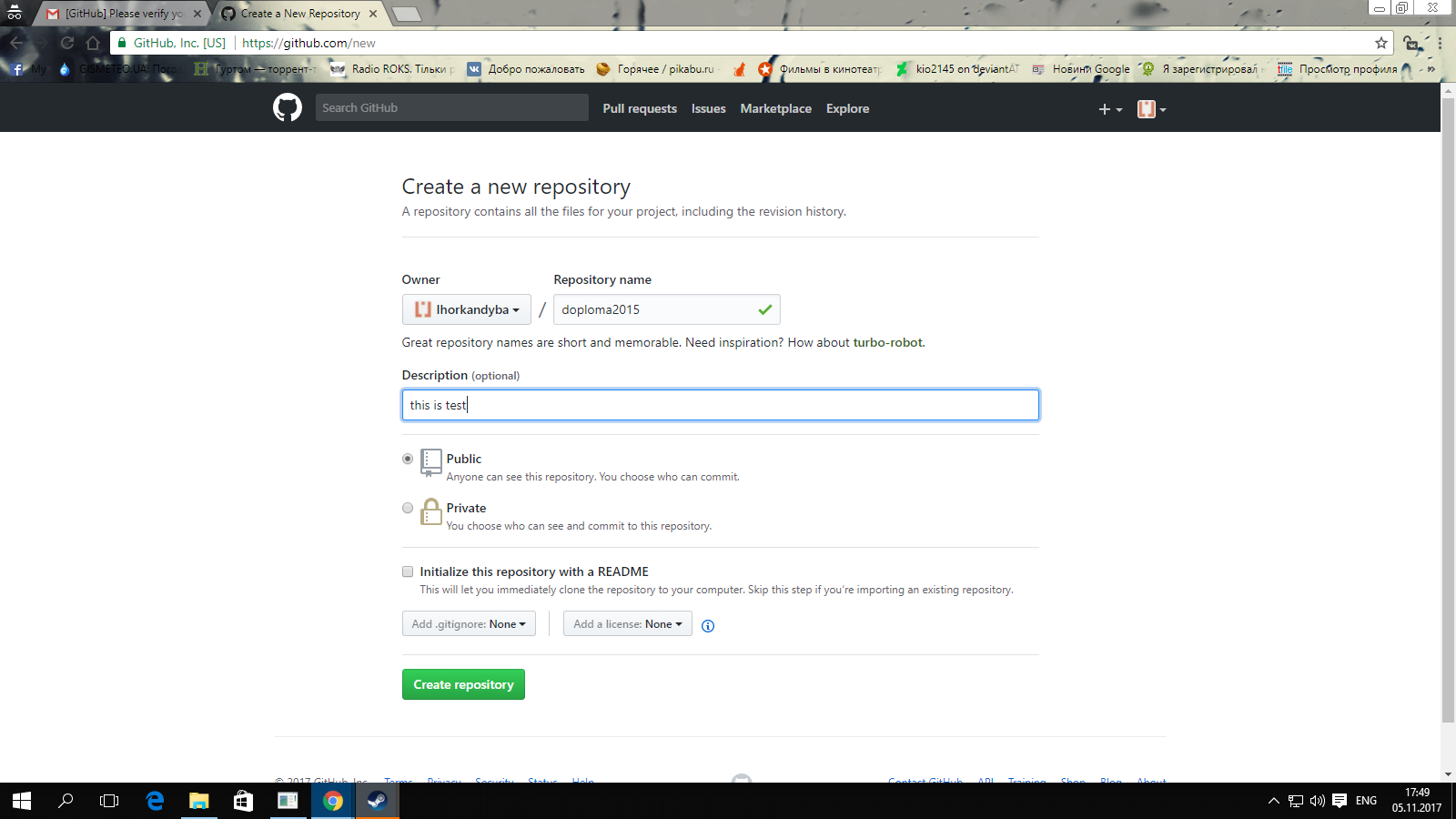


Рисунок 1 – Приклад створення репозиторію

1. працювати зі створеним репозиторієм можна використовуючи протокол SSH або HTTPS (для роботи з використанням протоколу SSH потрібна генерація спеціального ключа, тому для початку роботи бажано використати протокол HTTPS);



