

ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА

по дополнительной профессиональной программе
повышения квалификации

«Тестировщик программного обеспечения»

Выполнил: Мельников
Михаил Анатольевич
Поток: ТП-600
Преподаватель: Гриненко
Виталий Владимирович

2025г

Содержание

1. Анализ веб сайта (демонстрация пяти сценариев использования)
2. Тестовая документация (чек-лист, тест-кейсы, баг-репорты);
3. Отчет о тестировании мобильного приложения
4. SQL запрос
5. Коллекция Postman и результат отправки запроса на API
6. Презентация подготовленного резюме

Анализ веб-сайта

ПР-1

Для анализа выбран интернет-магазин Ozon (<https://www.ozon.ru>)

На главной странице и в процессе навигации были выявлены следующие ключевые функциональные элементы:

кнопка Каталог, открывающая каталог товаров,

поле поиска с автодополнением,

блок кнопок в правом верхнем углу:

кнопка авторизации пользователя (для авторизованного пользователя открывает доступ в личный кабинет),

кнопка Заказы (показывает актуальные и завершенные заказы),

кнопка Избранное,

кнопка Корзина;

карточки с товаром

Подвал сайта с контактной информацией и ссылками на юридические документы

Анализ веб-сайта

ПР-1

Сценарии использования.

1. Поиск товара по названию

Пользователь вводит в строку поиска запрос. Например, «газовая колонка». Сайт отображает список соответствующих товаров, пользователь выбирает нужную модель и переходит на её страницу.

2. Фильтрация товаров в категории

Пользователь переходит в категорию "Проточные газовые колонки" и использует фильтры: цена от 10000 до 15000 руб., мощность 20 кВт, наличие дисплея - Да. Сайт обновляет список товаров соответствии с указанными параметрами.

3. Добавление товара в корзину

На странице товара пользователь сравнивает технические характеристики, цвет прибора и нажимает кнопку «Добавить в корзину».

Система отобразит уведомление об успешном добавлении.

Анализ веб-сайта

ПР-1

4. Оформление заказа

Пользователь переходит в корзину, проверяет состав заказа, нажимает «Оформить заказ», заполняет данные доставки и оплаты, подтверждает покупку.

5. Отслеживание заказа

Узнать о ходе выполнения заказа пользователь может с помощью кнопки "Заказы" в правом верхнем углу сайта.

Тестовая документация: тест-кейсы

A	B	C	D	E	F	G	H	I	J	K	L	M
D	Заголовок	Связанное требование	Предварительные условия	Шаги	Тестовые данные	OP	Статус	Создано (Created by)	Дата создания	Выполнено	Дата выполнения	Тестовое окружение
TC-01	Заказ новой карты	REQ-111 <link>	Пользователь должен находиться во вкладке "Карты" - "Бизнес-карты"	1. Нажать кнопку "Заказать карту". 2. Выбрать нужную карту. 3. Нажать кнопку "Заказать карту" в поле выбранной карты. 4. В новом окне заполнить форму заказа 4. Нажать кнопку "Заказать" 5. В окне подтверждения заказа согласиться с условиями получения карты. 6. Подтвердить заявку путем получения СМС либо с помощью электронной подписи.	см. файл USERS <link>	Система отреагирует сообщением: "Ваша заявка принята. Информация о готовности карты будет направлена в виде СМС сообщения."	PASS	Мельников М.А.	28.10.25	Мельников М.А.	28.10.25	WIN 10 Microsoft Edge 141.0.3537.92

Тестовая документация: тест-кейсы

Список тест-кейсов в Test IT

Список тест-кейсов в Test IT									
Фильтр		Создать		Сгенерировать тесты		Импорт		Без группировки	
Приорит...	Секция	ID	Название	Статус	Дата создания	Автор	Теги		
---	БСПБ	45	Заказ новой карты	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	46	Блокирование карты	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	47	Подключение СМС дл...	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	48	Загрузка списка опер...	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	49	Задать вопрос о карте	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	50	Открыть депозит	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	51	Отказ в загрузке спис...	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	52	Отказ в открытии деп...	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	53	Превышение максима...	Готов	04.11.2025	M Михаил ...	regression		
---	БСПБ	54	Превышение максима...	Готов	04.11.2025	M Михаил ...	regression		

