**I. Задание 1, форма авторизации.**

**1) Расписать тест-кейсы.**

|  |  |
| --- | --- |
| Precondition: | Chrome browser Version 77.0.3865.120 (Official Build) (64-bit) |
| Шаги воспроизведения: | Ожидаемый результат: |
| Перейти по адресу:  https://calm-river-86040.herokuapp.com/ | Откроется форма регистрации представленная на картинке: |
| На форме регистрации кликнуть чек-бокс «Remember me» | Чек-бокс меняет состояние. |
| Ввести тестовые данные в поля: «Phone» - 066 0123456 «Email» - [a@a.aa](mailto:a@a.aa) «Password» - Q01234567  Кликнуть кнопку «Submit» | Отобразится всплывающее окно «Данные успешно отправлены»: |
| Во всплывающем окне кликнуть кнопку «ОК» | Всплывающее окно «Данные успешно отправлены»закроется. |

**2) Протестировать форму авторизации:**  
 а) позитивное тестирование — ввести все «валидные» данные, кликнуть «Submit» → откроется всплывающее окно «Данные успешно отправлены» → кликнуть «ОК» → проверить, что данные отправлены на сервер.

Пример «валидных» данных:

Для поля «Phone» — (066)123-4567, (092)123-4567, (098)123-4567.

Для поля «Email» — a@a.aa, some.name@gmail.com, name@i.ua.

Для поля «Password» — Q01234567, qw23eRty&0, vi$$erR0

б) негативное тестирование — ввсети не «валидные» данные и кликнуть «Submit» → поля с не валидными данными будут подсвечены и выведены сообщения об ошибках пользователю, всплывающее окно «Данные успешно отправлены» не будет показано, данные не отправятся на сервер.

Пример «не валидных» данных:

Для поля «Phone» — (»»»)»»»-»»»», (01@)$$$-$$$$, (350)123-4567.

Для поля «Email» — a@aaa, @a.aa, a@a.

Для поля «Password» — 1, a.

Также не валидными значениями будут пустые поля или поля состоящие из пробелов/переводов на новую строку для каждого из вариантов (IsNullOrEmpty or IsNullOrWhiteSpace Exceptions).

**3) Составить 2-3 баг-репорта:**

а) Обязательные к заполенению поля на форме регистрации не отмечены**.**

Перейти по адресу (открыть форму регистрации) https://calm-river-86040.herokuapp.com/:  
Ожидаемый результат: Поля обязательные к заполнению отмечены символом «\*» (к примеру).

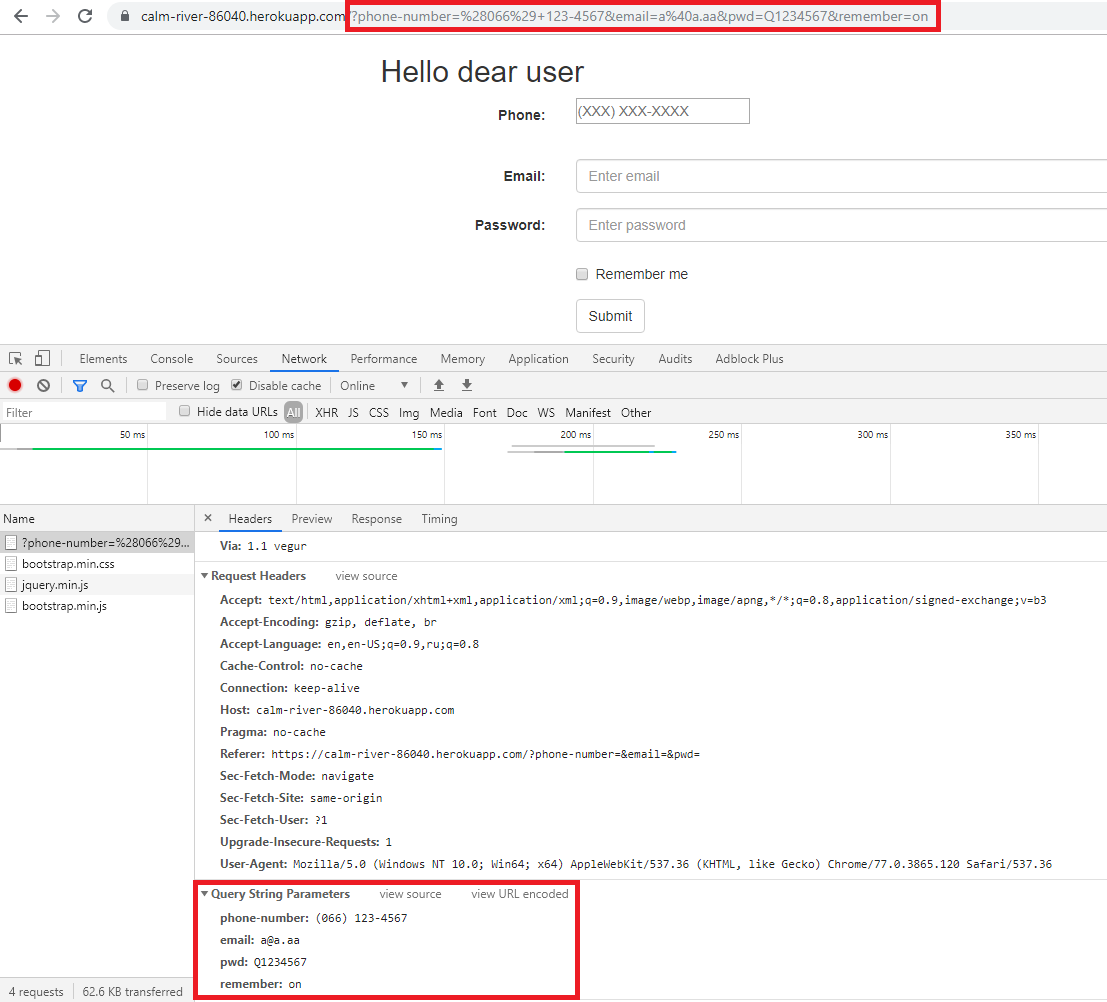
Фактический результат: Поля обазательные к заполнению никак не отмечены.