Рисунок 2 – Приклад вибору протоколу

1. завантаження вашого коду можна здійснити різними клієнтами, найбільш простим клієнтом є Smartgit, перший запуск якого вимагає налаштувань, а саме створення локального репозиторію, який буде в подальшому синхронізувати з github;

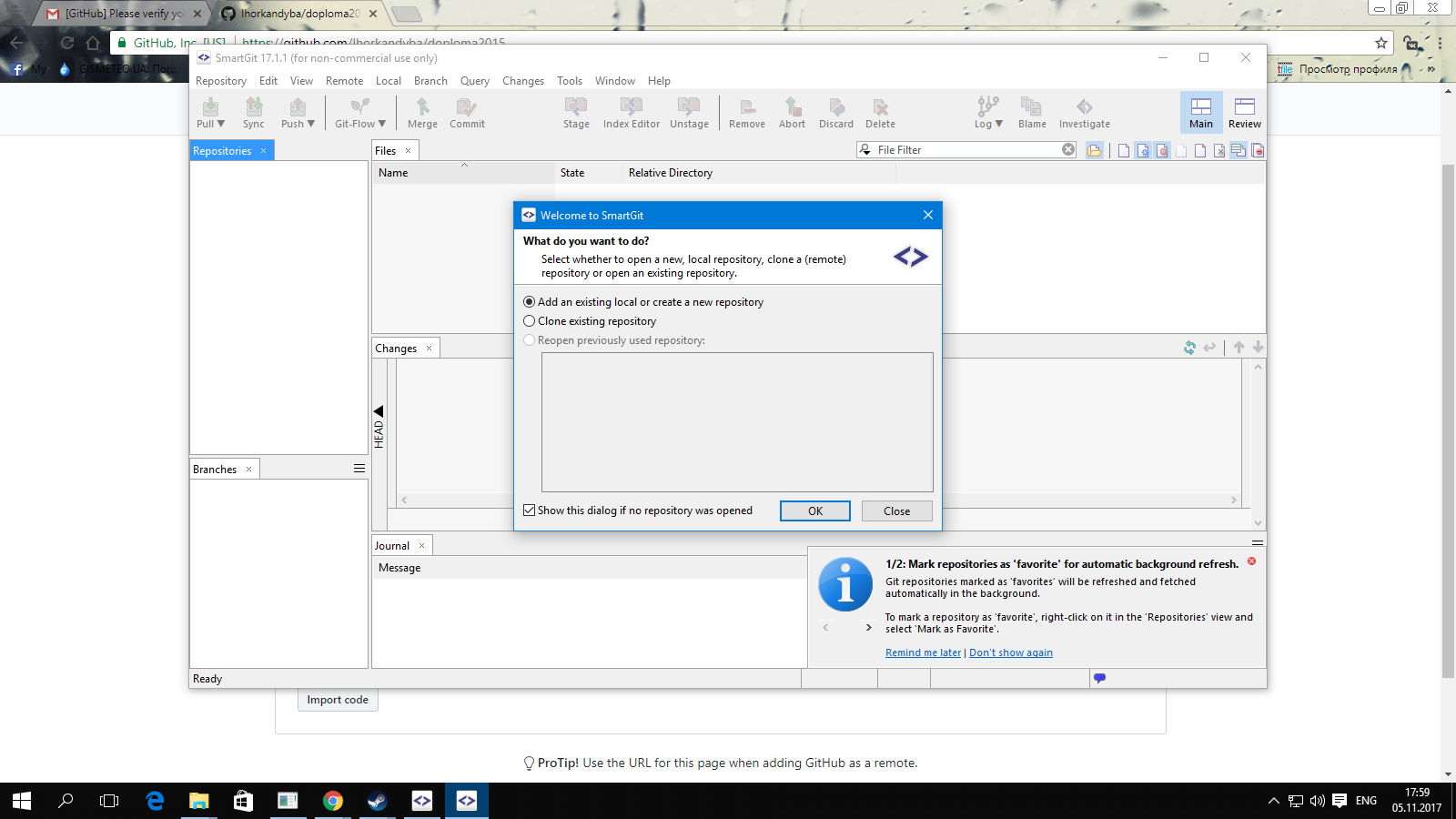


Рисунок 3 – Початок створення репозиторію

1. після створення локального репозиторію, необхідно натиснути кнопку «pull» (завантажити зміни з репозиторію, за замовчення github створю текстовий документ з ліцензією), але на даному етапі необхідно провести додаткові налаштування: ввести денні з github (рис. 4) для цього необхідно натиснути «Generate API Token»;

