

Step IT Academy s.r.o. Opletalova 1418/23, 110 00 Nové Město

Дипломная работа

Функциональное и нагрузочное тестирование сайта железнодорожной компании CD.cz

Автор: Olena Malinovska

Руководитель: Vitaliy Zamirovskiy

2025

Содержание

1. Актуальность исследования
2. Назначение и функции тестируемой программы
3. Архитектура тестируемой системы
4. Цель и задачи дипломной работы
5. Тестовая среда
6. Методология
7. Тест-кейсы и сценарии тестирования
8. Используемые инструменты тестирования
9. Чек лист
 - Функциональность
 - Юзабилити
 - Производительность
 - Безопасность
 - Контент
10. Практическая часть работы.
11. Заключение

1. Актуальность исследования

В повседневной жизни цифровизация транспортных услуг становится важнейшим направлением развития инфраструктуры. Железнодорожные перевозки активно внедряют онлайн-сервисы, которые позволяют пассажирам планировать маршруты, оформлять билеты и получать информацию о поездках в любое время. Надёжность и корректная работа таких сервисов напрямую влияет на удовлетворённость клиентов и имидж компании.

Железнодорожные связи со смежными странами Австрия, Германия, Польша и Словакия и не только. Сайт CD.cz является официальным онлайн-ресурсом Чешских железных дорог и выполняет функции, критически важные для пользователей: построение маршрутов, отображение расписаний, расчёт стоимости билетов и оформление проездных документов. Ошибки в этих функциях могут привести к неудобству, потерям времени или финансовым убыткам для пассажиров.

Проведение функционального и нагрузочного тестирования сайта CD.cz позволяет оценить его стабильность, точность расчётов и способность работать под высокой пользовательской активностью. Это особенно актуально в условиях сезонных пиков, изменений в расписании и других критических ситуаций, когда сервис должен оставаться доступным и корректным.

2. Назначение и функции тестируемой программы

Сайт CD.cz предназначен для предоставления пассажирам Чешских железных дорог актуальной информации о поездках, а также для самостоятельного планирования маршрутов и покупки билетов. Основные функции сайта включают:

- Поиск маршрутов между станциями с учётом времени и даты отправления;
- Отображение расписания движения поездов;
- Расчёт стоимости билета в зависимости от маршрута, категории пассажира, типа поезда и класса обслуживания;
- Покупку и бронирование билетов онлайн;

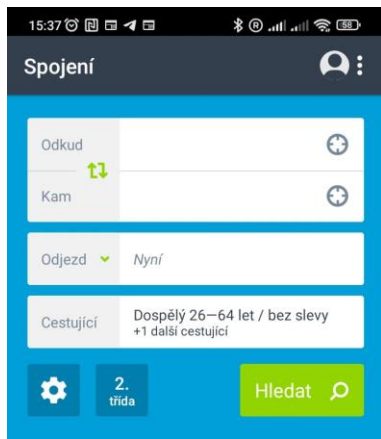
- Предоставление информации об акциях, скидках и специальных предложениях;
- Возможность связи с технической поддержкой и подачи заявлений через форму обратной связи;
- Раздел Fanshop — интернет-магазин с фирменной продукцией и сувенирами ČD.

Эти функции являются важной частью цифровой инфраструктуры транспортной компании и требуют регулярной проверки на корректность и устойчивость.

3. Архитектура тестируемой системы

Важно отметить, что веб-приложение CD.cz доступно на чешском и английском языках, что обеспечивает удобство для как местных, так и иностранных пользователей. Мобильное приложение поддерживает три языка: чешский, английский и немецкий, что расширяет его доступность для международных пассажиров.

Для наглядного представления интерфейсов сайта и мобильного приложения приведены соответствующие скриншоты.



