### **Задание 3. Тестирование функциональности платформы, подготовка отчетов о найденных дефектах и оценка обнаруженных проблем с альтернативными способами их решения.**

### **Анализ требований**

### **Потенциальные пробелы и уточняющие вопросы**

#### **1. Точность отображения результата**

* Как округляются проценты правильных ответов?
* Что происходит, если пользователь не отвечает ни на один вопрос?
* Отображается ли для пользователя количество правильных и неправильных ответов?
* Видит ли пользователь правильный вариант при неверном ответе?
* Отображаются ли ссылки не рекомендуемые образовательные ресурсы в случае неверного ответа?

#### **2. Список неверных ответов**

* Каким образом предоставляются объяснения (текстом, ссылкой на статью или видео)?
* Объяснения доступны для всех вопросов или только для неверных?
* Есть ли отображение неверных ответов во время прохождения теста?
* Есть ли у пользователя возможность пропускать ответ и возвращаться к нему во время прохождения теста?

#### **3. Предупреждение за 5 минут**

* Предупреждение отображается как всплывающее окно, звуковое оповещение или текст?
* Должно ли предупреждение обновляться, если время изменится из-за внешних факторов?
* Появляется ли таймер при наступлении 5 последних минут?
* Есть ли мигание или смена цвета таймера на последних секундах положенного времени?

#### **4. Сохранение отчетов в личном кабинете**

* Есть ли ограничение на количество сохраняемых отчетов?
* Происходит ли автоматическое удаление старых отчетов?
* Есть ли запрет на сохранение нового отчета до удаления старых?
* Можно ли удалить/скачать отчет пользователю локально?
* В каком формате происходит сохранение отчета локально?
* Сохраняются ли стили и форматирование?
* Работает ли функционал во всех поддерживаемых браузерах?

#### 

#### **5. Обработка ошибок**

* Что происходит с данными теста при потере соединения?
* Данные сохраняются локально (например, в кэше браузера) и синхронизируются с сервером после восстановления соединения?
* Данные теряются, если соединение пропадает до сохранения?
* Уточнить механизм работы: сохраняются ли промежуточные результаты автоматически или пользователь должен вручную подтвердить сохранение?
* Есть ли механизм восстановления сеанса?  
  Если пользователь случайно закрыл браузер или произошел сбой, предусмотрен ли механизм восстановления ?
* Восстанавливается ли тест с последнего сохраненного шага?
* Всплывает ли сообщение с предложением продолжить или начать заново?
* Восстанавливается ли пройденное время?
* Засчитывается ли попытка как «тест не пройден» при потере соединения?
* При пропуске ответов теста получит ли пользователь ответы на вопросы при завершении теста?

#### **6. Кроссбраузерная совместимость**

* Какие версии браузеров поддерживаются?
* На каких мобильных ОС доступно приложение?
* Какие мобильные браузеры поддерживают работу приложения?
* Проверяется ли поддержка устаревших браузеров?

### **Возможные несоответствия**

1. **Результаты округляются некорректно.**Например, 87.5% отображается как 87% вместо 88%. Это может запутать пользователей.
2. **Отсутствие реакции на пустые тесты.**Если пользователь не отвечает ни на один вопрос, система может не предоставить ожидаемого результата (например, 0%) или зависнуть.
3. **Предупреждение за 5 минут не обновляется.**Пользователь может получить уведомление не вовремя, если серверное время изменилось.
4. **Ограничение на сохранение отчетов не задокументировано.**Пользователь может столкнуться с неожиданным запретом на добавление нового отчета.
5. **Потеря данных при сбоях соединения.**Если механизм автоматического сохранения не предусмотрен, пользователи потеряют прогресс.
6. **Поддержка устаревших браузеров.**Некоторые пользователи могут использовать устаревшие версии, что приведет к проблемам с отображением интерфейса или функциональностью.

