

Портфолио

Емельянов Владимир Петрович

Тестировщик ПО

Город: Санкт-Петербург

тел. +7(918)485-4261

emelyanovvp@mail.ru



Содержание

- Смоки тестирование сайта demoblaze.com по тест-кейсам.
- Тестирование API сайта demoblaze.com по чек листам с использованием Postman.
- Тестирование мобильных версий сайта demoblaze.com по чек листам с использованием Chrome DevTools.
- Авто тестирование сайта demoblaze.com с использованием Selenium, Pytest, Pycharm и размещением проекта на GitHub.
- Баг-репорты по результатам проведенных видов тестирования сайта demoblaze.com.
- Тестирование базы данных “world” с использованием MySQL Workbench.
- Применение техники попарного тестирования с использованием Pairwise Tool.
- Тестирование интерфейса, производительности, куки с применением Chrome DevTools.

Смоки тестирование
сайта demoblaze.com по тест-кейсам

ID	Заголовок	Связанное требование	предварительное условие	шаги	тестовые данные	ожидаемый результат	статус	заметки
TC-1	проверить функцию добавления товара в корзину		войти на страницу "Nexus 6" сайта demoblaze https://www.demoblaze.com/prod.html?idp_=3	1.нажать кнопку "Add to cart"		появится модальное окно с сообщением "Product added"	passed	

ID	Заголовок	Связанное требование	предварительное условие	шаги	тестовые данные	ожидаемый результат	статус	заметки
TC-2	проверить функцию регистрации на сайте с валидным именем		войти на главную страницу сайта demoblaze https://www.demoblaze.com/index.html	1.нажать кнопку "Sign up"			passed	
				2.в появившемся модальном окне ввести имя пользователя в поле "user name"	Jeff Sample Dull			
				3.ввести пароль в поле "Password"	12345	"Sign up successful"		
				4.нажать кнопку "Sign up"				

ID	Заголовок	Связанное требование	предварительное условие	шаги	тестовые данные	ожидаемый результат	статус	заметки
TC-3	проверить вход в личный кабинет с невалидным паролем			1.кликнуть по ссылке "Log in" на верхней панели главной страницы			passed	
			войти на главную страницу сайта demoblaze https://www.demoblaze.com/index.html	2.в появившемся окне ввести зарегистрированное имя в поле "User name"	Jeff Sample Dull	появится модальное окно с сообщением "Wrong passport"		
				3.ввести невалидный пароль в поле "Password"	12346			
				4.нажать кнопку "Log in"				

ID	Заголовок	Связанное требование	предварительно е условие	шаги	тестовые данные	ожидаемый результат	статус	заметки
ТС-4	проверить функцию отправки сообщения с невалидным имейл		войти на главную страницу сайта demoblaze https://www.demoblaze.com/index.html	1.нажать на кнопку "Contact" в хедере		появится модальное окно с сообщением "Wrong email"	failed	BR-1 (форма отправки сообщения позволяет вводить невалидный имейл)
		2.в pop-up ввести невалидный имейл в поле "Contact Email"		plainaddress				
		3. ввести имя в поле "Contact Name"		Jeff Sample Dull				
		4.ввести сообщение в поле "Message"		Lorem ipsum dolor sit amet,				
		5.нажать кнопку "Send message"						

Тестирование API сайта demoblaze.com по чек-листам с использованием Postman

ID	Заголовок	Статус	Заметки
CL-1	проверить API регистрации нового пользователя vlad16	passed	username: vlad16, password: 12345
CL-2	проверить API получения данных по vlad16	failed	BR-2 (ошибка 405)

Ниже приведены скриншоты тестирования API регистрации и входа в личный кабинет нового пользователя. Ошибка 405 означает, что Сервер знает о запрашиваемом методе, но он был деактивирован и не может быть использован. Помним, что это демонстрационный сайт для авто тестирования и не предназначен для тестирования API. Два обязательных метода, GET и HEAD, никогда не должны быть деактивированы и не должны возвращать этот код ошибки. Проверка через вкладку Tests показывает, что данный пользователь уже существует.

The screenshot displays a REST client interface for a POST request to `https://api.demoblaze.com/signup`. The request body is a JSON object:

```
1 {"username": "vlad16",  
2  "password": "12345"  
3 }
```

The response status is `200 OK` with a response time of `530 ms` and a body size of `192 B`. The response body is currently empty, showing only a single quote character: `1 "'`.

Two red arrows point from the bottom of the image to the JSON body in the request and the `200 OK` status in the response.

Регистрация нового пользователя с использованием команды POST.

The screenshot shows a web browser's developer tools interface. At the top, the address bar displays the URL: `https://api.demoblaze.com/login?username=vlad16&password=12345`. Below the address bar, the request method is set to `GET`. The `Params` tab is active, showing a table of request parameters:

Key	Value	Description
<input checked="" type="checkbox"/> username	vlad16	
<input checked="" type="checkbox"/> password	12345	

Below the parameters, the `Body` tab is active, showing the response status: `405 Method Not Allowed` with a response time of `360 ms` and a size of `404 B`. The response is displayed in `HTML` format:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
2 <title>405 Method Not Allowed</title>
3 <h1>Method Not Allowed</h1>
4 <p>The method is not allowed for the requested URL.</p>
```

Two red arrows are drawn on the image: one points from the `password` value in the parameters table to the `405 Method Not Allowed` status, and the other points from the status to the `<h1>Method Not Allowed</h1>` line in the HTML response.

Попытка получения данных по зарегистрированному пользователю с использованием команды GET.

Через повторную регистрацию убедимся, что пользователь “vlad16”, уже зарегистрирован. Второй тест, созданный на отсутствие сообщения в запросе “errorMessage”, не пройден, т.к. появилось сообщение “This user already exist”.

The screenshot shows a REST client interface for a POST request to `https://api.demoblaze.com/signup`. The interface includes tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests (selected), Settings, and Cookies. The Tests tab contains two test cases:

