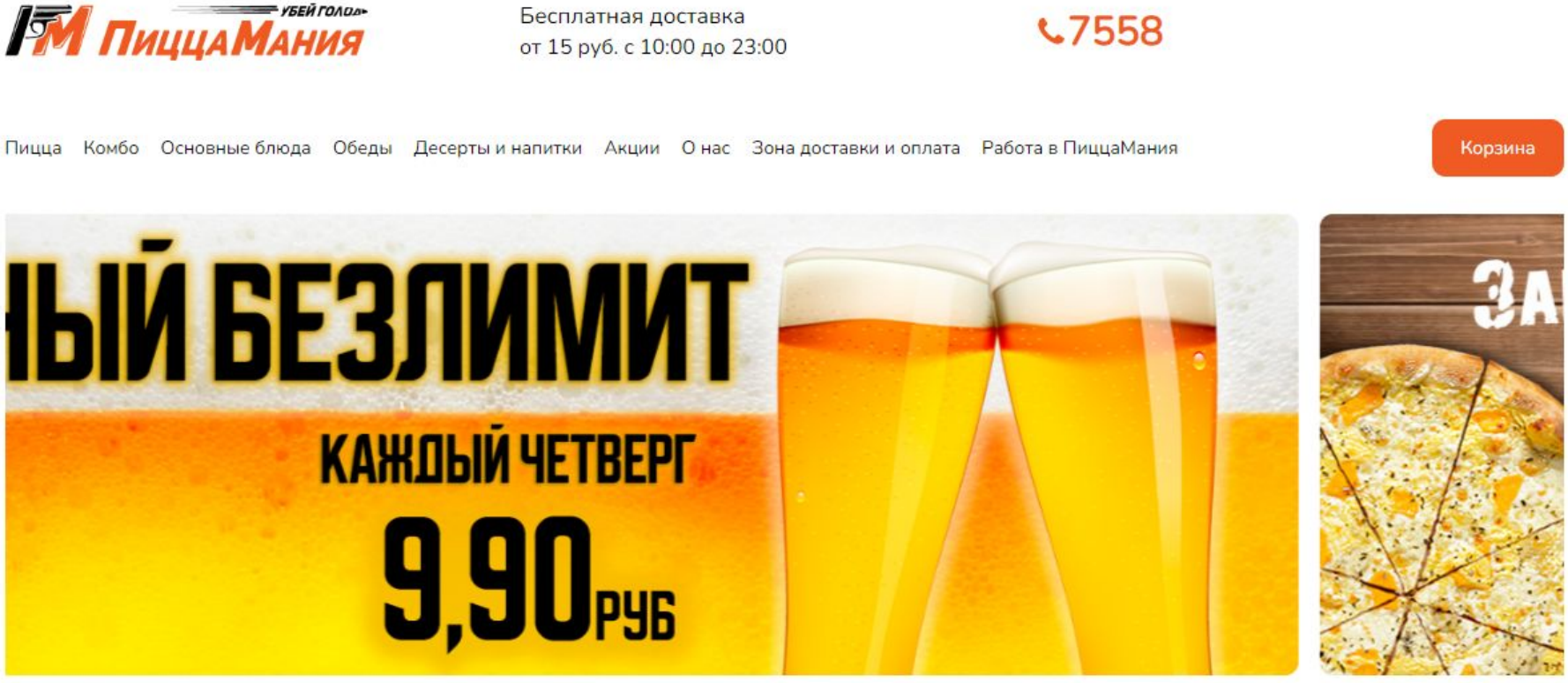
Тестирование сайта

ПиццаМания

(*https://pizzamania.by/*)



Задачи:

1. Зайти на домашнюю страницу сайта.

2. В разделе Пицца(-ы) добавить в Корзину пиццу Маргарита (или Четыре сезона) любого размера.

3. В разделе Напитки (Бар) добавить в Корзину любой напиток.

4. Перейти в Корзину.

5. Проверить, что пицца Маргарита (или Четыре сезона) есть в заказе.

6. Проверить, что выбранный напиток есть в заказе.

Архитектура решения:

Для реализации данного проекта был использован паттерн **Page Object**.

Page Object — паттерн проектирования, позволяющий скрыть реальное взаимодействие с элементами веб-страницы представить **API** для тестовых методов.

К основным преимуществам паттерна **Page Object** можно отнести то, что:

1.Внутренняя структура страницы(локаторы) скрыта.

2.Публичные методы представляют удобный **API** для работы с именно данной конкретной страницей.

3.Методы возвращают саму страницу, можно выстраивать цепочки вызовов.

4.Есть возможность описывать только те части страницы, которые нам нужны.

5.Все проверки выносятся в код теста, а низкоуровневые действия со страницей - в код **Page Object**.

Зависимости

1.junit, version:4.13.2

JUnit – это фреймворк, разработанный для тестирования программ, написанных с использованием технологии Java.