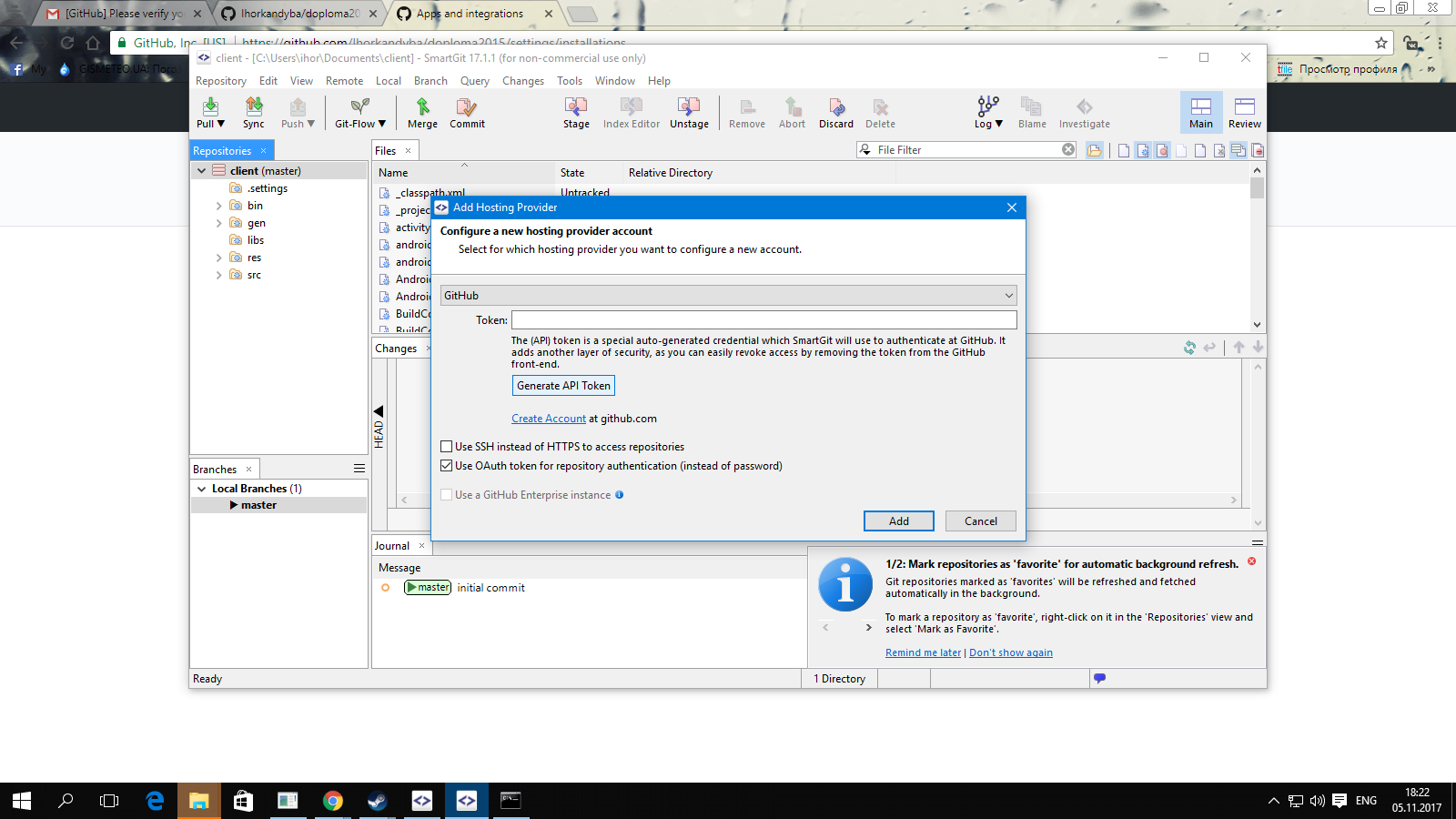


Рисунок 4 – Налаштування віддаленого доступу github

1. у випадку коректного введення даних Smartgit запропонує обрати створений раніше репозиторій (рис. 5);