```
2 pm.test("Response status code is 200", function () {
3   pm.expect(pm.response.code).to.equal(200);
4 });
5
6
7 pm.test("errorMessage should be an empty string", function () {
8   const responseData = pm.response.json();
9
10  pm.expect(responseData.errorMessage).to.equal("");

```

Below the code editor, the Test Results section shows the following results:

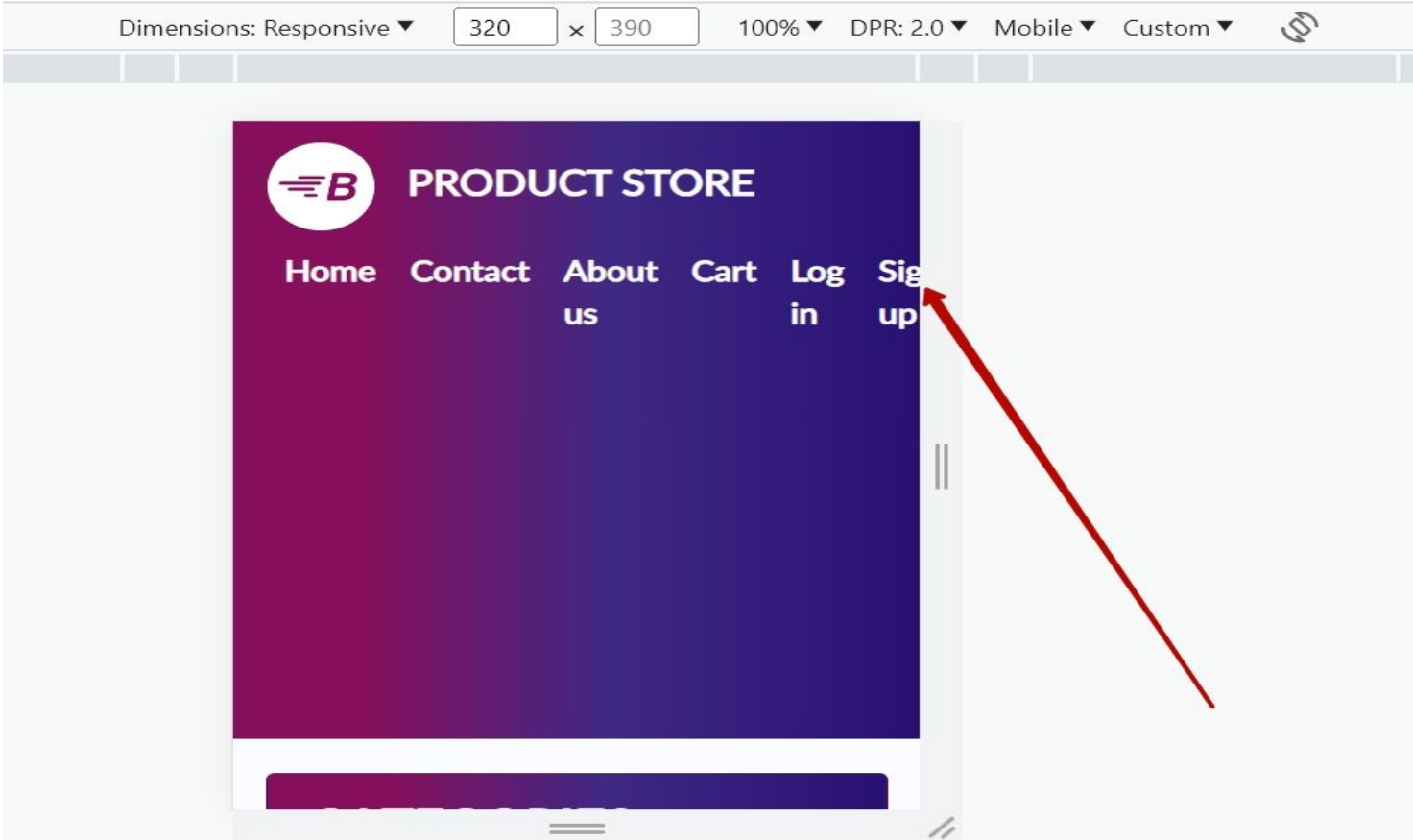
- PASS** Response status code is 200
- FAIL** errorMessage should be an empty string | AssertionError: expected 'This user already exist.' to equal ''

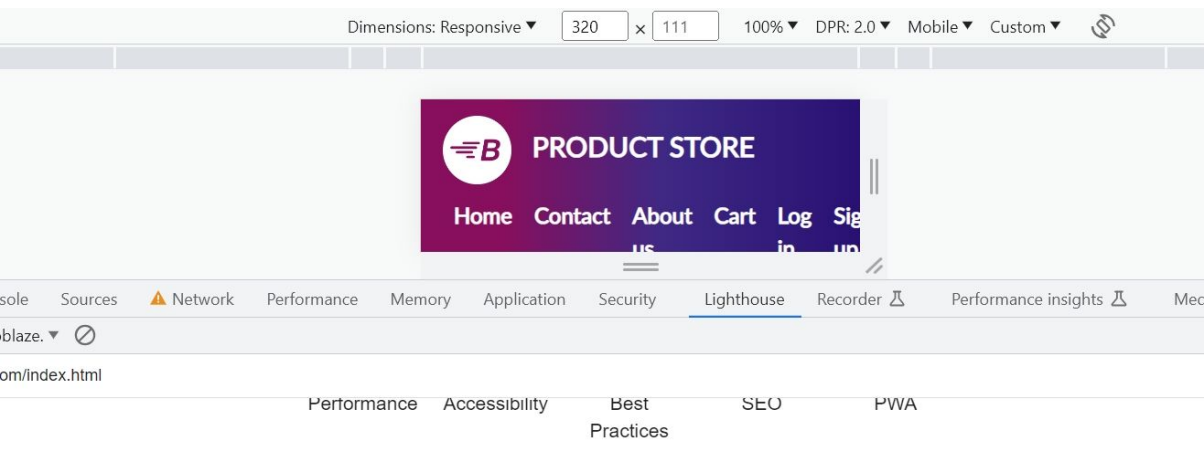
Two red arrows point from the failure message in the Test Results section to the corresponding test code in the Tests tab.

Тестирование мобильных версий
сайта demoblaze.com по чек-листам с использованием
Chrome DevTools

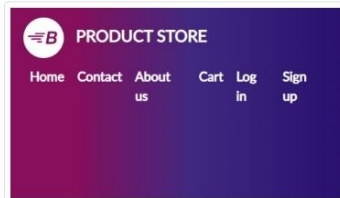
ID	Заголовок	Статус	Заметки
CL-3	проверить масштабируемость приложения для мобильных устройств с разными размерами экрана	failed	BR-3 (для экрана 320 X 390 элемент хедера "Sign Up" не поместился на видимой части экрана)