Тестовая документация: баг-репорты

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	Заголовок (ЧТО? ГДЕ? КОГДА?)	Подробное описание	STR (Шаги)	ОР	ФР	Важность (severity)	Приоритет (priority)	Статус	Окружение	Вложения			
2	BR-1	Превышение максимальной суммы при оформлении заявки на кредит "Гарантия без обеспечения".	При оформлении заявки на кредит "Гарантия без обеспечения" есть возможность превысить максимальную сумму 5000000 рублей.	1. На вкладке "Кредиты" нажать кнопку "Оформить заявку". 2. В открывшемся окне нажать на кнопку "Оформить" в строке продукта "Гарантия без обеспечения". 3. Написать в поле ввода "Сумма" число, превышающее указанный лимит в 50 000 000 рублей. 4. Нажать кнопку "Дальше".	Система покажет предупреждение о превышении максимальной суммы.	Открывается следующее окно, оформление кредита продолжится.	S3	P2	Новый	WIN 10 Microsoft Edge 141.0.3537.92	https://disk.yandex.ru/i/ZF6oSxD7M4SZeHq			
3	BR-2	Неактивна кнопка "Открыть счет" в форме заявления на открытие счета.	При открытии счета в форме "Заявления на открытие счета" нет реакции на нажатие кнопки "Открыть счет".	1. На вкладке "Счета" - "Заявление на открытие счета" нажать на кнопку "Открыть счет".	Откроется окно с формой заявления на открытие счета.	Система не реагирует на нажатие кнопки, хотя пользователь может нажать на нее.	S3	P2	Новый	WIN 10 Microsoft Edge 141.0.3537.92	https://disk.yandex.ru/i/tp_2HWrh7E79Q			

Отчет о тестировании мобильного приложения

Отчет по тестированию ПО "БСПБ"

В период с 01.11.2025 до 07.11.2025 проведено функциональное и UI/UX тестирование с целью выявить ошибки в программном обеспечении до его выпуска. Работа приложения тщательно проверена в различных браузерах и операционных системах, что поможет улучшить эффективность вашего бизнеса.

Окружение

Тестовые стенды

Stage	
Production	idemo.bspb.ru

ОС/Браузеры

ОС	Android V14.0.1
Браузер	Chrome v135.0.7049.111
ОС	Windows 10 v22H2
Браузер	Mozilla Firefox v144.0.2
Браузер	Chrome 142.0.7444.134

QA team

Ф.И.О.	Время	Роль/Деятельность	Готовность
Мельников М.А.	5ч.	Проведение тест-кейсов	<input checked="" type="checkbox"/>
Мельников М.А.	2ч.	Составление баг-репортов	<input checked="" type="checkbox"/>
Мельников М.А.	3ч.	Составление чек-листа	<input checked="" type="checkbox"/>
Мельников М.А.	4ч.	Составление отчета по тестированию моб. приложения	<input checked="" type="checkbox"/>

Отчет о тестировании мобильного приложения

Общая информация по тестированию		
Инструменты		
Xmind	Использован для построения схемы приложения.	
TestIT	Использован для составления тест-кейсов	
Описание процесса тестирования		
Проверка	Статус	Описание процесса тестирования
Приложение переключается между портретным и ландшафтным режимами.	<input checked="" type="checkbox"/>	При повороте устройства происходит переключение между портретным и ландшафтным режимами.
Соединение зашифровано по протоколу https	<input checked="" type="checkbox"/>	Соединение зашифровано
Есть ли у ПО отдельное мобильное приложение	<input type="checkbox"/>	Моб. приложения нет
Есть ли адаптация под различные размеры экранов при открытии приложения на моб. устройствах	<input type="checkbox"/>	Адаптации нет
Обязательные поля отмечены спец. символом	<input type="checkbox"/>	Обязательные поля не отмечены спец. символом
При попытке отправить форму без заполненных обязательных полей сработает валидация	<input checked="" type="checkbox"/>	Система покажет окно с предупреждением