### **Разработка тест-кейсов**

### **Тест-кейсы для функционального и кроссбраузерного тестирования**

### **Функциональное тестирование**

| **ID** | **Название теста** | **Шаги воспроизведения** | **Ожидаемый результат** |
| --- | --- | --- | --- |
| **TC\_001** | Проверка отображения результата | Предусловие:   * Тестовое задание, состоящее из 3 вопросов, пройдено с 2 правильными и 1 неправильным ответом. * Страница с результатами тестового задания открыта.   Шаги:   1. Проверить общий процент правильных ответов. 2. Проверить кол-во неверных ответов. 3. Проверить, что в списке неверных ответов отображается тот, на который был дан неверный ответ. 4. Проверить, что отображается правильный ответ на неверный ответ. 5. Проверить, что рекомендации отображаются для неверного ответа и они консистентные. | 1. 67% 2. 1 3. Верно 4. Верно 5. Верно |
|
| **TC\_002** | Отображение предупреждения за 5 минут до окончания | Предусловие:   * Тест начат, таймер установлен на 8 минут.   Шаги:   1. Дождаться, когда до окончания теста останется 5 минут, наблюдать за экраном. | 1. Должно появиться предупреждающее уведомление с текстом и таймер с обратным отсчетом времени. |
| **TC\_003** | Проверка сохранения отчета в личном кабинете | Предусловие:   * Проверяемое условие - тестовое задание, состоящее из 3 вопросов, пройти с 2 правильными и 1 неправильным ответом.   Шаги:   1. Пройти тест в соответствии с условием. 2. Проверить наличие отчета в разделе «Мои тесты». 3. Открыть отчет. 4. Сравнить соответствие результатов на экране и в отчете. | 1. Тест завершен, условие выполнено. 2. Завершенный тест отображается в разделе "Мои тесты". 3. Успешно. 4. Результат теста совпадает. |
| **TC\_004** | Проверка корректности объяснений | Предусловие:   * Тест завершен, тестовое задание, состоящее из 3 вопросов, пройдено с 2 правильными и 1 неправильным ответом. * Отчет прохождения теста открыт.   Шаги:   1. Проверить объяснения для неверных ответов. | 1. Объяснения соответствуют допущенным ошибкам. |
| **TC\_005** | Обработка потери соединения | Предусловия:   * Интернет-соединение стабильно на момент начала теста.   Шаги:   1. Приступить к выполнению теста. 2. Во время теста отключить интернет. 3. Восстановить подключение через 1–2 минуты. 4. Проверить сохраненный прогресс. 5. Проверить наличие прогресса в системе. 6. Продолжить выполнение теста. | 1. Успешно. 2. Появляется уведомление о разрыве соединения. 3. Появляется сообщение об успешном восстановлении соединения. 4. Данные восстанавливаются. 5. Ранее сохраненные ответы остаются в системе. 6. Тест продолжается с момента обрыва соединения без потери данных. |
| **TC\_006** | Скачивание отчета | Предусловия:   * Пользователь завершил тест, отчет доступен в личном кабинете.   Шаги:   1. Нажать на кнопку скачивания отчета. 2. Открыть загруженный файл. | 1. Отчет скачан успешно. 2. Загруженный файл открывается без ошибок совместимости. |
| **TC\_007** | Пустой тест | Предусловие:   * Проверяемое условие - тестовое задание, состоящее из 3 вопросов, пройти, пропуская ответы.   Шаги:   1. Проверить результаты теста. | 1. Отображается результат с 0% правильных ответов. |