CL-4	проверить скорость загрузки приложения для Samsung Galaxy A13 в условиях 3G	failed	BR-4 (время полной загрузки интерактивного контента (TBT), время загрузки видимой части страницы до скроллинга (LCP), а также уровень "прыжков" контента во время загрузки (CLS) оцениваются счетчиком Google Lighthouse как неудовлетворительные)

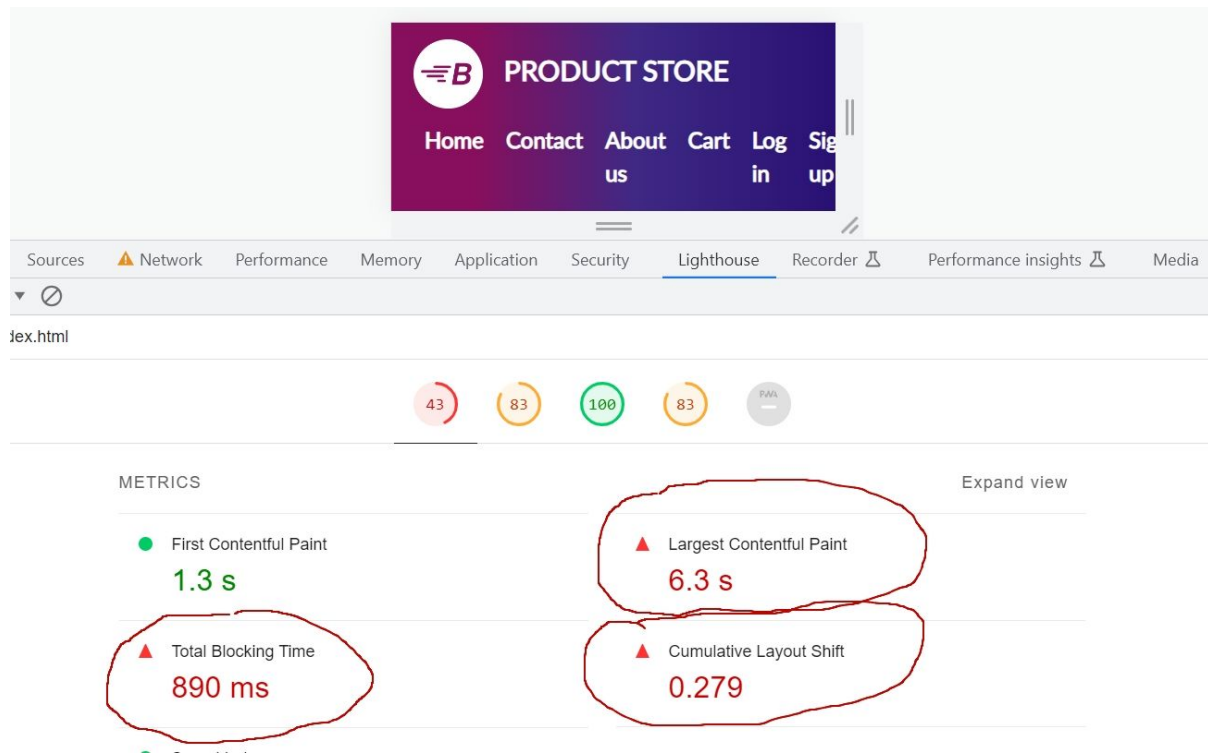
Здесь приведен скриншот тестирования масштабируемости приложения для мобильных устройств размером экрана 320 X 390, где видно, что кнопка Sign up не “влезла” в экран.





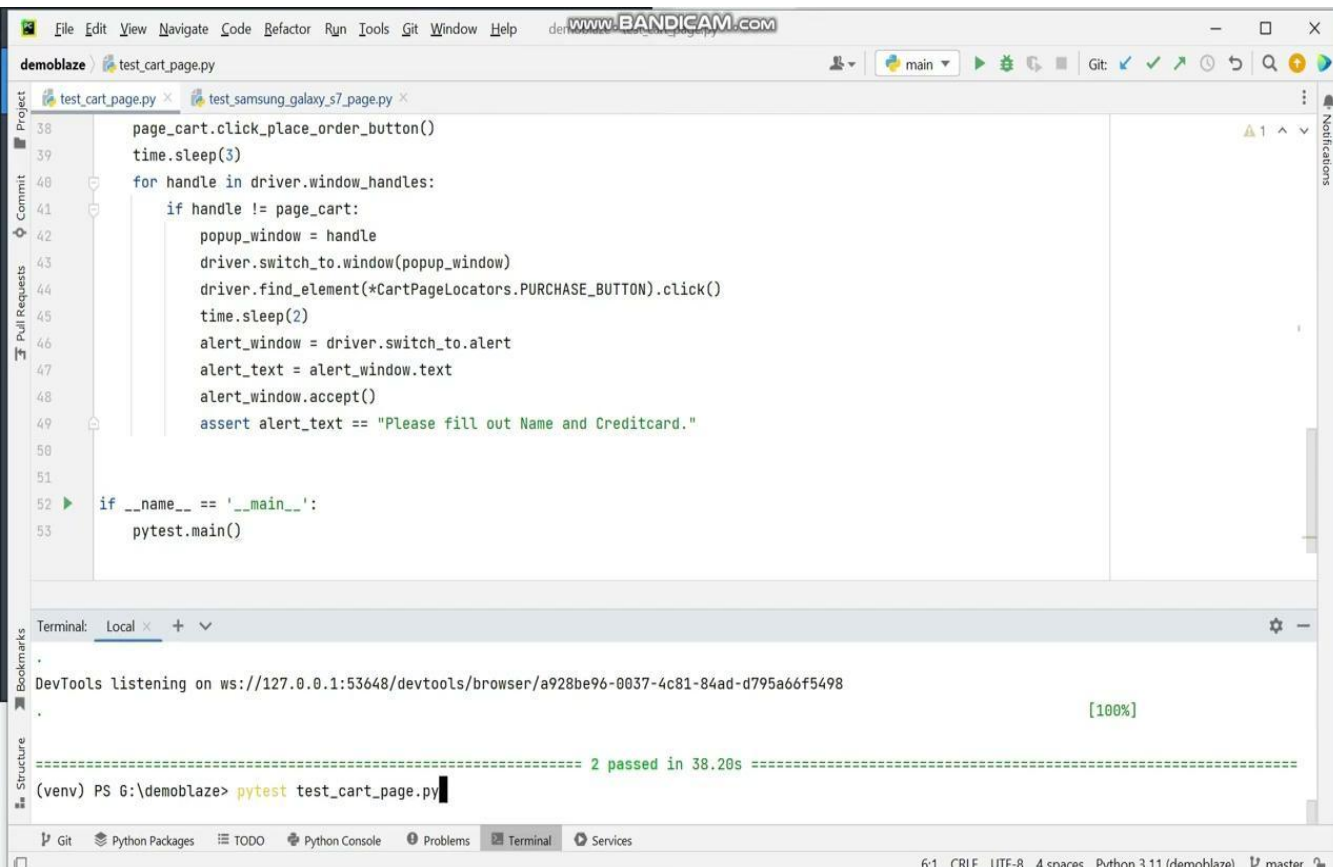
Здесь мы видим отрицательный результат тестирования скорости загрузки в условиях слабого сигнала (3G).





Три из четырех показателей производительности при слабом сигнале находятся в отрицательной зоне.

Автотестирование сайта demoblaze.com с использованием Selenium, Pytest, Pycharm и размещением проекта на GitHub.