Doporučená spojení

Praha - Brno



Рисунок 1 — Интерфейс мобильного приложения CD.cz

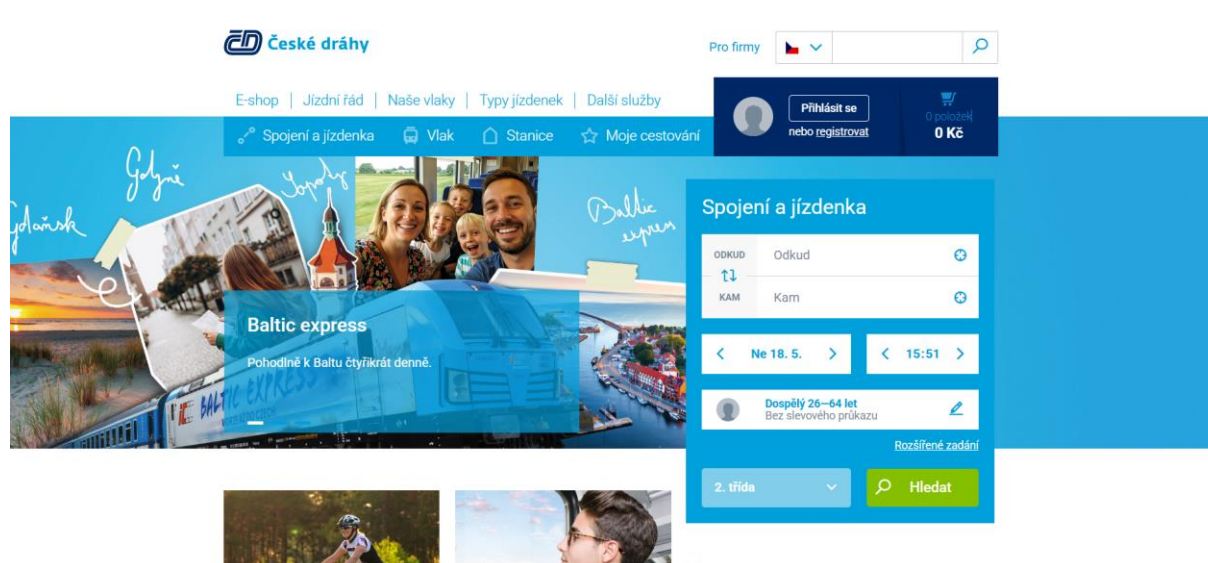


Рисунок 2 — Главная страница веб-сайта CD.cz

Кроме веб-сайта, компания ČD также предоставляет официальное мобильное приложение, доступное для платформ Android и iOS. Мобильное приложение предоставляет те же основные функции, что и сайт: поиск маршрутов, покупку билетов, отображение расписаний, персональный кабинет пользователя. Приложение предназначено для использования в пути и позволяет хранить электронные билеты без необходимости распечатки.

Сайт CD.cz представляет собой многоуровневую веб-систему с клиентской и серверной частью. Клиентская часть реализована с применением HTML, CSS, JavaScript и современных фреймворков, обеспечивающих динамическое взаимодействие пользователя с интерфейсом. Серверная часть обрабатывает маршруты, тарифы и бронирования, используя собственные алгоритмы и базы данных.

Основные компоненты архитектуры:

- **Пользовательский интерфейс (Frontend):** формы поиска маршрутов, выбора билетов, оформления заказа, а также формы обратной связи;
- **Сервер приложений (Backend):** логика обработки маршрутов, расчёта цен, проверка доступности мест, обработка форм;
- **База данных:** хранение информации о маршрутах, расписаниях, ценах, клиентах и операциях бронирования;

- **Сторонние сервисы:** системы оплаты, идентификация пользователя, API для отображения реального времени и погодных условий.

Такая архитектура требует комплексного тестирования как отдельных компонентов, так и их взаимодействия при нагрузке и реальной пользовательской активности.

4. Цель и задачи дипломной работы

Цель данной дипломной работы — провести функциональное и нагрузочное тестирование сайта железнодорожной компании CD.cz. В ходе работы были сформулированы следующие задачи:

- Проверить корректность построения маршрутов;
- Оценить работу сайта при высокой нагрузке;
- Оттестировать расчёт стоимости билета при разных параметрах.