Отчет о тестировании мобильного приложения

При попытке ввода невалидных данных сработает валидация	<input checked="" type="checkbox"/>	Система покажет окно с предупреждением
При отправке формы с пустыми полями сработает валидация	<input checked="" type="checkbox"/>	Система покажет окно с предупреждением
Проверка допустимой длины (макс, мин)	<input checked="" type="checkbox"/>	Система покажет окно с предупреждением
Допустимы ли цифры в текстовом поле	<input type="checkbox"/>	При вводе цифр в текстовое поле система принимает такой вариант
Допустимы ли пробелы в начале или в конце строки	<input type="checkbox"/>	Система принимает такой вариант
Возможно ли копирование и вставка	<input checked="" type="checkbox"/>	Да, возможна
Возможно ли ввести дробное число	<input checked="" type="checkbox"/>	Да, возможно
При вводе значения, выходящего за пределы установленных лимитов, сработает валидация	<input checked="" type="checkbox"/>	Система покажет окно с предупреждением
При вводе отрицательного числа сработает валидация	<input checked="" type="checkbox"/>	Система покажет окно с предупреждением
Важен ли регистр при вводе email	<input checked="" type="checkbox"/>	При вводе имейла регистр не учитывается
Проверка валидности email	<input checked="" type="checkbox"/>	При попытке ввести невалидный имейл срабатывает валидация
Важен ли регистр при вводе пароля	<input checked="" type="checkbox"/>	При вводе пароля регистр учитывается
Важна ли сложность пароля	<input checked="" type="checkbox"/>	Да, важна. Требования к пароля описаны в настройках сайта