### **Кроссбраузерное тестирование**

| **ID** | **Название теста** | **Шаги воспроизведения** | **Ожидаемый результат** |
| --- | --- | --- | --- |
| **TC\_001** | Тестирование функционала в браузере Chrome | Предусловия:   * Установлена последняя версия браузера Google Chrome. * Пользователь авторизован в системе и имеет доступ к тесту. * Тест открыт в браузере Google Chrome.   Шаги:   1. Пройти тест, ответив на все вопросы. 2. Завершить тест. 3. Перейти в личный кабинет. 4. Открыть отчет. | 1. Каждое действие пользователя фиксируется системой без задержек, интерфейс работает корректно. 2. Появляется страница с результатами, отображающая:% правильных и неправильных ответов соответственно,Объяснения для неверных ответов. 3. В разделе "Мои тесты" появляется отчет о завершенном тесте. 4. Отчет доступен для просмотра, Содержимое отчета соответствует результатам теста. |
| **TC\_002** | Тестирование функционала на мобильном устройстве | Предусловия:   * Устройство под управлением iOS версии 14 или выше. * Установлен браузер Safari (последняя версия). * Приложение открыто в мобильной версии браузера Safari.   Шаги:   1. Пройти тест, ответив на все вопросы. 2. Завершить тест. 3. Перейти в личный кабинет. 4. Открыть отчет. | 1. Интерфейс адаптирован для мобильного устройства, элементы отображаются корректно (кнопки, текстовые поля). 2. Появляется страница с результатами, отображающая:% правильных и неправильных ответов соответственно,Объяснения для неверных ответов. 3. Раздел "Мои тесты" доступен, отчет о тесте отображается. 4. Отчет корректно отображается на мобильном устройстве, можно прокрутить отчет без искажений интерфейса. |
| **TC\_003** | Тестирование функционала в браузере Firefox | Предусловия:   * Установлена последняя версия браузера Mozilla Firefox. * Пользователь авторизован и имеет доступ к тесту. * Тест открыт в браузере Firefox.   Шаги:   1. Пройти тест, ответив на все вопросы. 2. Завершить тест. 3. Перейти к разделу "Мои тесты" в личном кабинете. 4. Открыть отчет. | 1. Все элементы интерфейса (например, кнопки, текстовые поля) работают корректно, без задержек или ошибок. 2. Появляется страница с результатами, отображающая:% правильных и неправильных ответов соответственно,Объяснения для неверных ответов. 3. Отчет теста отображается в списке завершенных тестов. 4. Содержимое отчета соответствует результатам теста, отчет доступен для скачивания. |

### 

### **3. Тестирование**

#### **Подходы к выполнению ручного тестирования**

1. **Тестирование API с использованием Postman**:

**Цель:** Проверить корректность взаимодействия клиента и сервера при работе с отчетами.

**Действия:**

Создать коллекцию запросов для API. Например:

* + - 1. **GET /reports** — получение списка отчетов пользователя.
      2. **POST /reports** — сохранение нового отчета.
      3. **GET /reports/{id}** — получение данных конкретного отчета.

Настроить тестовые данные (токены авторизации, ID пользователей и отчетов).

Выполнить тесты с различными сценариями:

* + - 1. Корректные данные запроса (валидные токены, существующие отчеты).
      2. Неверные данные (отсутствие токена, несуществующий ID отчета).
      3. Граничные условия (пустые поля, превышение лимитов).

Проверить, возвращает ли API ожидаемые статусы и данные (например, 200 OK, 401 Unauthorized, 404 Not Found).

1. **Ручное тестирование интерфейса на разных устройствах и браузерах**:

**Цель:** Убедиться в корректности отображения интерфейса и функциональности на различных платформах.

**Действия:**

Тестировать приложение на популярных браузерах: Chrome, Firefox, Edge, Safari.

Использовать инструменты эмуляции в DevTools (например, эмуляция мобильных устройств в Chrome DevTools).

Проверить все элементы интерфейса:

* + - 1. Корректность расположения кнопок, полей и сообщений.
      2. Работа адаптивного дизайна на мобильных устройствах.
      3. Отображение отчетов в личном кабинете.

Выполнять ключевые сценарии: прохождение теста, завершение, проверка результатов.

1. **Эмуляция плохого соединения через инструменты браузера (DevTools):**

**Цель:** Оценить работу приложения в условиях нестабильного интернета.

**Действия:**

Открыть DevTools в браузере.

Перейти в раздел **Network** и установить ограничение скорости (например, "Slow 3G").

Выполнить следующие сценарии:

* + - 1. Начать тест, затем потерять соединение, восстановить его и проверить, сохраняется ли прогресс.
      2. Завершить тест при медленном соединении, проверить, корректно ли отправляются данные.
      3. Открыть личный кабинет и проверить загрузку отчетов.

#### **Сценарии для автоматизации тестирования**

1. **Проверка отображения результатов теста**:

**Шаги автоматизации:**

Авторизоваться под тестовым пользователем.

Пройти тест (можно эмулировать через API).

Перейти на экран результатов.

Проверить отображение:

* + - 1. Процент правильных ответов.
      2. Список неверных ответов с объяснениями.
      3. Рекомендации.

**Инструменты:** Selenium, Playwright.

**Почему важно:** Это ключевой сценарий, который показывает пользователю результаты его работы.

1. **Сохранение отчетов и их доступность в личном кабинете**:

**Шаги автоматизации:**

Завершить тест (через интерфейс или API).