```
File Edit View Navigate Code Refactor Run Tools Git Window Help demoblaze.com  
demoblaze test_cart_page.py  
test_cart_page.py x test_samsung_galaxy_s7_page.py x  
38 page_cart.click_place_order_button()  
39 time.sleep(3)  
40 for handle in driver.window_handles:  
41     if handle != page_cart:  
42         popup_window = handle  
43         driver.switch_to.window(popup_window)  
44         driver.find_element(*CartPageLocators.PURCHASE_BUTTON).click()  
45         time.sleep(2)  
46         alert_window = driver.switch_to.alert  
47         alert_text = alert_window.text  
48         alert_window.accept()  
49         assert alert_text == "Please fill out Name and Creditcard."  
50  
51  
52 if __name__ == '__main__':  
53     pytest.main()  
  
Terminal: Local x + v  
DevTools listening on ws://127.0.0.1:53648/devtools/browser/a928be96-0037-4c81-84ad-d795a66f5498  
[100%]  
===== 2 passed in 38.20s =====  
(venv) PS G:\demoblaze> pytest test_cart_page.py
```

При написании кода использована POM (page oriented model) модель тестирования телефона Samsung Galaxy S7. Здесь приведен видеоролик проверки работы кнопки Покупка: с заполнением формы заказа и без заполнения формы заказа. Оба автотеста пройдены.

The screenshot shows an IDE window with a Python test file named `test_samsung_galaxy_s7_page.py`. The code defines a test function `test_add_samsung_galaxy_s7_to_cart` that performs several actions: opening a page, adding an item to the cart, waiting for an alert, and asserting the alert text. The terminal output shows the test environment setup, including plugins and the DevTools listening address.

```
1 import time
2 import pytest
3 from .samsung_galaxy_s7_page import SamsungGalaxyS7Page
4
5
6 def test_add_samsung_galaxy_s7_to_cart(driver):
7     url = "https://www.demoblaze.com/prod.html?idp_#4#"
8     page = SamsungGalaxyS7Page(driver, url, timeout=2)
9     page.open()
10    page.add_samsung_galaxy_s7_to_cart()
11    time.sleep(2)
12    alert = driver.switch_to.alert
13    text = alert.text
14    alert.accept()
15    assert text == "Product added", "Product wasn't added"
16
17
```

Terminal output:

```
plugins: print-0.3.1
collected 2 items

test_samsung_galaxy_s7_page.py
DevTools listening on ws://127.0.0.1:56718/devtools/browser/2bcd594-878e-4d0b-979d-8c4e32724738
```

Во втором ролике также два теста: проверка добавления товара в Корзину и проверка перехода на страницу Корзина. Оба теста успешно пройдены.

Это скриншот страницы с автокодом тестирования сайта demoblaze.com на GitHub. По этой ссылке можно посмотреть полный автотест: <https://github.com/emelyanovvp/demoblaze>

emelyanovvp / demoblaze

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

demoblaze Public

Pin Unwatch 1 Fork 0 Star 0

master 1 branch 0 tags Go to file Add file Code

emelyanovvp changed 5 files 3f3623b on Apr 21 3 commits		
.idea	changed 7 files	4 months ago
__init__.py	Initial commit	4 months ago
cart_page.py	changed 7 files	4 months ago
conftest.py	Initial commit	4 months ago
home_page.py	changed 5 files	4 months ago

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Баг-репорты по результатам проведенных видов тестирования сайта demoblaze.com

ID	Summary	Description	Steps	Actual result	Expected result	Attachment	Серьезность (Severity)	Приоритет (Priority)	Status	Окружение (Environment)
BR-1	на вкладке "Contact" в форме отправки сообщения отсутствует запрет на ввод и отправку невалидных имейл		<p>1.нажать на кнопку "Contact" в хедере</p> <p>2.в pop-up ввести невалидный имейл в поле "Contact Email"</p> <p>3. ввести имя в поле "Contact Name"</p> <p>4.ввести сообщение в поле "Message"</p> <p>5.нажать кнопку "Send message"</p>	появится модальное окно с сообщением "Thanks for the message!!"	появится модальное окно с сообщением "Wrong email"	https://docs.google.com/document/d/1OJ9-IKFI_Sqns_Ch2s_BUrnOp4_Q09MJl5od-nexOla4/e/dit?usp=sharing	minor	medium	opened	win10 desktop Chrome 115.0 1920 X 1080

ID	Summary	Description	Steps	Actual result	Expected result	Attachment	Серьезность (Severity)	Приоритет (Priority)	Status	Окружение (Environment)
BR-2	проверить работу API входа в личный кабинет зарегистрированного пользователя		1.в Postman на вкладке запроса выбрать команду GET	сообщение 405 Method Not Allowed	сообщение 200 OK	https://docs.google.com/document/d/1BhdvIxUfJn29Pt1pN3U70UVIRUksH_vdR0ek5Oun5B8/edit?usp=sharing	major	high	opened	win10 desktop Chrome 115.0 1920 X 1080
			2.ввести url https://api.demoblaze.com/login							
			3.на вкладке Params ввести 'username' в поле Key							
			4.ввести 'vlad16' в поле Value							
			5.ввести 'password' в поле Key							
			6.ввести "12345" в поле Value							
			7.нажать кнопку Send							

ID	Summary	Description	Steps	Actual result	Expected result	Attachment	Серьезность (Severity)	Приоритет (Priority)	Status	Окружение (Environment)
BR-3	<p>проверить масштабируемость приложения для мобильных устройств с разными размерами экрана</p>		1.на главной странице demoblaze.com открыть Chrome DevTools	<p>элемент хедера "Sign Up" не помещился на видимой части экрана</p>	<p>элемент хедера "Sign Up" виден полностью</p>	<p>https://docs.google.com/document/d/1QTwaH-8yIMsdAOiRI-FEiOxwOqM9aMZQBgL1d0h-Q8/edit?usp=sharing</p>	minor	normal	opened	<p>win10 desktop Chrome 115.0 1920 X 1080</p>
			2.в drop-box Dimensions выбрать Responsive							
			3.вверху в ленте для выбора разных экранов выбрать Mobile S							

ID	Summary	Description	Steps	Actual result	Expected result	Attachment	Серьезность (Severity)	Приоритет (Priority)	Status	Окружение (Environment)
BR-4	проверить скорость загрузки приложения для Samsung Galaxy A13 в условиях 3G		<p>1.на главной странице demoblaze.com открыть Chrome DevTools</p> <p>2.в drop-box Dimensions выбрать Samsung Galaxy A13</p> <p>3.выбрать вкладку Lighthouse и нажать кнопку Analyze Page Load</p>	показатели и полной загрузки оцениваются счетчиком Google Lighthouse как неудовлетворительные	показатели удовлетворительные	https://docs.google.com/document/d/1BxhgxRrvwxAxpJ-EpQc_aRUNx6S_HAKytGFJ8fAlt2ets/edit?usp=sharing	minor	normal	opened	win10 desktop Chrome 115.0 1920 X 1080

Тестирование базы данных “world” с использованием MySQL Workbench

The screenshot displays the MySQL Workbench interface. At the top, the SQL editor contains the query: `select * from world.country where Continent = 'Europe';`. Two red arrows point to the asterisk in the `select *` clause and the single quote in the `'Europe'` value. Below the editor is the 'Result Grid' showing a table of 46 rows. The table has columns: Code, Name, Continent, Region, SurfaceArea, IndepYear, Population, LifeExpectancy, GNP, GNPOld, and LocalName. The first few rows are: Albania, Andorra, Austria, Belgium, Bulgaria, Bosnia and Herzegovina, Belarus, Switzerland, Czech Republic, Germany, Denmark, Spain, and Estonia. At the bottom, the 'Output' pane shows the execution log with a message: `select * from world.country where Continent = 'Europe' LIMIT 0, 1000` and a status: `46 row(s) returned`. A red arrow points to the `46 row(s) returned` text.

Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName
ALB	Albania	Europe	Southern Europe	28748.00	1912	3401200	71.6	3205.00	2500.00	Shqipëria
AND	Andorra	Europe	Southern Europe	468.00	1278	78000	83.5	1630.00	NULL	Andorra
AUT	Austria	Europe	Western Europe	83859.00	1918	8091800	77.7	211860.00	206025.00	Österreich
BEL	Belgium	Europe	Western Europe	30518.00	1830	10239000	77.8	249704.00	243948.00	België/Belgique
BGR	Bulgaria	Europe	Eastern Europe	110994.00	1908	8190900	70.9	12178.00	10169.00	Balgarija
BIH	Bosnia and Herzegovina	Europe	Southern Europe	51197.00	1992	3972000	71.5	2841.00	NULL	Bosna i Hercegovina
BLR	Belarus	Europe	Eastern Europe	207600.00	1991	10236000	68.0	13714.00	NULL	Belarus
CHE	Switzerland	Europe	Western Europe	41284.00	1499	7160400	79.6	264478.00	256092.00	Schweiz/Suisse/Svizzera/Sviz
CZE	Czech Republic	Europe	Eastern Europe	78866.00	1993	10278100	74.5	55017.00	52037.00	Česko
DEU	Germany	Europe	Western Europe	357022.00	1955	82164700	77.4	2133367.00	2102826.00	Deutschland
DNK	Denmark	Europe	Nordic Countries	43094.00	800	5330000	76.5	174099.00	169264.00	Danmark
ESP	Spain	Europe	Southern Europe	505992.00	1492	39441700	78.8	553233.00	532031.00	España
EST	Estonia	Europe	Baltic Countries	45777.00	1991	1430700	69.5	5738.00	3371.00	Eesti

На примере демо базы данных “world”, представленной на MySQL Workbench, проведем проверку таблицы Country для всех стран, находящихся в Европе. Здесь мы можем проверить столбцы на соответствующее наполнение. Получили 46 стран.

The screenshot shows a database query tool interface. At the top, a toolbar contains various icons for file operations and editing. Below the toolbar, a SQL query is entered in a text area: `1 • select * from world.city where Population >= 1000000;`. A red arrow points from the text below to the `Population >= 1000000` part of the query. Below the query editor, a "Result Grid" section displays a table of results. The table has columns for ID, Name, CountryCode, District, and Population. Below the table, there is a tab labeled "city 2" and an "Output" section. The "Output" section shows a table with columns for #, Time, Action, and Message. A red arrow points from the text below to the "238 row(s) returned" message in the Output table.