5. Тестовая среда

Тестирование проводилось в среде, максимально приближённой к пользовательской, с использованием следующих конфигураций:

- **Операционные системы:** Windows 10 PRO 22H2;
- **Браузеры:** Google Chrome (версии 123–124), Mozilla Firefox (версия 124) ; Microsoft Edge (136.0.3240.76)
- **Разрешение экрана:** 1920x1080 (Full HD);
- **Устройства:** ноутбук DESKTOP-NLC59KB, смартфон Redmi 9C NFC с версией Android 11 RP1A200720.011;
- **Сетевое подключение:** проводное соединение 100 Мбит/с, Wi-Fi.

Для тестирования мобильной версии сайта применялись инструменты эмуляции в DevTools Chrome, а также реальные устройства Android. Автоматизированные тесты запускались в браузере Chrome с использованием последней версии ChromeDriver.

6. Методология

В процессе работы использовалась комбинированная методология, включающая как ручное, так и автоматизированное тестирование. Для ручного тестирования применялись чек-листы и тест-кейсы, основанные на пользовательских сценариях и основных функциях сайта. Для автоматизированного тестирования использовалась библиотека Selenium на языке Python, что позволило воспроизвести действия пользователей и проверить стабильность выполнения операций.

Проверка форм, маршрутов и расчёта стоимости билета выполнялась с учётом различных параметров: тип поездки, дата, возраст и статус пассажира. Тесты разрабатывались по принципам позитивного и негативного тестирования, включая граничные значения и ввод некорректных данных.

Для проверки отправки форм использовался инструмент Postman, позволяющий эмулировать HTTP-запросы, что дополняло тестирование клиентского интерфейса серверной валидацией. Результаты тестов документировались и при обнаружении отклонений оформлялись баг-репорты.

7. Тест-кейсы и сценарии тестирования

Для системной проверки ключевых функций сайта CD.cz были разработаны тест-кейсы. Они описывали входные данные, ожидаемые результаты и фактическое поведение системы. Основное внимание уделялось следующим категориям:

- Построение маршрута между станциями;
- Расчёт стоимости билета с учётом параметров (дата, пассажир, направление);
- Отправка формы обратной связи;
- Проверка поведения системы при вводе некорректных данных;
- Проверка скидок (например, детских, студенческих);
- Работа навигационного меню и внутренних ссылок.
- Тест-кейсы были оформлены в таблицах, содержали заголовок, шаги воспроизведения, тестовые данные, ожидаемый результат, статус прохождения и комментарии. Также были учтены позитивные и негативные сценарии, включая граничные значения. Примеры тест-кейсов приведены в приложении.

8. Используемые инструменты тестирования

- **Jira** — система управления проектами и отслеживания ошибок, использовалась для ведения баг-репортов, отслеживания статуса тестов и фиксации дефектов, найденных в процессе тестирования;
- **JMeter** — инструмент для нагрузочного тестирования, применялся для оценки производительности сайта при высокой пользовательской активности. Позволяет создавать сценарии с одновременным выполнением большого количества запросов и анализировать поведение сервера под нагрузкой;
- **Lighthouse** — инструмент от Google для оценки качества веб-страниц, включая производительность, доступность, SEO и соблюдение стандартов. Применялся для анализа скорости загрузки и выявления потенциальных узких мест в клиентской части;
- **Selenium WebDriver** — библиотека автоматизации браузера, использовалась для написания автотестов на языке Python. Позволила воспроизвести действия пользователей на сайте, включая ввод маршрутов, выбор даты, расчёт стоимости и отправку форм;
- **Google Chrome DevTools** — встроенные средства разработчика, применялись для анализа сетевых запросов, структуры DOM и отладки интерфейса сайта;
- **Средства ручного тестирования** — чек-листы, таблицы и сценарии, используемые для последовательной ручной проверки функциональности.
- **Postman** — это один из самых популярных инструментов для тестирования API
- **TestRail** — это профессиональный инструмент для организации, планирования и документирования тестирования
- **SSL Labs** - это инструмент анализа безопасности HTTPS-соединения
- **SecurityHeaders.com** — это онлайн-инструмент для проверки наличия и правильности настройки HTTP-заголовков безопасности
- **GitHub** — хостинг кода.
- **BrowserStack** — тестирование браузеров