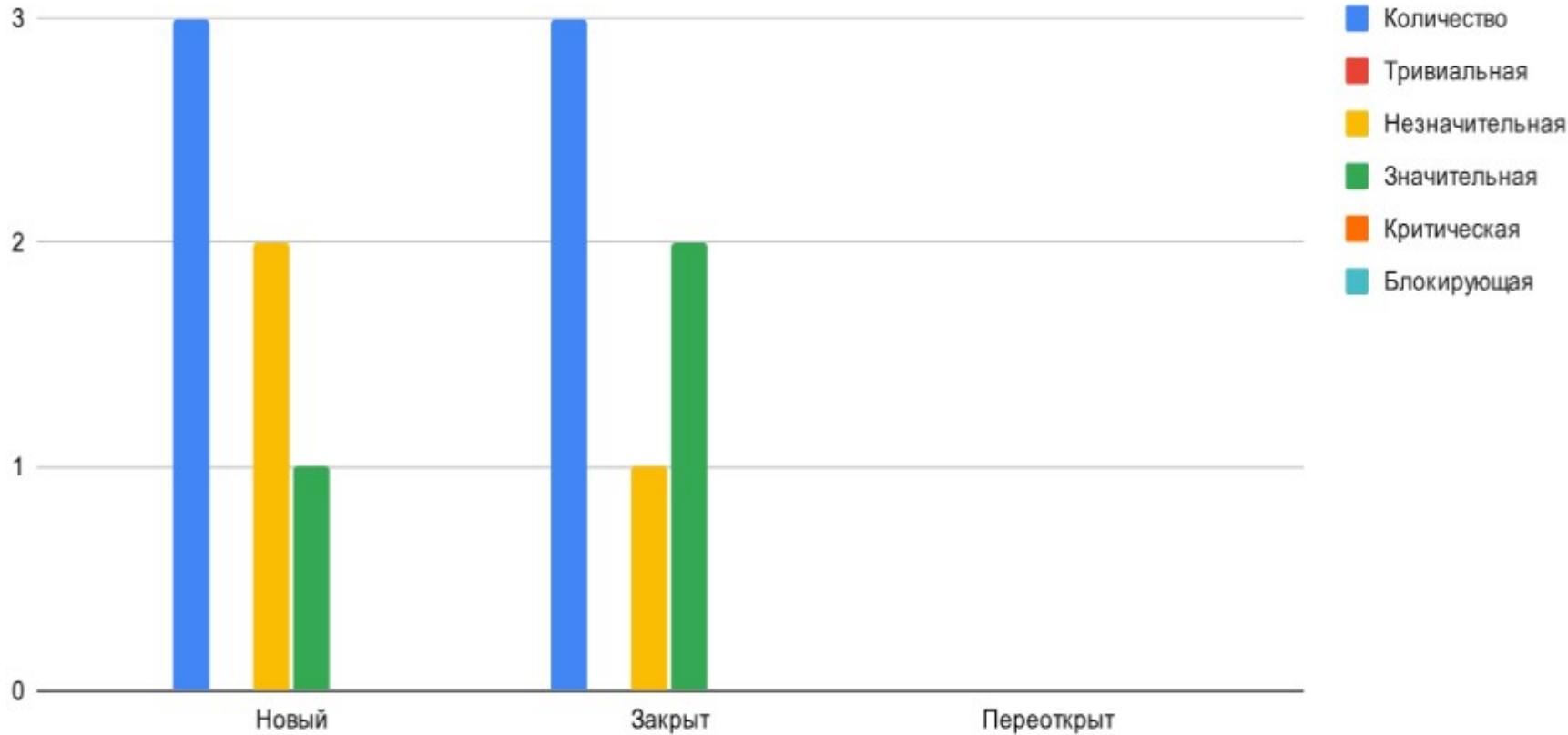
Отчет о тестировании мобильного приложения

Есть ли возможность просмотреть введенный пароль	<input checked="" type="checkbox"/>	Возможность есть
Проверить корректность выбора даты дата пикером	<input checked="" type="checkbox"/>	Дата выбирается корректно
Есть ли автосохранение данных	<input checked="" type="checkbox"/>	Автосохранение есть
Клик по логотипу переносит на главную страницу	<input checked="" type="checkbox"/>	Да

Статистика по дефектам

Статус	Количество 0	Важность				
		Тривиальная	Незначительная	Значительная	Критическая	Блокирующая
Новый	3	0	2	1	0	0
Закрыт	3	0	1	2	0	0
Переоткрыт	0	0	0	0	0	0