```
1 • select * from world.city where Population >= 1000000;
```

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
35	Alger	DZA	Alger	2168000
56	Luanda	AGO	Luanda	2022000
69	Buenos Aires	ARG	Distrito Federal	2982146
70	La Matanza	ARG	Buenos Aires	1266461
71	Córdoba	ARG	Córdoba	1157507
126	Yerevan	ARM	Yerevan	1248700
130	Sydney	AUS	New South Wales	3276207
131	Melbourne	AUS	Victoria	2865329
132	Brisbane	AUS	Queensland	1291117

#	Time	Action	Message
1	10:40:53	select * from world.city where Population >= 1000000 LIMIT 0, 1000	238 row(s) returned

По таблице City мы запросили все города с населением больше миллиона. Получили 238 городов.

```
1 • select country.Name, city.Name, city.Population
2   from country
3   inner join city
4   on country.Code = city.CountryCode
5   where country.Continent = 'Europe' and city.Population >= 1000000;
6
```

Result Grid

Name	Name	Population
Austria	Wien	1608144
Bulgaria	Sofija	1122302
Belarus	Minsk	1674000
Czech Republic	Praha	1181126
Germany	Berlin	3386667
Germany	Hamburg	1704735
Germany	Munich [München]	1194560
Spain	Madrid	2879052
Spain	Barcelona	1503451
France	Paris	2125246
United Kingdom	London	7285000
United Kingdom	Birmingham	1013000
Hungary	Budapest	1811552
Italy	Roma	2643581

Result 10 ×

Output

Action Output

#	Time	Action	Message
1	11:07:11	select country.Name, city.Name, city.Population from country inner join city on co...	36 row(s) returned

В третьем запросе мы объединили данные по названиям страны и города с условием, что город миллионник и страна в Европе. Для этого воспользовались командами “inner join”, “where” и Foreign Keys (Code-CountryCode), чтобы “вытащить” нужные данные из двух разных таблиц. Получили 36 строк, которые легче тестировать на соответствие страна-город, правописание и др.


Применение техники попарного тестирования с использованием Pairwise Tool

Часто при тестировании блока регистрации на сайтах необходимо проверять четыре поля для ввода данных (имя, фамилия, пароль, пароль). Введем данные для этих полей в таблицу Pairwise Tool по трем значениям (пустой, невалидный, валидный).

