##### **Итоговый проект по автоматизации тестирования функционала страницы** [***https://lk.rt.ru/***](https://lk.rt.ru/)**сайта "Ростелеком"**

При тестировании сайта были написаны:

* тест-кейсы;
* баг-репорты;
* автоматизированные тесты.

При тестировании сайта были применены следующие техники тест-дизайна:

* разбиение на классы эквивалентности;
* анализ граничных значений.

Данные техники применялись для полей ввода данных при тестировании негативных сценариев.

Некоторые автотесты требуют ручной ввод капчи с картинки, т.к. срабатывает защита сайта.

Папка pages содержит следующие файлы:

* api\_reg\_email.py - GET-запросы к виртуальному почтовому ящику (1secmail.com) для получения валидного email и кода для регистрации на сайте и восстановления пароля;
* elements.py - функции для взаимодействия с элементами страницы сайта при проведении автотестов;
* base.py - функции для получения главной страницы сайта и пути текущей страницы;
* rostelecom.py - функции для взаимодействия с url страницами и локаторы для элементов сайта;
* settings.py - учетные данные, используемые в процессе теста.

Папка tests содержит следующие файлы:

* test1\_positive\_registration.py - позитивные тесты страницы регистрации;
* test2\_positive\_authorisation.py - позитивные тесты страницы авторизации;
* test3\_positive\_recovery\_pass.py - позитивные тесты страницы восстановления пароля;
* test4\_negative\_registration.py - негативные тесты страницы регистрации;
* test5\_negative\_authorisation.py - негативные тесты страницы авторизации;
* test6\_negative\_recovery\_pass.py - негативные тесты страницы восстановления пароля.

Также проект содержит такие файлы, как:

* conftest.py - фикстуры для работы с браузером;
* pytest.ini - маркеры для параметризации;
* requirements.txt - используемые при тестировании библиотеки PyCharm.

Для подготовки к запуску автотестов необходимо установить необходимые библиотеки PyCharm с помощью вводимой команды в консоли терминала:

pip install -r requirements.txt

Также необходимо ввести валидные данные уже авторизованного пользователя на сайте в файле .env и скачать актуальную версию драйвера для вашего браузера для успешного прохождения автотестов.

Для запуска автотестов необходимо вводить команды в консоли терминала.

Для позитивных тестов страницы регистрации:

python -m pytest -v --driver Chrome --driver-path <chromedriver\_directory>\<chromedriver\_file> tests\test1\_positive\_registration.py -k TestPositivePageRegistration

Для позитивных тестов страницы авторизации:

python -m pytest -v --driver Chrome --driver-path <chromedriver\_directory>\<chromedriver\_file> tests\test2\_positive\_authorisation.py -k TestPositivePageAuthorisation

Для позитивных тестов страницы восстановления пароля:

python -m pytest -v --driver Chrome --driver-path <chromedriver\_directory>\<chromedriver\_file> tests\test3\_positive\_recovery\_pass.py

Для негативных тестов страницы регистрации:

python -m pytest -v --driver Chrome --driver-path <chromedriver\_directory>\<chromedriver\_file> tests\test4\_negative\_registration.py -k TestNegativePageRegistration

Для негативных тестов страницы авторизации:

python -m pytest -v --driver Chrome --driver-path <chromedriver\_directory>\<chromedriver\_file> tests\test5\_negative\_authorisation.py -k TestNegativePageAuthorisation

Для негативных тестов страницы восстановления пароля:

python -m pytest -v --driver Chrome --driver-path <chromedriver\_directory>\<chromedriver\_file> tests\test6\_negative\_recovery\_pass.py -k TestNegativePageRecoveryPass

<chromedriver\_directory>\<chromedriver\_file> - путь к директории файла драйвера\название файла браузера. Например: C:\Chrome-selenium\chromedriver.exe

<https://github.com/RomanQA89/Graduation_project_Rostelecom.git>