1. Возврат False в функциях MySQL\_Get и searchResult позволяют вызвать функцию NoItemFound, которая:

* Сгенерирует HTML – страницу с «No result for: %s (тут это будет поисковым запросом)»
* Вместе с этим будет использован render\_template\_string для рендеринга шаблона

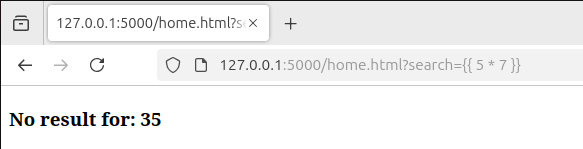
1. Маршрут /home.html получит search из строки запроса и экранирует пользовательский ввод с помощью html.escape
2. Функция использует render\_template\_string, которая уязвима к SSTI, несмотря на наличие экранирования, ибо render\_template\_string принимает строку как шаблон Jinja2.

**КАК ВОСПРОИЗВЕСТИ:**

1. Поднимаем flask-приложение:

*python app.py*

1. Переходим по запросу со встроенным кодом (как пример, используем умножение чисел). Если интерпретатор выполнит выражение, то атака успешна



**Как защитить:**

1. Использование **render\_template**вместо**render\_template\_string**:

- render\_template безопасен, так как он не интерпретирует пользовательский ввод как шаблон

2. Можно использовать шаблонизатор Jinja2.sandboxedEnvironment, что ограничит инициализацию произвольного кода.