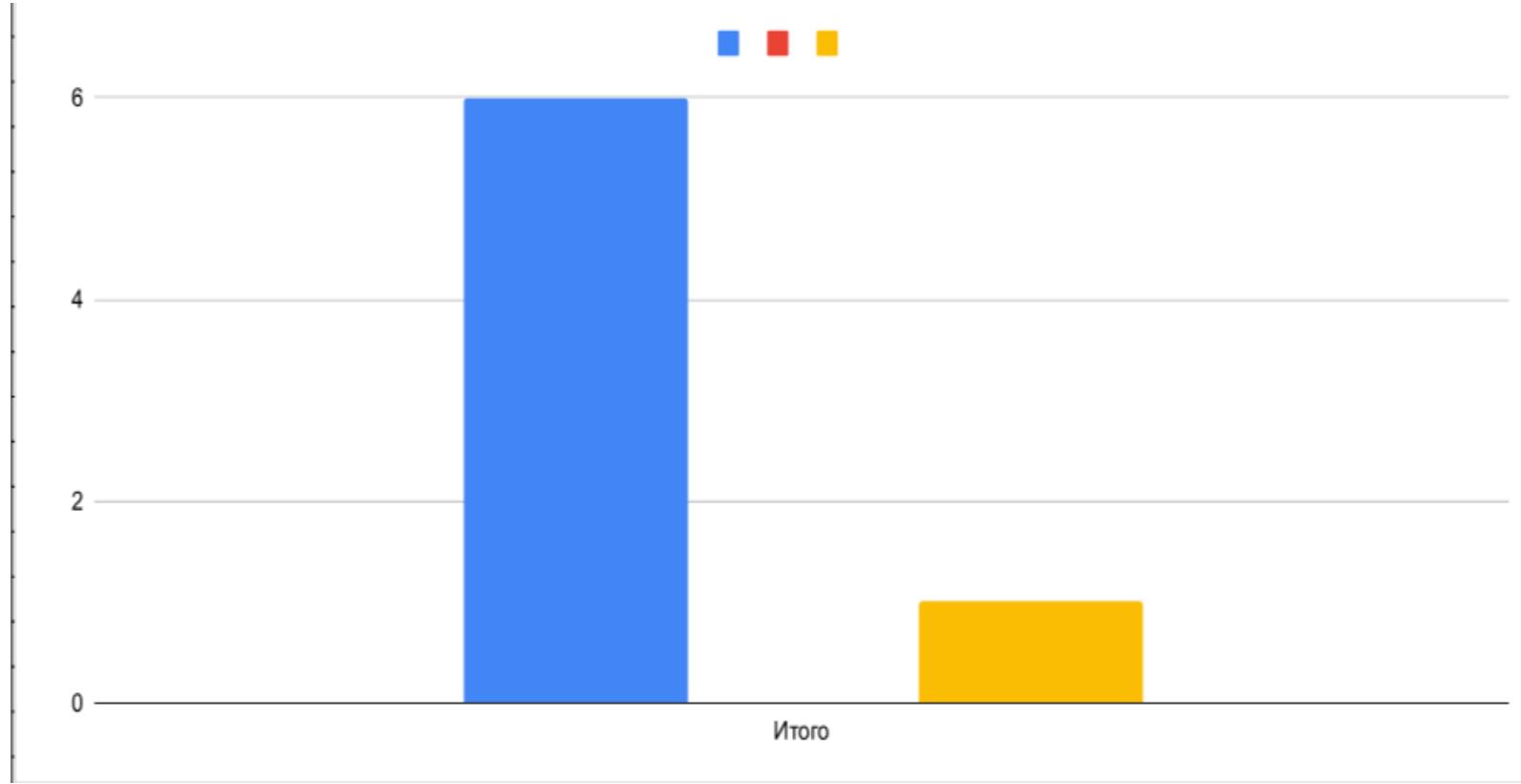
Отчет о тестировании мобильного приложения



Отчет о тестировании мобильного приложения

ID	Frontend	Backend	Mobile
BR-1	✓	□	✓
BR-2	✓	□	✓
BR-3	✓	□	✓
BR-4	✓	□	□
BR-5	✓	□	✓
BR-6	✓	□	✓
Итого	6	0	1

Отчет о тестировании мобильного приложения



Отчет о тестировании мобильного приложения

Выводы и рекомендации

Текущая сборка 1.0.1.1 была успешно инсталлирована под обеими платформами (Windows 10 v22H2, Android V14.0.1). Смоук-тест пройден успешно. Приложение работает стабильно, основная функциональность работоспособна. Особое внимание следует уделить адаптации под мобильные устройства или созданию нативного приложения. Большинство найденных проблем относится к функциональности, внедренной в последнем релизе. Найденные ранее ошибки успешно устранены и верифицированы. Отмечается значительный прогресс в качестве приложения. На текущий момент выполнено более 70% запланированных тестов (остальные планируется выполнить до конца следующей недели). Была обнаружена одна новая ошибка с важностью «значительная».

Билд 1.0.1.7 работает стабильно и рекомендован в продакшн. Все найденные баги закрыты. Рекомендуется уделить особое внимание адаптации приложения под мобильные устройства.