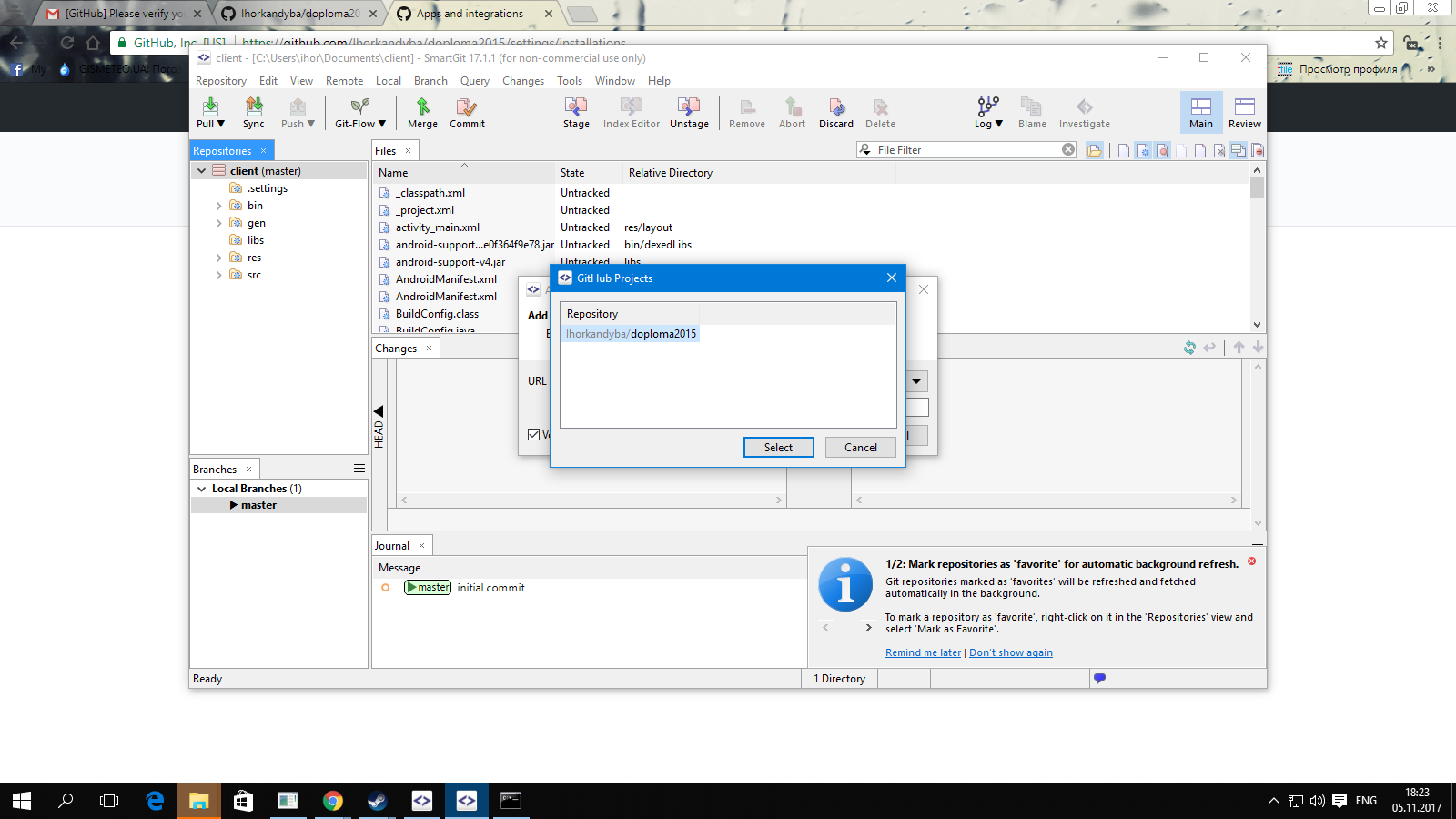


Рисунок 5 – Вибір віддаленого репозиторію

1. дані завантажені з віддаленого серверу, при внесенні змін користувачу необхідно проводити «commit» (для внесення змін у локальний репозиторій) у відповідному контекстному мені, обов’язково заповнити поле з інформацією про внесені зміни, після натиснути кнопку «commit» (локально) або «commit&push» (після внесення у локальний репозиторій дані будуть відправлені на віддалений);