9.Чек лист

Чек-лист для тестирования сайта — это инструмент, который помогает систематизировать и упорядочить процесс проверки веб-ресурса. Он позволяет избежать пропусков важных аспектов тестирования и обеспечивает более качественный результат. Использование чек-листа

особенно полезно для новичков таких как я, так как он помогает структурировать работу и не забыть о ключевых моментах.

Тестирование сайта включает в себя множество аспектов, начиная от функциональности и заканчивая безопасностью. Каждый из этих аспектов требует тщательной проверки, чтобы обеспечить пользователям наилучший опыт взаимодействия с ресурсом. Чек-лист помогает не только организовать процесс тестирования, но и сделать его более прозрачным и управляемым. Это особенно важно для команд, работающих над крупными проектами, где координация и четкость действий играют ключевую роль.

Основные элементы моего чек-листа:

Создание эффективного чек-листа требует понимания основных элементов, которые должны быть проверены. Вот несколько ключевых категорий:

Функциональность

Функциональность сайта — это его основная способность выполнять заявленные функции. Проверка функциональности включает в себя множество аспектов, таких как:

Навигация: Проверка всех ссылок и меню на сайте. Уверенность, что все ссылки работают корректно и ведут на нужные страницы. Навигация должна быть интуитивно понятной и удобной для пользователя.

Формы: Тестирование всех форм на корректность ввода и отправки данных. Проверка, что все поля формы работают правильно, и данные отправляются корректно. Это включает в себя проверку обязательных полей, форматов ввода и сообщений об ошибках.

Кнопки и ссылки: Проверка всех кнопок и ссылок на правильность работы. Уверенность, что все кнопки выполняют свои функции, а ссылки ведут на правильные страницы. Это особенно важно для кнопок, связанных с покупками или отправкой данных.

Юзабилити

Юзабилити — это удобство использования сайта. Хорошо спроектированный сайт должен быть интуитивно понятным и удобным для пользователя. В чек-лист включены следующие пункты:

Интерфейс: Оценка удобства использования сайта. Проверка, насколько легко пользователю найти нужную информацию и выполнить основные действия.

Доступность: Проверка доступности контента для всех пользователей, включая людей с ограниченными возможностями. Это включает в себя проверку совместимости с экранными читалками и другими вспомогательными технологиями.

Мобильная версия: Тестирование адаптивности сайта на различных устройствах. Нужно убедиться, что сайт корректно отображается и работает на смартфонах и планшетах.

Производительность

Производительность сайта — это его способность быстро и эффективно обрабатывать запросы пользователей.

Скорость загрузки: Измерение времени загрузки страниц. Проверка, насколько быстро загружаются страницы сайта. Это важно для удержания пользователей и улучшения SEO.

Стресс-тестирование: Проверка работы сайта под высокой нагрузкой. Уверенность, что сайт справляется с большим количеством одновременных пользователей и не падает под нагрузкой.

Безопасность

Безопасность сайта — это его способность защищать данные пользователей и предотвращать атаки. В чек-лист включены следующие пункты:

- SSL-сертификат: Проверка наличия и корректности SSL-сертификата. Убедитесь, что все страницы сайта защищены и данные передаются по защищенному каналу.

- Защита данных: Оценка методов защиты пользовательских данных. Проверка, что данные пользователей хранятся и передаются безопасно, и что сайт защищен от атак, таких как SQL-инъекции и XSS.