Перейти в личный кабинет.

Проверить, что отчет отображается в разделе «Мои тесты».

Сравнить содержимое отчета (заголовок, время, данные).

**Инструменты:** Selenium, Postman (для проверки сохранения через API).

**Почему важно:** Пользователь ожидает стабильного сохранения отчетов для дальнейшего анализа.

1. **Обработка ошибок соединения**:

**Шаги автоматизации:**

Запустить тест.

Эмулировать потерю соединения (например, через DevTools или инструмент throttling).

Проверить:

* + - 1. Сохранился ли прогресс теста.
      2. Восстановилась ли связь после подключения интернета.
      3. Корректно ли отображается сообщение об ошибке.

**Инструменты:** Selenium, Puppeteer, или специализированные библиотеки для эмуляции сетевых условий.

**Почему важно:** Потеря соединения — частый сценарий, и приложение должно сохранять стабильность.

### **4. Отчетность**

### **Баг №1: Отсутствие предупреждения за 5 минут до окончания**

## **Summary**

Предупреждение за 5 минут до окончания теста не отображается, несмотря на то, что время подходит к указанному порогу.

## **Environment**

**Билд:** v1.0.0

**Окружение:** Тестовое

**Клиент:**

Браузеры: Chrome 119.0.0, Firefox 120.0.1, Safari 17.0

Операционные системы: Windows 11, macOS Ventura 13.5

Устройства: ПК, MacBook

**Дата тестирования:** 2024-12-11

## **Steps to reproduce**

1. Авторизоваться на платформе под тестовым аккаунтом пользователя.
2. Перейти в раздел доступных тестов и выбрать любой тест.
3. Начать прохождение теста.
4. Убедиться, что общее время теста составляет, например, 8 минут.
5. Дождаться момента, когда до окончания теста остается 5 минут (следить за таймером в верхней части экрана).
6. Проверить, отображается ли предупреждение.

## **Expected result**

За 5 минут до окончания теста появляется визуальное уведомление (всплывающее окно или выделенное сообщение в интерфейсе) с текстом, предупреждающим о том, что время подходит к концу.

## **Actual result**

Предупреждение за 5 минут не отображается. Пользователь не получает никакого сигнала о приближении времени окончания.

## **Recommendations**

1. Проверить, корректно ли работает таймер и передает ли он сигнал модулю уведомлений.
2. Убедиться, что условие отображения уведомления за 5 минут прописано в коде и выполняется.
3. Добавить автоматические тесты для проверки логики таймера.

### **Баг №2: Отчет не сохраняется в личном кабинете**

## **Summary**

После завершения теста отчет с результатами не сохраняется в личном кабинете пользователя.

## **Environment**

**Билд:** v1.0.0

**Окружение:** Тестовое

**Клиент:**

Браузеры: Chrome 119.0.0, Firefox 120.0.1

Операционные системы: Windows 11, macOS Ventura 13.5

Устройства: ПК, MacBook

**Дата тестирования:** 2024-12-11

## **Steps to reproduce**

1. Авторизоваться на платформе под тестовым аккаунтом пользователя.
2. Перейти в раздел доступных тестов.
3. Выбрать любой тест из списка и начать прохождение.
4. Завершить тест, ответив на все вопросы или дождавшись окончания времени.
5. Перейти в личный кабинет (раздел «Мои тесты»).
6. Проверить, отображается ли отчет с результатами только что завершенного теста.

## **Expected result**

Отчет с результатами теста (включающий процент правильных ответов, список неверных с объяснениями и рекомендации) появляется в личном кабинете в разделе «Мои тесты».

## **Actual result**

Отчет с результатами не отображается в личном кабинете. Раздел «Мои тесты» остается пустым.

## **Recommendations**

1. Проверить API-сервис сохранения отчетов на сервере (возможно, запрос не отправляется или завершается с ошибкой).
2. Проверить, передаются ли результаты теста от фронтенда к серверу.
3. Добавить логирование и уведомления об ошибках на случай, если отчет не может быть сохранен.
4. Разработать интеграционные тесты для проверки взаимодействия фронтенда и бэкенда в этом сценарии.