б) Отсутствует проверка валидации полей обязательных к заполнению на форме регистрации

Перейти по адресу (открыть форму регистрации) https://calm-river-86040.herokuapp.com/: → не заполнив ни одно поле кликнуть «Submit»:  
Ожидаемый результат: В полях обязательных к заполнению будет выведено сообщение об ошибке, оно будет подсвечено.

Фактический результат: Будет открыто всплывающее окно «Данные успешно отправлены».

**4) Сделать скрин успешно отправленных данных из Dev-Tools**

****

**II. Задание 2, SQL.**

**1) Выбрать все данные где цена книги равняется 15000**

select \* from BOOKS where BOOK\_PRICE = 15000;

**2) Выбрать все данные где цена книги не равняется 30000**

select \* from BOOKS where BOOK\_PRICE != 30000;

**3) Выбрать название книги Автора Lutor**

select \* from BOOKS where BOOK\_AUTHOR = ‘Lutor’;

**4) Отсортировать по цене книги по убыванию**

select \* from BOOKS order by BOOK\_PRICE desc;

**5) Отсортировать по цене книги по возрастанию**

select \* from BOOKS order by BOOK\_PRICE asc;

**III. Задание 3, SQL.**

**1) Необходимо выбрать все результаты автора Alex, разработчик составил запрос-**

**select \* from BOOKS where BOOK\_AUTHOR like “Alex%”’**

**Найти ошибку и составить баг репорт.**

select \* from BOOKS where BOOK\_AUTHOR = ‘Alex’;

а) В запросе допущена ошибка, использованы двойные кавычки вместо одинарных.

б) В условии задания не указано выводить авторов начинающихся с «Alex».

Предоставленный запрос выведет результаты не соответствующие условиям задания после выполнения.

Выполнить запрос: **select \* from BOOKS where BOOK\_AUTHOR like “Alex%”’**

Ожидаемый результат: Выбраны все результаты автора «Alex».

Фактический результат: Ошибка выполнения запроса (выбраны все результаты авторов «Alex» и «Alexandr»).

P.S. Не предоставлена полная информация о базе данных. Возможно в поиске книг по цене придется использовать одинарные кавычки, к примеру:  
**select \* from BOOKS where BOOK\_PRICE = ‘15000’;**

**IV. Задание 4, протестировать API.**

**1) Выписать все названия книг**

а) "id": 1, "name": "Harry Pooter",

б) "id": 2, "name": "Lord of the ring",

в) "id": 3, "name": "Figth Club",

г) "id": 4, "name": "CHERRY",

д) "id": 5, "name": "KATERINA",

е) "id": 6, "name": "FRENCH EXIT",

ё) "id": 7, "name": "HOW TO BE FAMOUS",

ж) "id": 8, "name": "A TERRIBLE COUNTRY",

з) "id": 9, "name": "BAD BLOOD: SECRETS AND LIES IN A SILICON VALLEY STARTUP",

|  |  |
| --- | --- |
| Get запрос из POSTMAN:  https://calm-river-86040.herokuapp.com/bookStore/0/  Ответ от API: {  "id": 1,  "name": "Harry Pooter",  "desc": "Story about boy who knows a magic",  "author": "Rolling",  "price": 15  } |  |

**2) Купить 2 книги.**

|  |  |
| --- | --- |
| Get запрос из POSTMAN:  https://calm-river-86040.herokuapp.com/bookStore/0/2/  Ответ от API:  For two books yout cost 26$  Enjoy Reading :) |  |

**3) Записать название сумму любимой книги с использованием промокодов.**

|  |  |
| --- | --- |
| Post запрос из POSSTMAN:  https://calm-river-86040.herokuapp.com/bookStore/1/userPromo/ByBookWithPromo2018  Ответ от API:  You paid 13$. |  |

|  |  |
| --- | --- |
| Post запрос из POSSTMAN:  https://calm-river-86040.herokuapp.com/bookStore/1/userPromo/ByBookWithPromo2017  Ответ от API:  You paid 18$. |  |

**4) Написать текущий баланс пользователя.**

|  |  |
| --- | --- |
| Get запрос из POSTMAN:https://calm-river-86040.herokuapp.com/bookStore/6/7/  Ответ от API:  You have no money for this  You have 70$  Books cost 89$ |  |