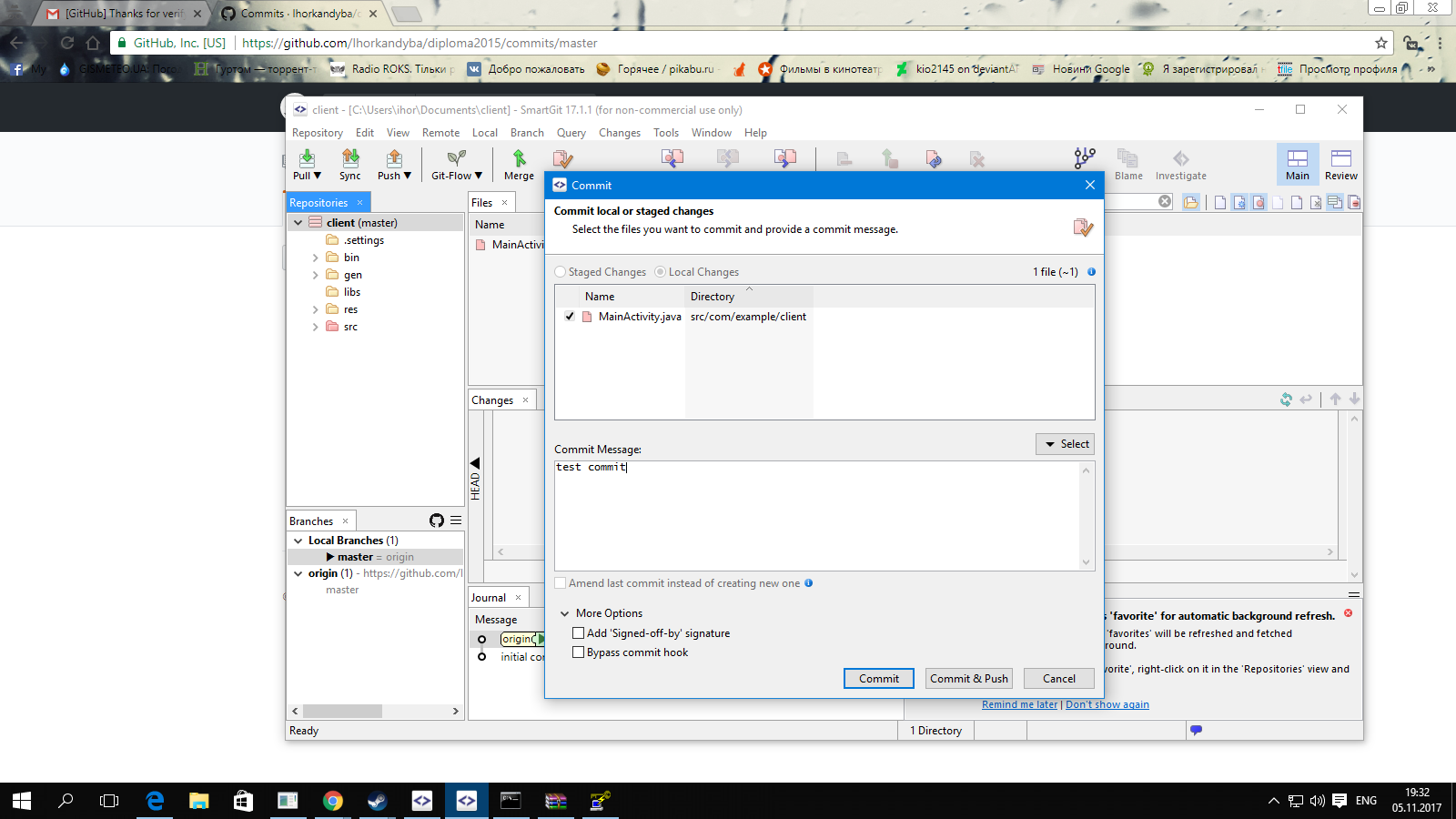


Рисунок 6 – Меню створення commit

1. перевірити внесені зміни та деталі можна безпосередньо на вебсайті github;