### 

### **Баг №3: Некорректное отображение результатов на мобильных устройствах**

## **Summary**

На мобильных устройствах список неверных ответов не помещается на экран, что приводит к необходимости горизонтального скроллинга, усложняя навигацию.

## **Environment**

**Билд:** v1.0.0

**Окружение:** Тестовое

**Клиент:**

Браузеры: Safari 17.0, Chrome 119.0.0 (мобильная версия)

Операционные системы: iOS 17.1, Android 14

Устройства: iPhone 14, Samsung Galaxy S23

**Дата тестирования:** 2024-12-11

## **Steps to reproduce**

1. Авторизоваться на платформе через мобильное устройство.
2. Перейти в раздел доступных тестов и начать тестирование.
3. Завершить тест, ответив на все вопросы.
4. Дождаться отображения результатов теста.
5. Проверить, помещается ли весь текст списка неверных ответов на экран без горизонтального скроллинга.

## **Expected result**

Список неверных ответов адаптируется под ширину экрана мобильного устройства. Все элементы видны без необходимости горизонтального скроллинга.

## **Actual result**

Список неверных ответов частично выходит за границы экрана, из-за чего требуется горизонтальный скроллинг для просмотра содержимого.

## **Recommendations**

1. Настроить адаптивные стили (CSS) для корректного отображения элементов на мобильных устройствах.
2. Проверить использование CSS-свойства word-wrap или overflow-wrap для предотвращения выхода текста за границы.
3. Провести тестирование на нескольких разрешениях экранов, чтобы убедиться, что проблема устранена.

### **Основания для оценки серьезности проблем**

1. **Баг с предупреждением времени**Этот баг оценивается как критичный, поскольку он непосредственно влияет на возможность пользователя завершить тест вовремя. Если пользователь пропустит предупреждение или получит его в неверный момент, он может потерять шанс ответить на оставшиеся вопросы. Это напрямую затрагивает функциональность системы и вызывает недовольство пользователей.  
   **Критичность**:

Затрагивает всех пользователей.

Связан с функционалом тестирования, который является основным.

Приводит к потере результата, что является ключевой проблемой.

1. **Проблема с мобильным интерфейсом**Эта проблема менее критична, так как связана с удобством, а не с функциональностью. Пользователь может все же пройти тест, но в неудобных условиях, например, если элементы интерфейса отображаются некорректно или требуется больше времени на взаимодействие.  
   **Критичность**:

Затрагивает только часть аудитории (пользователей мобильных устройств).

Проблемы с интерфейсом не препятствуют завершению теста.

Негативно влияет на впечатление о продукте.

### **Возможные способы улучшения**

#### **1. Исправление бага с предупреждением времени**

1. **Проверить корректность логики расчета времени**Убедиться, что расчет времени на клиенте синхронизируется с серверным временем. Использовать:
   1. **WebSocket или Server-Sent Events (SSE)** для постоянной синхронизации времени.
   2. Корректировать предупреждение, если сервер изменяет оставшееся время.
2. **Добавить визуальные и звуковые индикаторы**Использовать всплывающее окно с видимым таймером, которое невозможно пропустить. Например:
   1. Отобразить обратный отсчет в окне.
   2. Добавить звук или вибрацию для мобильных устройств.
3. **Автосохранение при предупреждении**При отображении предупреждения сохранить текущие ответы пользователя, чтобы минимизировать потерю данных.

#### **2. Улучшение мобильного интерфейса**

1. **Протестировать адаптивность интерфейса**Использовать инструменты браузера (DevTools) для проверки отображения интерфейса на различных разрешениях (320px, 375px, 414px и т.д.). Проблемы с адаптивностью можно решить:
   1. Настройкой CSS (например, через @media queries).
   2. Использованием flexbox или grid для адаптивной верстки.
2. **Оптимизировать для тач-устройств**Увеличить размеры кнопок и интервал между ними для удобного взаимодействия. Пример: минимальный размер кнопки — 48px × 48px согласно рекомендациям Google.
3. **Добавить тестирование на реальных устройствах**Убедиться, что элементы интерфейса корректно работают на популярных устройствах (iOS и Android) и браузерах (Chrome, Safari).