 pairwiseTool

 Edit Conditions

 Generate Pairwise

 Generate All Combinations

 Create



	× first name	× last name	× password 1	× password 2
× Row 1	blank	blank	blank	blank
× Row 2	invalid	invalid	invalid	invalid
× Row 3	valid	valid	valid	valid

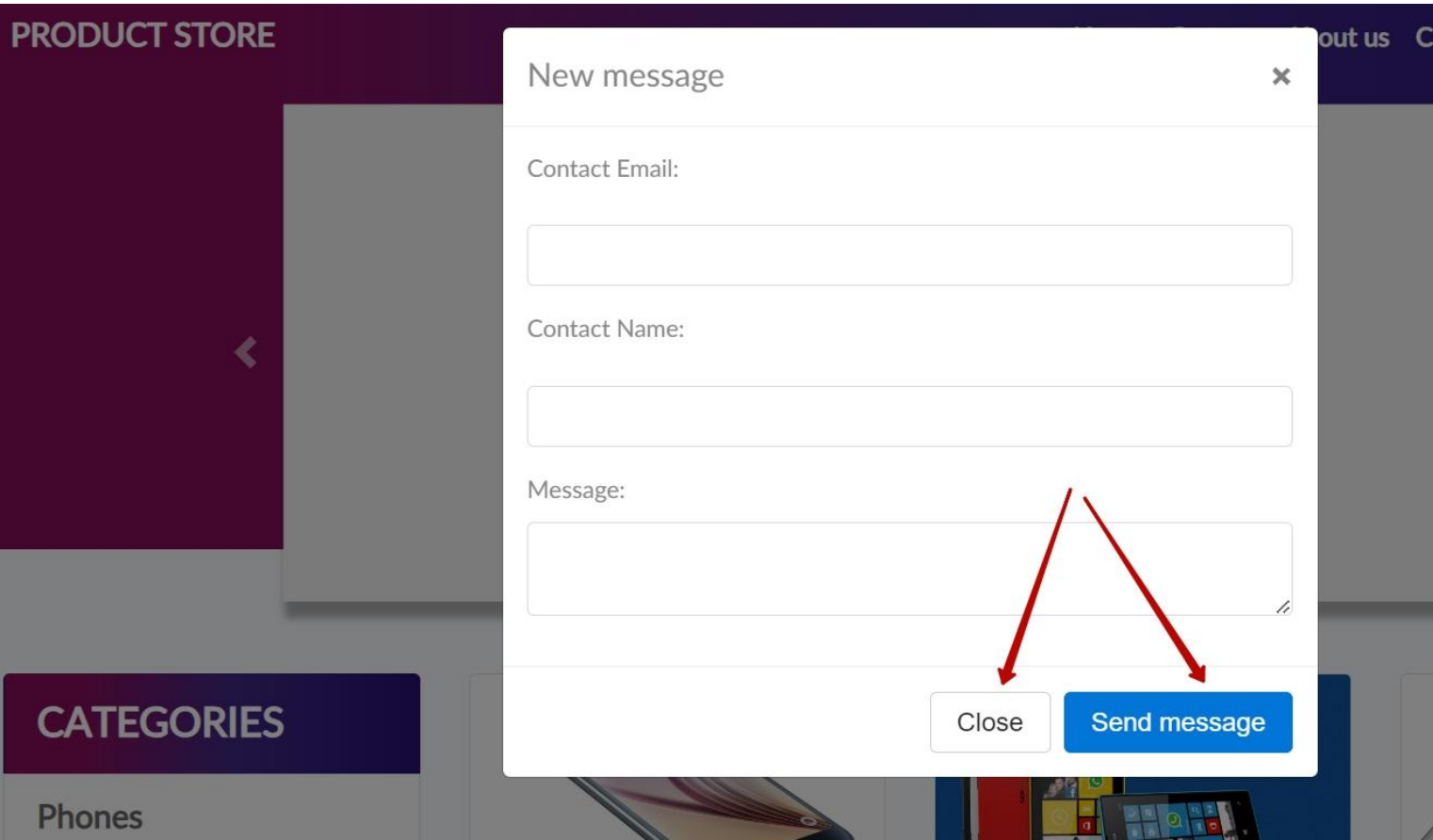
Чтобы проверить все комбинации для 100% покрытия тестами блока регистрации без использования техники попарного тестирования надо провести 81 проверку.

	first name	last name	password 1	password 2
67	valid	invalid	invalid	blank
68	valid	invalid	invalid	invalid
69	valid	invalid	invalid	valid
70	valid	invalid	valid	blank
71	valid	invalid	valid	invalid
72	valid	invalid	valid	valid
73	valid	valid	blank	blank
74	valid	valid	blank	invalid
75	valid	valid	blank	valid
76	valid	valid	invalid	blank
77	valid	valid	invalid	invalid
78	valid	valid	invalid	valid
79	valid	valid	valid	blank
80	valid	valid	valid	invalid
81	valid	valid	valid	valid

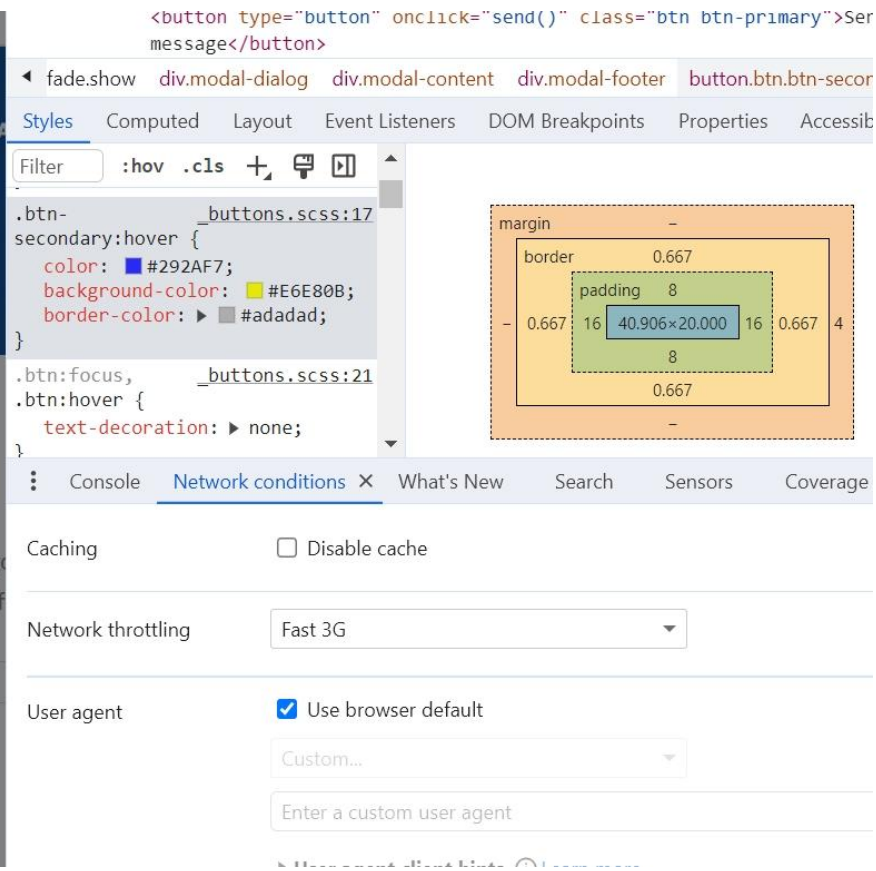
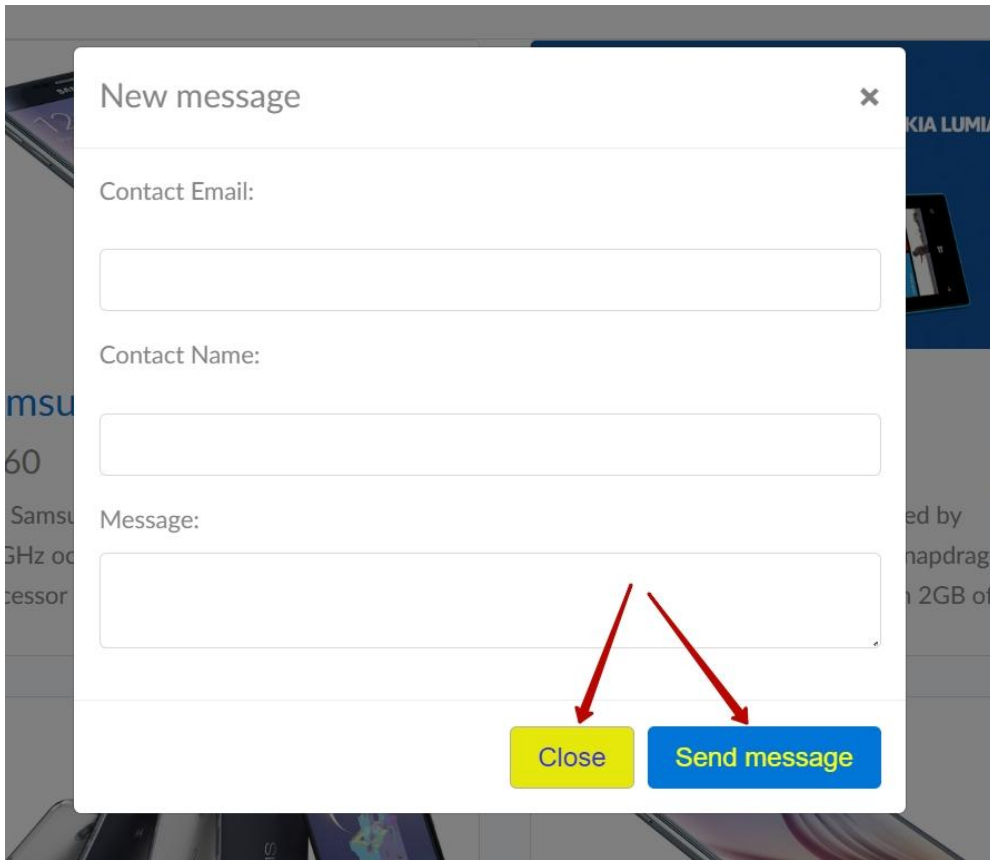
В то время, как применяя технику попарного тестирования и используя Pairwise Tool, получим 9 тестов, что сократит количество проверок в девять раз при том же 100% покрытии тестов.

	first name	last name	password 1	password 2
1	blank	blank	blank	blank
2	blank	invalid	invalid	invalid
3	blank	valid	valid	valid
4	invalid	invalid	valid	blank
5	invalid	valid	blank	invalid
6	invalid	blank	invalid	valid
7	valid	valid	invalid	blank
8	valid	blank	valid	invalid
9	valid	invalid	blank	valid

Тестирование интерфейса, производительности, куки с применением Chrome DevTools.



При тестировании интерфейса удобно использовать Chrome DevTools для проверки соответствия цветов, шрифтов, размеров и др.



Используя вкладку Styles предложим разработчику другой цвет кнопок, сверим размеры и типы шрифтов по их кодировке на разных страницах приложения.



Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights Media