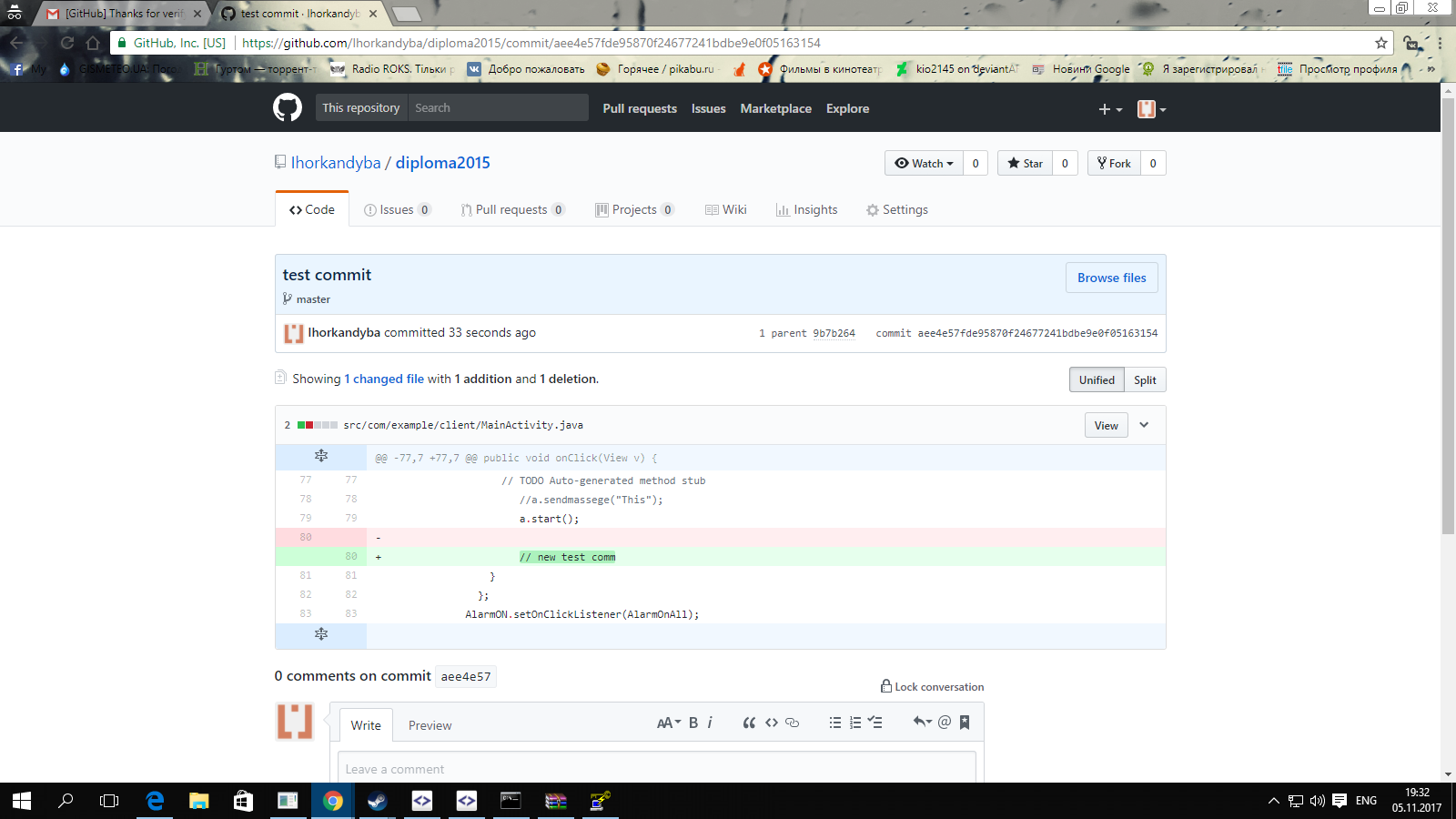


Рисунок 6 – Відображення змін на вебсайті github

**Завдання:**

1. створити акаунт та репозиторій на github;
2. відправити одну з лабораторних робіт дисципліни «основи програмування» на свій віддалений репозиторій;
3. змінити частину коду лабораторної, створити commit та відправити зміни на віддалений репозиторій;
4. перевірити дані на вебсайті github;
5. оформити звіт згідно з методичними вказівками;

**Завдання на максимальний бал:** підключити віддалений репозиторій **git** до середовища розробки (наприклад, Visual Studio).