SQL запрос к базе данных

```
- CREATE TABLE products (
    id INT UNSIGNED AUTO_INCREMENT primary key,
    pname VARCHAR(50) not null,
    manufacturer VARCHAR(50) not null,
    count INTEGER UNSIGNED not null,
    price SMALLINT UNSIGNED DEFAULT 0
);

- INSERT products VALUES (null,
    'S24 Ultra CN 12/512',
    'Sanxung', 100, 6740);

- INSERT products VALUES (null,
    'Redmi A5',
    'Xiaomi', 85, 5556);

- INSERT products VALUES (null,
    'Poco C65',
    'Xiaomi', 110, 6562);

- INSERT products VALUES (null,
    'realme Note',
    'realme', 55, 5024);

- INSERT products VALUES (null,
    'Redmi A5',
    'Xiaomei', 20, 5556);

- INSERT products VALUES (null,
    'Spark 40C',
    'Tecno', 110, 6692);

- INSERT products VALUES (null,
    'Spark Go 1',
    'Tecno', 111, 6048);

- INSERT products VALUES (null,
    'CAMON 40 pro',
    'Tecno', 50, 7008);

- INSERT products VALUES (null,
    'Mblu 22',
    'Meizu', 45, 5140);

- INSERT products VALUES (null,
    'HOT 50 i',
    'Infinix', 100, 6798);

SELECT * FROM products ORDER BY price DESC
```

+-----+	id pname	manufacturer	count	price	+-----+
	8 CAMON 40 pro	Tecno	50	7008	
	10 HOT 50 i	Infinix	100	6798	
	1 S24 Ultra CN 12/512	Sanxung	100	6740	
	6 Spark 40C	Tecno	110	6692	
	3 Poco C65	Xiaomi	110	6562	
	7 Spark Go 1	Tecno	111	6048	
	2 Redmi A5	Xiaomi	85	5556	
	5 Redmi A5	Xiaomei	20	5556	
	9 Mblu 22	Meizu	45	5140	
	4 realme Note	realme	55	5024	

Тестирование API

The screenshot shows the Postman application interface. At the top, there are navigation links for 'Workspaces' and 'More', followed by a search bar and various settings icons. Below the header, a toolbar includes icons for 'Create', 'Import', 'Export', 'Settings', 'Vault', and 'Runner'. The main workspace displays a list of requests: 'Get user' (GET), 'Get list' (GET, currently selected), 'Create user' (POST), and 'Delete user' (DELETE). A message 'No environment' is shown with a dropdown arrow. On the left sidebar, there are icons for 'Workspaces', 'Recent', 'Logs', 'Variables', 'Schemas', and 'Collection'. The central area shows a request card for 'Get list users': method 'GET', URL 'https://reqres.in/api/users?page=2', a 'Send' button, and a preview section. Below the card, tabs for 'Docs', 'Params', 'Auth', 'Headers (9)', 'Body', 'Scripts', 'Settings', and 'Cookies' are visible. The 'Post-res' tab contains a script:

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 };  
4 pm.test("Content-Type is present", function () {  
5     pm.response.to.have.header("Content-Type");  
6 });
```

The 'Body' tab shows a successful response: '200 OK' with a duration of '250 ms' and a size of '2.3 KB'. There are buttons to 'Save Response' and 'Copy'. The 'Preview' tab shows the JSON response structure:

```
{  
  "page": 2,  
  "per_page": 6,  
  "total": 12,  
  "total_pages": 2,  
  "data": [  
    {  
      "id": 1,  
      "email": "eve.holt@reqres.in",  
      "first_name": "Eve",  
      "last_name": "Holt",  
      "avatar": "https://reqres.in/img/faces/1.jpg"  
    },  
    {  
      "id": 2,  
      "email": "charlie.wright@reqres.in",  
      "first_name": "Charlie",  
      "last_name": "Wright",  
      "avatar": "https://reqres.in/img/faces/2.jpg"  
    },  
    {  
      "id": 3,  
      "email": "tracey.ramos@reqres.in",  
      "first_name": "Tracey",  
      "last_name": "Ramos",  
      "avatar": "https://reqres.in/img/faces/3.jpg"  
    },  
    {  
      "id": 4,  
      "email": "gwen�smith@reqres.in",  
      "first_name": "Gwen",  
      "last_name": "Smith",  
      "avatar": "https://reqres.in/img/faces/4.jpg"  
    },  
    {  
      "id": 5,  
      "email": "lindsay.ferguson@reqres.in",  
      "first_name": "Lindsay",  
      "last_name": "Ferguson",  
      "avatar": "https://reqres.in/img/faces/5.jpg"  
    },  
    {  
      "id": 6,  
      "email": "kerry.corcoran@reqres.in",  
      "first_name": "Kerry",  
      "last_name": "Corcoran",  
      "avatar": "https://reqres.in/img/faces/6.jpg"  
    }  
  ]  
}
```

At the bottom, there are buttons for 'Cloud View', 'Console', 'Runner', 'Vault', and other navigation icons.

Тестирование API

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Workspaces' and 'More' dropdowns, a search icon, and several tool icons. Below the header, a toolbar includes icons for back, forward, search, and settings, along with a red 'Run' button.

The main workspace displays an 'HTTP API Request / Create user' collection. A 'POST' request is selected, targeting the URL <https://reqres.in/api/users>. The 'Send' button is highlighted in blue.

The 'Headers' tab is active, showing the following configuration:

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});  
pm.test("Content-Type is present", function () {  
    pm.response.to.have.header("Content-Type");  
});
```

The 'Body' tab shows the response details: a status of '201 Created', a response time of '307 ms', and a size of '1.69 KB'. The response body is displayed as JSON:

```
{ } JSON
```

```
{  
  "name": "morpheus",  
  "job": "leader",  
  "id": "684",  
  "createdAt": "2025-11-17T18:11:01.399Z"
```

At the bottom, there are tabs for 'Cloud View', 'Console', 'Runner', 'Vault', and other application tools.

Тестирование API

The screenshot shows the Postman application interface. At the top, there are navigation buttons for Workspaces and More, along with search and settings icons. Below the header is a toolbar with various operation buttons like GET, POST, and DELETE, and a dropdown for environments.

The main workspace displays an API request titled "API Reqres / Delete user". The method is set to "DELETE" and the URL is "https://reqres.in/api/users/5". There are "Save" and "Share" buttons, along with a copy link icon.

The request details section shows the "Headers" tab selected, containing the following code:

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});  
pm.test("Content-Type is present", function () {  
    pm.response.to.have.header("Content-Type");  
});
```

The "Body" tab shows the response status "204 No Content", execution time "332 ms", and size "1.54 KB". There are "Raw", "Preview", and "Visualize" options, along with a "Save Response" button.

At the bottom, there are links for "Cloud View", "Console", "Runner", "Vault", and other navigation icons.

Тестирование API

The screenshot shows the Postman application interface. At the top, there are tabs for Workspaces and More, along with various icons for search, add, settings, and sharing. Below the tabs, a navigation bar shows several requests: GET Bakai, GET Bcx, POST Login, and PATCH Upc. The current request is PATCH Upc, which is highlighted with a red border. The URL for this request is https://reqres.in/api/users/2.

The main area of the interface is the Request section, which includes fields for Method (PATCH), URL, and Headers. Below these, there are tabs for Docs, Params, Auth, Headers (10), Body, Scripts (selected), and Settings. The Scripts tab contains a snippet of JavaScript code:

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });  
4 pm.test("Content-Type is present", function () {  
5     pm.response.to.have.header("Content-Type");  
6 });
```

Below the Scripts tab, the Body tab is selected. It shows a successful response with a status of 200 OK, a duration of 324 ms, and a size of 1.71 KB. The response body is displayed as JSON:

```
{ } JSON ▾
```

```
1 {  
2     "name": "moipheus",  
3     "job": "zion resident",  
4     "updatedAt": "2025-11-17T18:15:32.102Z"  
5 }
```

At the bottom of the interface, there are buttons for Cloud View, Console, Runner, Vault, and other navigation options.