Контент

Контент сайта — это его текстовая и мультимедийная информация. В чек-лист включены следующие пункты:

- Орфография и грамматика: Проверка текстов на наличие ошибок.
Уверенность, что все тексты на сайте написаны грамотно и без ошибок.
- Медиафайлы: Тестирование загрузки и отображения изображений и видео.
Проверка, что все изображения и видео корректно загружаются и отображаются на всех устройствах.

10. Практическая часть работы.

Функциональное тестирование проводилось для проверки ключевых сценариев:

- Ввод пунктов отправления и прибытия;
- Выбор даты поездки;
- Отображение маршрутов;
- Отображение и расчёт цены билетов;
- Обратная связь через форму.

Тесты проводились ручными и автоматизированными средствами (библиотека Selenium).

Функциональное тестирование авторизации

Одним из ключевых элементов сайта является форма авторизации, предоставляющая доступ к личному кабинету пользователя. В рамках функционального тестирования была проверена корректность работы формы логина при вводе как валидных, так и невалидных данных.

Цель тестирования:

Проверить корректность обработки формы авторизации при различных сценариях ввода:

- при корректных данных;

- при ошибочном логине и/или пароле.

Тест кейс на корректный логин

TestRail | České dráhy

Add Search Working On Olena lion Admin

České dráhy

Project Overview
To-Do
Test Cases
Overview
Details
Tests & Results
Defects
History

Test Runs & Results
Milestones
Reports

In section České dráhy.

People & Status
Unassigned Change

Created
Olena lion

C6 Корректный логин

| | | | |
|--------------------|-------------------------|---------------------|------------------|
| Type Other | Priority High | Assigned To None | Estimate None |
| References None | Automation Type None | | |

Preconditions
Selenium WebDriver на языке Python, а также браузер Google Chrome

Steps
1. Ввод действующего логина и пароля.
2. Нажатие кнопки входа.

Expected Result
Пользователь перенаправляется в личный кабинет.
После входа открылась страница профиля, подтверждающая успешную авторизацию.

Код автоматизации

Ссылка на код автоматизации

```
import pytest
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.service import Service
from selenium.webdriver import ActionChains
from webdriver_manager.chrome import ChromeDriverManager
from config import URL, correct_login, correct_password
options = webdriver.ChromeOptions()
options.add_argument("--disable-blink-features=AutomationControlled")
options.add_argument("--window-size=1920,1080")

driver =
webdriver.Chrome(service=Service(ChromeDriverManager().install()),
options=options)
wait = WebDriverWait(driver, 10)
```

```

def test_correct_login():
    try:
        driver.get(URL)

        wait.until(EC.element_to_be_clickable((By.ID,
"consentBtnall"))).click()
        wait.until(EC.invisibility_of_element_located((By.ID,
"consentBtnall")))

        login_buttons = wait.until(
            EC.presence_of_all_elements_located((By.XPATH,
"//button[contains(@class, 'userbox--toggle')]"))
        )

        visible_login_button = next((b for b in login_buttons if
b.is_displayed()), None)

        if visible_login_button:
            actions = ActionChains(driver)

actions.move_to_element(visible_login_button).perform()
            print("Навели на видимую кнопку входа")

driver.execute_script("arguments[0].scrollIntoView({block:
'center'});", visible_login_button)
            visible_login_button.click()
            print("Кликнули на Přihlásit se")
            time.sleep(3)

            email_field =
wait.until(EC.visibility_of_element_located((By.ID, "emailId")))
            password_field =
wait.until(EC.element_to_be_clickable((By.ID, "passwordId")))

            email_field.clear()
            email_field.send_keys(correct_login)
            print("Ввели email")

```

```

        password_field.clear()
        password_field.send_keys(correct_password)
        print("Ввели правильный пароль")

        submit_buttons =
wait.until(EC.presence_of_all_elements_located(
            (By.CSS_SELECTOR, "button.btn.btn--blue.btn--
filled.btn--full-width")
        ))

        submit_btn = next((b for b in submit_buttons if
b.is_displayed()), None)

        if submit_btn:

driver.execute_script("arguments[0].scrollIntoView({block:
'center'});", submit_btn)
            driver.execute_script("arguments[0].click();",
submit_btn)

            print(" Клик по кнопке 'Přihlásit' выполнен")
            time.sleep(3)
        else:
            print(" Кнопка отправки входа не найдена!")

    except Exception as e:
        print(f" Ошибка во время выполнения теста: {e}")
    finally:
        driver.quit()