```
<html lang="en"> == $0
  <head> ... </head>
  <body> ... </body>
</html>
```

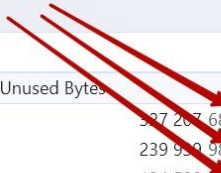
Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

filter :hov .cls

Console Network conditions Coverage

URL filter All Content scripts

RL	Type	Total Bytes	Unused Bytes	Usage Visualization
https://www.demoblaze.com/node_modules/video.js/dist/video.min.js	JS (per func...	480 293	327 267 68.1%	
rome-extension://fbnkflkahlmhdgkddaafgnnokifobik/assets/pageScripts.js	JS (per func...	243 261	239 950 98.6%	
https://www.demoblaze.com/node_modules/bootstrap/dist/css/bootstrap.min.css	CSS	150 996	134 588 89.1%	
https://www.demoblaze.com/node_modules/videojs-contrib-hls/dist/videojs-contrib-hls.min.js	JS (per func...	232 413	109 713 47.2%	
https://www.demoblaze.com/node_modules/jquery/dist/jquery.min.js	JS (per func...	86 659	43 819 50.6%	
https://www.demoblaze.com/node_modules/video.js/dist/video-js.min.css	CSS	40 024	39 602 98.9%	
https://www.demoblaze.com/node_modules/bootstrap/dist/js/bootstrap.min.js	JS (per func...	46 653	28 850 61.8%	
https://www.demoblaze.com/node_modules/tether/dist/js/tether.min.js	JS (per func...	24 989	21 347 85.4%	
https://www.demoblaze.com/css/latofonts.css	CSS	9 762	9 762 100%	



Используя вкладку Coverage, определим неиспользуемые коды. Как видно из таблицы 68% и 98% для JS, 89% для CSS скриптов не используются, это замедляет скорость загрузки страницы сайта, что является багом.



Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights Media

storage

- Local storage
- Session storage
- IndexedDB
- Web SQL
- Cookies
- Private state tokens
- Interest groups
- Shared storage

Filter

Only show cookies with an issue

Name	Value	Domain	Path	Expires ...	Size	HttpOnly	Secure	SameSite	Partitio...	Priority
tokenp_	dmxhZDU3MTY5NzI1OA==	www.d...	/	Session	27					Medium
user	22b08628-7810-58d2-c732-2c8561299fab	www.d...	/	Session	40					Medium

Используя вкладку Cookies, проверим надежность хранения личных данных в куки. Для этого зарегистрируем юзера vlad57, проверим, закодированы ли данные vlad57 в токене (да, закодированы), а затем удалим закодированные данные в столбце Value.



Elements Console Sources Network Performance Memory **Application** Security Lighthouse Recorder Performance insights Media

Storage

- Local storage
- Session storage
- IndexedDB
- Web SQL
- ▼ Cookies
 - https://www.demoblaze.com
 - Private state tokens
 - Interest groups
 - Shared storage

Filter

Name	Value	Domain	Path	Expires ...	Size	HttpOnly	Secure	SameSite	Partitio...	Priority
tokenp_		www.d...	/	Session	7					Medium
user	22b08628-7810-58d2-c732-2c8561299fab	www.d...	/	Session	40					Medium

Как видно на скриншоте, куки прошли тест: приветствие “Welcome vlad57” исчезло, появилось “Log in”, доступ к личным данным прекратился.