Тестирование API

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Workspaces' and 'More' dropdowns, a search icon, and several status indicators. Below the header, a tab bar shows various requests: 'akai-' (GET), 'Bakai-' (GET), 'Bcex i' (GET), 'Login' (POST, highlighted in yellow), 'PATCH' (PATCH), and '+'. A dropdown menu indicates 'No environment'.

The main workspace displays an 'API Request / Login successful' card. It includes a 'Save' button, a 'Share' button, and a link icon. Below this, a request configuration panel shows 'POST' selected, the URL 'https://reqres.in/api/login', and a large blue 'Send' button.

Underneath the request panel, tabs for 'Docs', 'Params', 'Auth', 'Headers (10)', 'Body', 'Scripts' (which is underlined in red), and 'Settings' are visible. The 'Cookies' section is also present.

In the 'Pre-req' section, there is a script block:

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });  
4 pm.test("Content-Type is present", function () {  
5     pm.response.to.have.header("Content-Type");  
6 });
```

The 'Post-res' section shows the response body:

```
{ } JSON ▾ ▶ Preview ▾ Visualize ▾
```

The response details show a green '200 OK' status, a duration of '135 ms', a size of '1.67 KB', and a save response button. The response body itself is:

```
1 {  
2     "token": "QpwL5tke4Pnpja7X4"  
3 }
```

At the bottom, there are links for 'Cloud View', 'Console', 'Runner', 'Vault', and other navigation icons.

Выводы об оптимальности выбранной стратегии тестирования

Я считаю, что выбранная для учебного проекта стратегия тестирования была оптимальна. В ходе обучения я использовал необходимые техники тест-дизайна, протестировал UI и функциональность приложения. Все это позволило добиться максимального тестового покрытия и отобразить в отчетах все результаты работы.

Презентация подготовленного резюме



1. Контактная информация

Мельников Михаил Анатольевич

тел. 8915986****

[GitHub](#)

2. Желаемая должность

Претендую на позицию QA Engineer, Software Tester

График работы: удаленная работа

Готовность к командировкам: нет

3. Ключевые навыки.

Ручное тестирование — составление тест-плана, тест-кейсов, чек-листов и баг-репортов.

Проведение функционального, регрессионного и кросбраузерного тестирования веб-приложений.

Презентация подготовленного резюме

Использование Devtools для анализа работы сайта.

Нагрузочное тестирование с Apache JMeter.

Написание запросов к базе данных на языке SQL. Тестирование API с помощью программы Postman (написание и отправка HTTP-запросов, получение ответов, создание коллекций запросов, написание тестов на JavaScript, использование готовых тестов (snippetов))

Автоматизированное тестирование с помощью Selenium.

Во время обучения работал над тестовым проектом [Интернет банк - Банк Санкт-Петербург](#)

За время тестирования было проведено 10 тест-кейсов и составлено 3 баг-репорта.

4. Образование

ЯАМТ (Ярославский автомеханический техникум), специальность «Техническое обслуживание станков с ЧПУ и робототехнических комплексов». 1993 — 1997 гг.

АНО ДПО «ЦМИ» специальность «Тестировщик программного обеспечения». 16.10.2025 - 27.11.2025

5. Дополнительная информация

Уровень владения английским A2

Рефлексия

В процессе обучения я приобрел знания в новой для меня сфере тестирования ПО. Получил практические навыки в работе с некоторыми инструментами, например, XMind, TestIT, Jira, Postman, Selenium, Mysql.

Наиболее интересным направлением для меня стало автотестирование, работа с базами данных и API.

Благодарю за внимание!

