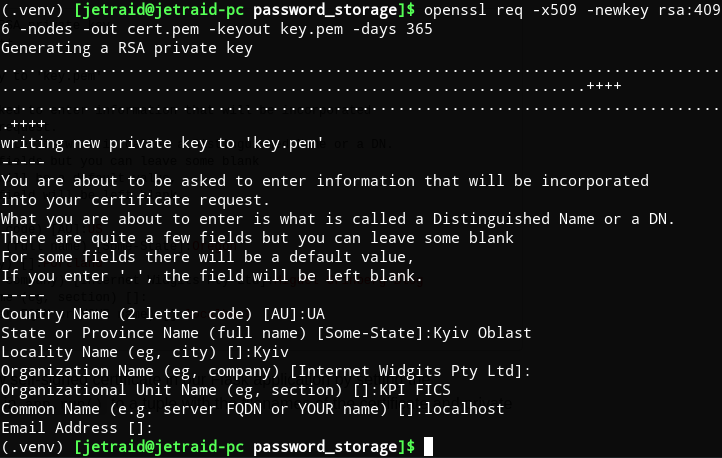
Лабораторна робота виконана на основі клієнт-серверної програми для лабораторної роботи sensetive\_information.

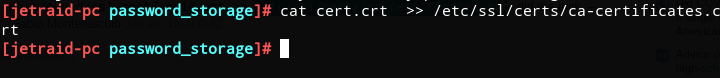
Протоколом шифрування каналів був обраний TLS версії 1.2 (TLSv1\_2). Це найсучашніша версія і найкращий вибір для максимального захист. В цій версії були заборонені старі криптографічні хеш-функції MD5 і SHA-1, а саме їх замінили на SHA-256.

Для цього спочатку згенеруємо сертифікат за допомгою *openssl*:

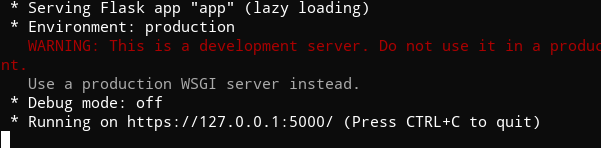
Після цього спробуємо додати його до нашого веб застосунку використовуючи біліотеку *ssl* для створення контексту для *Flask app.*

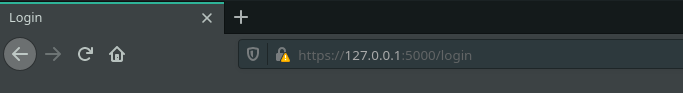
Нові версії фласку підтримують ssl нативно, тому просто передаємо контекст.

Як бачимо для створення контексту потрібно вказати шлях до файлів з сертифікатом та ключем.

Після продіяного можна спробувати запустити застосунок, але ми всеодно не отримаємо бажаного результату, оскільки сертифікат повинен бути *trusted.* Тобто таким якому довіряє браузер, оскільки це *self-signed* сертифікат то найпростіше що ми можемо зробити це додати його до списку корньових довірених сертифікатів.

Після цього вже можна спробувати запустити застосунок.

Можемо спостерігати що тепер він спілкується по протоколу HTTPs.

І браузер вказує що сайт зашихений, проте й повідомляє що сертифікат само підписаний.

Щодо зберігання сертифікату та ключа. Сертифікат є публічною інформацією, тобто немає різниці де його зберігати. А ключ потрібно захистити. Найчастіше це роблять таким чином: створюють окремого користувача (або просто використовують root), або групу яка має права на читання приватних ключів якими користується сервер, і змінюють права на доступ до нього за допомогою *chmod.* В нашому випадку сертифікат та ключ зберігаються поряд із застосунком у директорії *data.*