Свойства JUnit

- Фреймворк с открытым исходным кодом, который используется для написания и выполнения тестов.
- Позволяет писать код более быстро и качественно.
- Крайне прост в использовании.
- Поддерживает аннотации для идентификации методов.
- Поддерживает утверждения для тестирования получаемых результатов.
- Тесты могут быть организованы в связки тестов (test suites).
- Имеет визуальную индикацию состояния тестов (красные – не пройдены, зелёные – пройдены).

Зависимости

selenium-java, version:4.2.1

Selenium WebDriver — это инструмент, который позволяет производить кросс-браузерное тестирование, то есть проверять, как отображается сайт в разных браузерах.

Основные преимущества:

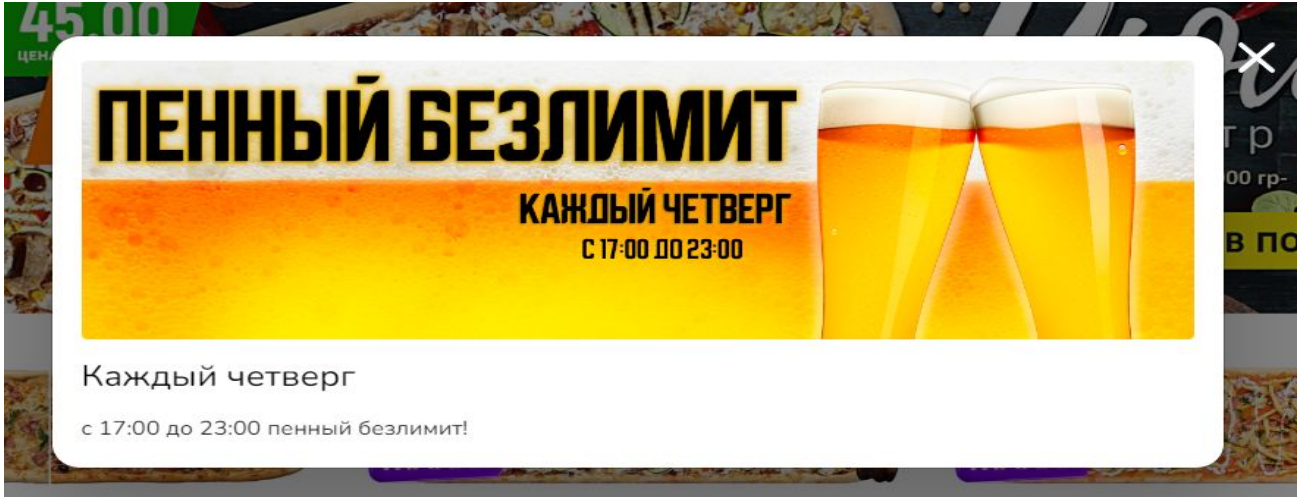
1. Поддерживает разные языки программирования (Java, C#, PHP, Ruby, Perl, Python), а значит, его может использовать большое количество разработчиков.
2. Облегчает кроссбраузерное тестирование и поддерживает различные браузеры: Firefox, Opera, Chrome, Edge.
3. Есть фреймворки, облегчающие разработку.
4. Позволяет проводить параллельное тестирование в нескольких браузерах одновременно.
5. Имеет открытый исходный код, он бесплатный для любого разработчика.
6. Большое сообщество пользователей — при возникновении трудностей в работе есть у кого попросить помощи.

Зависимости

webdrivermanager, version:5.2.0

WebDriverManager — это библиотека **Java** с открытым исходным кодом, которая осуществляет управление (т. е. загрузку, настройку и обслуживание) драйверами, необходимыми для **Selenium WebDriver** (например, chromedriver, geckodriver, msedgedriver и т. д.), полностью автоматизированным способом. Кроме того, начиная с версии 5, **WebDriverManager** предоставляет другие важные функции, такие как возможность обнаружения браузеров, установленных в локальной системе, создание объектов WebDriver (таких как ChromeDriver, FirefoxDriver, EdgeDriver и т. д.), беспрепятственный запуск браузеров в контейнерах Docker, и возможности мониторинга.

Интересный момент в работе над проектом



В процессе выполнения теста на наличие пиццы “Маргарита” и напитка в корзине, при открытии главной страницы сайта всплывает модальное окно рекламы с долгой загрузкой. В результате чего скрипт выполняется намного быстрее реакции приложения на команды. Поэтому для успешного выполнения теста нужно дождаться определенного состояния приложения для дальнейшего взаимодействия с ним. Что было реализовано использованием неявного ожидания - Implicit Waits:

```
public WebDriverUtil() {
    WebDriverManager.chromedriver().setup();
    driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait( Duration.ofSeconds(5));
    driver.manage().window().maximize();
}
```