#### **3. Комбинированные улучшения**

1. **Обеспечить кроссбраузерное и кроссплатформенное тестирование**Использовать инструменты вроде BrowserStack или Sauce Labs для тестирования интерфейса и функционала в разных браузерах и на мобильных устройствах.
2. **Добавить пользовательские настройки предупреждений**Дать пользователю возможность выбрать тип предупреждения: звуковое, визуальное или вибрация (на мобильных устройствах). Это улучшит пользовательский опыт и позволит учесть разные предпочтения.

### **Обоснование предложенных решений**

1. **Тестирование и исправление бага с предупреждением времени** обеспечивает, что пользователь всегда получает актуальное уведомление. Синхронизация с сервером исключает расхождения времени.
2. **Мобильная оптимизация интерфейса** улучшает опыт пользователей мобильных устройств, увеличивает охват аудитории и делает продукт удобным для большего числа пользователей.
3. **Автосохранение данных** и выбор типа уведомления повышают надежность системы и пользовательский комфорт.

### **Применение методологий и инструментов:**

### **1. Postman для тестирования API**

**Как использовался:**

Отправка запросов к API (GET, POST, PUT, DELETE).

Проверка структуры и содержимого ответов (JSON).

Автоматизация повторяющихся тестов с использованием коллекций.

Тестирование граничных случаев: отсутствие параметров, неверные данные.

**Почему выбрали:**

Удобство работы с API без написания кода.

Встроенные возможности для проверки данных (тесты на JavaScript).

Легкость создания и управления коллекциями тестов.

**Зачем:**

Чтобы убедиться, что серверная часть работает корректно и обрабатывает запросы в любых условиях.

Для проверки функционала до создания клиентской части (веб-интерфейса).

### **2. BrowserStack для кроссбраузерного тестирования**

**Как использовался:**

Проверка интерфейса и функционала на разных браузерах (Chrome, Safari, Firefox, Edge) и операционных системах.

Тестирование на мобильных устройствах (iOS, Android).

**Почему выбрали:**

Поддержка множества реальных устройств и браузеров.

Возможность тестировать без необходимости физического доступа к устройствам.

**Зачем:**

Для выявления проблем с отображением интерфейса на устаревших или менее популярных браузерах.

Чтобы гарантировать доступность продукта для всех пользователей.

### **3. DevTools для тестирования UI и производительности**

**Как использовался:**

Проверка адаптивности интерфейса (режимы эмуляции мобильных устройств).

Эмуляция медленного соединения (например, 2G/3G) для тестирования работы в условиях низкой скорости интернета.

Анализ сетевых запросов (Network tab), чтобы выявить задержки или ошибки.

**Почему выбрали:**

Встроенный инструмент, доступный в любом браузере.

Удобство анализа без дополнительных программ.

**Зачем:**

Для проверки поведения интерфейса и функционала в условиях, приближенных к реальным.

Для оптимизации скорости загрузки страницы.

### **4. Selenium или Playwright для автоматизации тестирования**

**Как использовался:**

Написание автотестов для проверки критического функционала:

Отображение результата теста.

Сохранение отчетов в личном кабинете.

Проверка обработки ошибок при потере соединения.

**Почему выбрали:**

Возможность автоматизировать тесты и выполнять их регулярно (например, в CI/CD).

Поддержка работы с любыми браузерами и платформами.

**Зачем:**

Чтобы сократить время на регрессионное тестирование.

Для обеспечения стабильности ключевого функционала при внесении изменений.

### **5. JIRA/Trello для управления тестированием**

**Как использовался:**

Постановка задач на тестирование.

Ведение тест-планов и чек-листов.

Отслеживание прогресса тестирования и фиксация найденных багов.

**Почему выбрали:**

Возможность структурировать работу команды.

Удобное визуальное представление задач (Kanban-доска).

**Зачем:**

Чтобы следить за процессом тестирования и своевременно устранять обнаруженные проблемы.

Для повышения прозрачности работы команды.

Каждый инструмент использовался, чтобы покрыть различные аспекты тестирования:

1. Postman позволил гарантировать стабильную работу API.
2. BrowserStack обеспечил совместимость с разными устройствами и браузерами.
3. DevTools помог оптимизировать производительность и адаптивность интерфейса.
4. Автоматизация тестов с помощью Selenium/Playwright ускорила проверку регрессионных сценариев.
5. JIRA/Trello способствовали упорядочению работы и фиксации всех обнаруженных проблем.