***Дипломная работа: практический пример для компании «Ростелеком Информационные Технологии»***

***Автоматизация тестирования страницы https://b2c.***

***passport.rt.ru сайта «Ростелеком»***

***Бриф от заказчика: см.бриф***

***В рамках проверки сайта были разработаны:***

★ документы с ручными чек-листами, тест-кейсами, а также баг-репорты Итогового проекта;

★ автоматизированные сценарии тестирования.

***Во время тестирования ресурса были использованы такие подходы тест-дизайна:***

★ деление на эквивалентные классы значений;

★ пограничный анализ;

★ проработка тестирования состояний и переходов.

***В ходе тестирования применялись следующие библиотеки на базе PyCharm:***

requests

python-dotenv

pytest

selenium

faker

***Запуск автотестов реализован через Run.***

Ссылка на репозиторий на GitHub.

***ОБРАТИТЕ ВНИМАНИЕ!!!***

Для страниц восстановления пароля при выполнении как положительных, так и отрицательных тестов требуется САМОСТОЯТЕЛЬНО ввести символы, изображённые на “капче”, в поле для ввода “капча”, задержка для ввода включена автоматически.

Чтобы проверить авторизацию по email и паролю, также обязательно ручное введение капчи!

***Краткий обзор директории:***

***В структуре проекта имеются две директории — pages и tests, а также файлы conftest.py и pytest.ini.***

❖ conftest.py — содержит фикстуру для инициализации браузера.

❖ pytest.ini — содержит маркеры, необходимые для параметризации тестов.

***Папка pages включает в себя такие скрипты:***

➔ registration\_email.py — отправляет GET-запросы к виртуальной почте для получения корректного email и проверочного кода, которые применяются для регистрации пользователя и восстановления доступа;

➔ config.py — хранит основной адрес тестируемого интернет-ресурса;

➔ auth.py — содержит обёртки для работы с локаторами, разбитые по классам в зависимости от типа тестирования;

➔ base.py — определяет методы работы с локаторами через явные ожидания, а также функции получения главной и текущей страниц;

➔ locators.py — определяет XPath- и CSS-локаторы для элементов сайта;

➔ settings.py — содержит различные данные, которые используются в ходе тестирования.

***Директория tests содержит следующие тестовые файлы:***

➔ test\_auth\_page\_positive — выполняет позитивные проверки страницы входа (для теста автопереключения вкладок берётся действительный номер лицевого счета, корректно оформленный (не активен), ручные проверки проходят корректно, но при автоматизированном переключении этого не происходит: две последние цифры не учитываются, и система переводит на страницу с вводом телефонного номера);

➔ test\_auth\_page\_negative — тестирует негативные сценарии на странице авторизации;

➔ test\_new\_password\_positive — проверяет успешное восстановление пароля;

➔ test\_new\_password\_negative — отвечает за сценарии с ошибочным восстановлением доступа;

➔ test\_registr\_positive — набор успешных тестов страницы регистрации;

➔ test\_registr\_negative — сценарии с негативными сценариями регистрации.