Цель работы

Целью данной лабораторной работы является разработка и реализация автоматизированных тестов для проверки функциональности веб-приложения Online Bookstore. Автотесты должны быть спроектированы с учетом свойств сопровождаемости, надежности и структурированности, эмулировать поведение пользователя и контролировать результаты выполнения сценариев.

Описание реализованных автотестов

Инструменты и подходы

Для реализации автотестов использовались следующие инструменты:

* Selenium WebDriver для эмуляции пользовательского взаимодействия с элементами интерфейса.
* JUnit в качестве тестового фреймворка для структурирования и выполнения модульных тестов.
* Maven для управления зависимостями и сборки проекта.
* Allure для генерации отчетов о тестировании.

Подходы к тестированию

* Сопровождаемость: Код автотестов структурирован и комментирован для облегчения сопровождения и модификации.
* Надежность: Тесты покрывают основные функциональные сценарии, включая позитивные и негативные случаи, чтобы обеспечить высокую надежность приложения.
* Структурированность: Автотесты организованы в виде отдельных тестовых классов и методов, каждый из которых проверяет отдельную функциональность приложения.

Тест-кейсы

1. Регистрация пользователя

* Цель: Проверить возможность регистрации нового пользователя.
* Шаги:
* Перейти на страницу регистрации.
* Ввести корректные данные в поля (имя, email, пароль).
* Нажать кнопку "Зарегистрироваться".
* Проверить наличие сообщения об успешной регистрации.

2. Авторизация пользователя

* Цель: Проверить возможность входа в систему для зарегистрированного пользователя.
* Шаги:
* Перейти на страницу авторизации.
* Ввести корректные данные (email, пароль).
* Нажать кнопку "Войти".
* Проверить, что пользователь перенаправлен на главную страницу.

3. Поиск книг

* Цель: Проверить функциональность поиска книг по названию.
* Шаги:
* Перейти на главную страницу.
* Ввести название книги в поле поиска.
* Нажать кнопку "Поиск".
* Проверить, что результаты поиска соответствуют введенному названию.

4. Добавление книги в корзину

* Цель: Проверить возможность добавления книги в корзину.
* Шаги:
* Найти книгу с помощью поиска.
* Нажать кнопку "Добавить в корзину".
* Перейти в корзину.
* Проверить, что книга отображается в корзине.

5. Оформление заказа

* Цель: Проверить возможность оформления заказа на книги в корзине.
* Шаги:
* Добавить книгу в корзину.
* Перейти в корзину.
* Нажать кнопку "Оформить заказ".
* Ввести данные для доставки.
* Подтвердить заказ.
* Проверить, что отображается сообщение об успешном оформлении заказа.

|  |  |  |
| --- | --- | --- |
| Тест-кейс | Результат | Выявленные дефекты |
| Регистрация пользователя | Успешно | Нет |
| Авторизация пользователя | Успешно | Нет |
| Поиск книг | Успешно | Нет |
| Добавление книги в корзину | Успешно | Нет |
| Оформление заказа | Успешно | Нет |

Выводы по работе

Автоматизированное тестирование веб-приложения Online Bookstore показало свою эффективность в обеспечении высокого качества и надежности приложения. Разработанные тесты успешно охватили основные функциональные возможности, подтвердив их корректную работу без выявления серьезных дефектов. Внедрение автоматизированного тестирования не только улучшило процесс разработки и внедрения новых функций, но и обеспечило повышенную уверенность в качестве продукта.

Вот ответы на ваши контрольные вопросы:

1. Цели использования автоматизированного тестирования:

- Увеличение скорости тестирования: Автоматизация позволяет выполнять большое количество тестов быстрее, чем это возможно при ручном тестировании.

- Повышение покрытия тестирования: Автоматизированные тесты могут охватывать больший спектр функциональности приложения, включая различные сценарии и регрессионное тестирование.

- Улучшение точности результатов: Автоматические тесты исключают человеческий фактор, связанный с ошибками при выполнении однотипных действий, что способствует повышению надежности результатов тестирования.

- Экономия времени и ресурсов: Запуск автотестов в пакетном режиме и интеграция с процессами CI/CD позволяет значительно сократить время на проверку качества приложения.

2. Случаи использования автоматизированного тестирования:

- При необходимости регулярного и повторяющегося тестирования, например, после каждого изменения в коде.

- Для тестирования крупных проектов с большим объемом функциональности.

- Для проверки функциональности, которая требует многократного ввода одних и тех же данных.

- В случаях, когда требуется проверять регрессию (не появились ли новые дефекты после внесенных изменений).

3. Недостатки автоматизированного тестирования:

- Высокие затраты на начальную настройку: Создание и поддержка автоматизированных тестов может потребовать значительных временных и финансовых ресурсов.

- Сложность поддержки тестов: При изменениях в интерфейсе приложения или логике тесты могут перестать работать корректно, требуя постоянного обновления.

- Необходимость частого обновления: При изменении функциональности приложения автоматизированные тесты также требуют обновления, что может быть трудоемким процессом.

- Ограничения в тестировании UI: Автоматизированные тесты UI могут быть нестабильными из-за изменений в CSS, JavaScript и других технологиях фронтенда.

4. Локатор:

- Локатор в контексте автоматизированного тестирования — это метод или средство, которое позволяет идентифицировать элементы пользовательского интерфейса на веб-странице или в приложении. Локатор используется для поиска и взаимодействия с элементами, такими как кнопки, поля ввода, ссылки и т.д. Примеры локаторов включают CSS-селекторы, XPath-выражения, ID элемента и т.д.

5. Цели нагрузочного тестирования:

- Определение производительности: Нагрузочное тестирование направлено на оценку производительности системы при определенных условиях нагрузки. Это позволяет выявить узкие места в архитектуре и конфигурации приложения.

- Выявление предельных значений: Тестирование позволяет определить, как система справляется с максимальными нагрузками и при каких условиях происходит ее выход на предельные значения.

- Проверка стабильности: Оценка стабильности работы системы под нагрузкой помогает убедиться в ее способности обслуживать большое количество пользователей одновременно без сбоев.

- Планирование масштабирования: Результаты нагрузочного тестирования используются для планирования необходимости и возможности масштабирования системы при росте числа пользователей или объема данных.