

Тестирование документации Лаб 7 Сиразиева Яна Пи21-3

Цель работы

Цель работы заключается в проведении анализа документации, описывающей требования к проектированию, разработке программного обеспечения и структуре базы данных проекта веб-сайта о Стамбуле. Необходимо оценить качество документации, выявить несоответствия и предложить рекомендации по их исправлению.

Описание тестируемой документации

Тестируемая документация включает следующие разделы:

- **Требования к проектированию и разработке веб-сайта:**
 1. Общие требования
 2. Функциональные требования
 3. Нефункциональные требования
- **Структура базы данных:**
 1. Описание таблиц
 2. Описание связей между таблицами
- **Описание функциональности и пользовательского интерфейса:**
 1. Механизмы выбора языка
 2. Календарь для выбора дат пребывания
 3. Отображение мероприятий и экскурсий
 4. Фильтрация по ценовому диапазону
 5. Отображение текущей погоды
- **Интеграция с внешними API:**
 1. Конвертация валют по текущему курсу Центрального банка
 2. Получение данных о погоде из Google Погода

Эта документация будет условной и обобщённой, чтобы соответствовать описанному проекту. В реальном проекте такая документация может включать гораздо больше деталей и подразделов, но для целей данного анализа предположим, что именно эти разделы присутствуют и требуют оценки

Описание критериев качества тестируемой документации

Для оценки качества документации используются *следующие критерии:*

- Полнота – вся необходимая информация присутствует в документации.
- Точность – информация в документации точна и актуальна.
- Понятность – документация написана ясным и доступным языком.

- Последовательность – структура и форматирование документации последовательны.
- Соответствие стандартам – документация соответствует общепринятым стандартам и лучшим практикам.

Описание и обоснование метода тестирования документации

Метод тестирования документации включает следующие этапы:

- Чтение и анализ – детальное изучение всех разделов документации.
- Сравнение с критериями качества – оценка каждого раздела на соответствие вышеуказанным критериям.
- Выявление несоответствий – фиксация всех обнаруженных несоответствий.
- Формулирование рекомендаций – разработка предложений по устранению выявленных несоответствий.

Теперь составим пример такой документации для проекта сайта о Стамбуле:

1. Требования к проектированию и разработке веб-сайта

1.1. Общие требования

- Веб-сайт должен быть разработан с использованием React.
- Поддержка многоязычности: русский, турецкий, английский, арабский и китайский языки.
- Совместимость с основными браузерами (Chrome, Firefox, Safari, Edge).

1.2. Функциональные требования

- Пользователь может выбрать язык интерфейса.
- Возможность указания дат пребывания через календарь.
- Отображение расписания мероприятий, доступных в выбранные даты.
- Фильтрация мероприятий по ценовому диапазону.
- Интеграция с внешними сервисами для конвертации валют и отображения погоды.

1.3. Нефункциональные требования

- Высокая производительность и быстрая загрузка страниц.
- Обеспечение безопасности пользовательских данных.
- Модульная и расширяемая архитектура кода.

2. Структура базы данных

2.1. Таблицы

Users user_id (PK) username	Events event_id (PK) event_name	Bookings booking_id (PK) user_id (FK)
--	--	--

email password preferred_language	description date price location	event_id (FK) booking_date
---	--	-------------------------------

2.2. Связи между таблицами

Users - Bookings

Events - Bookings

3. Описание функциональности и пользовательского интерфейса

3.1. Механизмы выбора языка

Пользователь может выбрать язык из выпадающего списка на главной странице. Выбор языка сохраняется в cookies для дальнейших сессий.

3.2. Календарь для выбора дат пребывания

Пользователь может выбрать диапазон дат с помощью календарного виджета. После выбора дат отображается список доступных мероприятий.

3.3. Отображение мероприятий и экскурсий

На основе выбранных дат отображается список мероприятий с возможностью фильтрации по цене.

3.4. Фильтрация по ценовому диапазону

Пользователь может установить минимальную и максимальную цену для отображения мероприятий в заданном диапазоне.

3.5. Отображение текущей погоды

На главной странице отображается текущая погода в Стамбуле, полученная из Google Погода.

4. Интеграция с внешними API

4.1. Конвертация валют

Использование API Центрального банка для получения текущих курсов валют и их конвертации в реальном времени.

4.2. Получение данных о погоде

Использование API Google Погода для получения актуальной информации о погоде в Стамбуле.

Список несоответствий в документации критериям качества

1. Полнота

- Требования к проектированию и разработке: Отсутствует описание механизмов аутентификации и авторизации пользователей.
- *Рекомендация:* Добавить раздел о требованиях к аутентификации и авторизации, включая описание процесса регистрации и входа пользователей.
- Структура базы данных: Не указаны типы данных для полей таблиц.
- *Рекомендация:* Добавить типы данных для каждого поля в таблицах базы данных.

2. Точность

- Интеграция с внешними API: Не указаны конкретные API и методы для конвертации валют и получения данных о погоде.
- *Рекомендация:* Указать конкретные API

3. Понятность

- Описание функциональности и пользовательского интерфейса: Недостаточно деталей о механизме выбора языка.
- *Рекомендация:* Добавить скриншоты и примеры кода, показывающие, как реализован выбор языка и как сохраняются предпочтения пользователя.

4. Последовательность

- Функциональные требования: Не все требования согласуются с описанием пользовательского интерфейса.
- *Рекомендация:* Проверить, что все функциональные требования (например, поддержка многоязычности) описаны одинаково в разных разделах документации.

5. Соответствие стандартам

- Описание пользовательского интерфейса: Отсутствуют ссылки на используемые стандарты UX/UI.
- *Рекомендация:* Добавить ссылки на стандарты UX/UI, которые используются при разработке интерфейса.

Выводы по работе

Документация проекта веб-сайта о Стамбуле охватывает основные аспекты разработки и функциональности, но имеет некоторые недостатки в плане полноты, точности, понятности, последовательности и соответствия стандартам. Выявленные несоответствия требуют дополнений и уточнений,

что повысит качество документации и облегчит дальнейшую разработку и поддержку проекта.

Используемые источники:

1. Документация в порядке // <https://habr.com/ru/articles/549588/>
2. Техническая документация по разработке ПО // <https://habr.com/ru/articles/507414/>
3. Тестирование документации к программным продуктам // <https://habr.com/ru/articles/346290/>
4. Как проводить тестирование документации // <https://sky.pro/media/kak-provodit-testirovanie-dokumentaczii/>