

| Business Template  **Test Automation & UNIT TESTING Basics** |
| --- |
| Task 1 **Provide the Python code based on the** [**fixture\_task.py**](https://github.com/AntonLazarchik/DQE_LAB_2024_CODE_SAMPLES/blob/main/fixture_task.py) **which will contain 2 fixtures. The first fixture is used to track the time of the test execution. The second will track the time of the whole suite execution. All tests must use fixture 1 except the last one. The fixture for the suite time execution must be invoked once.**  **What is expected to be used in the task: yield in fixtures, fixture scopes, autouse flag, only time module.**  **Expected result: The Python file with implemented code. The screenshot with run results.**    **Link to documentation:**  [**https://docs.pytest.org/en/6.2.x/fixture.html#yield-fixtures-recommended**](https://docs.pytest.org/en/6.2.x/fixture.html#yield-fixtures-recommended)   Task 2 **You need to write the logic for the tests and use parametrisation for the test\_add\_numbers function based on the code** [**parametrize\_task.py**](https://github.com/AntonLazarchik/DQE_LAB_2024_CODE_SAMPLES/blob/main/parametrize_task.py)**. The names of the cases must be also used from the config file** [**config.yaml**](https://github.com/AntonLazarchik/DQE_LAB_2024_CODE_SAMPLES/blob/main/config.yaml)**.**  **For test\_add\_invalid\_types you need to verify that the specific exception type is returned. Also, mark the test\_add\_numbers as smoke and test\_add\_invalid\_types as critical. Launch tests 3 times using the flags in pytest to run only smoke and second time only critical path tests.**  **What is expected to be used in the task: @pytest.mark, pytest.raises, assert, pytest run commands and flags.**  **Expected result: The Python file with implemented code. The screenshot with 3 run results. First run only for tests marked as smoke. Second run for critical path tests. The third run for the whole suite.**    **Links to documentation:**  [**https://docs.pytest.org/en/6.2.x/mark.html**](https://docs.pytest.org/en/6.2.x/mark.html)  [**https://docs.pytest.org/en/6.2.x/parametrize.html**](https://docs.pytest.org/en/6.2.x/parametrize.html)   Task 3 **General Overview: You need to implement the basic framework to test the tables in your database.**  **1. 6 tests must be implemented: 3 for smoke tests like the presence of the objects. 3 for the critical path. Any tests can be used for smoke or critical path tests. The main requirement is to have the possibility to run smoke or critical path tests in the same suite or separately.**  **2. The result of the test run must be provided using the allure. 1 Report for smoke tests. 1 report for critical path. 1 for the whole suite (smoke + critical). Use the ‘–single-file’ flag.**  **3. Your tests must have custom marks for smoke and critical paths.**  **4. The SQL queries must be passed through ‘parametrize’ functionality and stored in config files. Config structure example:** [**config\_SQL\_example.yaml**](https://github.com/AntonLazarchik/DQE_LAB_2024_CODE_SAMPLES/blob/main/config_SQL_example.yaml)  **5. The connection to the DB must be implemented and passed to the tests through a fixture. The connection must be closed as a teardown step.**  **6. conftest.py must be used to store the global configs and fixtures.**  **7\*. Generate the allure report during the test run. (pytest\_sessionfinish and subprocess)**  **The connection to the DB could be established using the pyodbc module for SQL Server or the psycopg2 module for Postgres. The configuration to connect to the DB can be found in the code samples** [**useful\_stuff.py**](https://github.com/AntonLazarchik/DQE_LAB_2024_CODE_SAMPLES/blob/main/useful_stuff.py)**.**  **What is expected to be used in the task: @pytest.mark, assert, allure commands, allure.step, conftest, pytest.ini, pytest run commands and flags.**  **Expected result: The Python files with implemented code. 3 Allure reports (First run only for tests marked as smoke. Second run for critical path tests. The third run for the whole suite)**          **Links to documentation:**  [**Python PostgreSQL Tutorial Using Psycopg2 [Complete Guide] (pynative.com)**](https://pynative.com/python-postgresql-tutorial/#h-install-psycopg2-using-the-pip-command)  [**https://allurereport.org/docs/gettingstarted-installation/**](https://allurereport.org/docs/gettingstarted-installation/)  [**https://allurereport.org/docs/pytest/**](https://allurereport.org/docs/pytest/) |