```

✓ Tests passed: 1 of 1 test – 12 sec 297 ms

correct_login.py::test_correct_login PASSED

[100%]Навели на видимую кнопку входа

Кликнули на Přihlásit se

Ввели email

Ввели правильный пароль

Клик по кнопке 'Přihlásit' выполнен

===== 1 passed in 17.82s =====

Process finished with exit code 0

Тест кейс на не корректный логин

The screenshot shows the TestRail web interface. On the left is a sidebar with navigation links: Project Overview, To-Do, Test Cases (selected), Overview, Details, Tests & Results, Defects, History, Test Runs & Results, Milestones, and Reports. Below these are sections for 'People & Status' and 'Created' with a user 'Olana lion'. The main area displays a test case titled 'Некорректный логин' (Incorrect login) with ID 'C7'. It includes a table with fields: Type (Other), Priority (High), Assigned To (None), and Estimate (None). Below the table are sections for Preconditions (Selenium WebDriver на языке Python, а также браузер Google Chrome), Steps (1. Ввод несуществующего e-mail и пароля. 2. Нажатие кнопки входа.), and Expected Result (Отображается сообщение об ошибке, например „Nesprávny e-mail nebo heslo“ (в переводе — «Неправильный e-mail или пароль»)).

Ссылка на код автоматизации

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.service import Service
from selenium.webdriver import ActionChains
from webdriver_manager.chrome import ChromeDriverManager
from config import URL, correct_login, incorrect_password

options = webdriver.ChromeOptions()
options.add_argument("--disable-blink-features=AutomationControlled")
options.add_argument("--window-size=1920,1080")
driver =
webdriver.Chrome(service=Service(ChromeDriverManager().install()),
options=options)
wait = WebDriverWait(driver, 10)

def test_incorrect_login():
    try:
        driver.get(URL)
```

```

        wait.until(EC.element_to_be_clickable((By.ID,
"consentBtnall"))).click()
        wait.until(EC.invisibility_of_element_located((By.ID,
"consentBtnall")))

        login_buttons = wait.until(
            EC.presence_of_all_elements_located((By.XPATH,
"//button[contains(@class,'userbox--toggle')]"))
        )

        visible_login_button = next((b for b in login_buttons if
b.is_displayed()), None)

        if visible_login_button:
            actions = ActionChains(driver)
            actions.move_to_element(visible_login_button).perform()
            print("Навели на видимую кнопку входа")

            driver.execute_script("arguments[0].scrollIntoView({block:
'center'}));", visible_login_button)
            visible_login_button.click()
            print("Кликнули на Přihlásit se")
            time.sleep(3)

            email_field =
wait.until(EC.visibility_of_element_located((By.ID, "emailId")))
            password_field =
wait.until(EC.element_to_be_clickable((By.ID, "passwordId")))

            email_field.clear()
            email_field.send_keys(correct_login)
            print("Ввели email")

            password_field.clear()
            password_field.send_keys(incorrect_password)
            print("Ввели неправильный пароль")

            submit_buttons =
wait.until(EC.presence_of_all_elements_located(
                (By.CSS_SELECTOR, "button.btn.btn--blue.btn--
filled.btn--full-width")

```



```

    ))

    submit_btn = next((b for b in submit_buttons if
b.is_displayed()), None)

    if submit_btn:

driver.execute_script("arguments[0].scrollIntoView({block:
'center'});", submit_btn)
        driver.execute_script("arguments[0].click();",
submit_btn)

        print(" Клик по кнопке 'Přihlásit' выполнен")
        time.sleep(3)
    else:
        print(" Кнопка отправки входа не найдена!")

    parts = ['nesprávný', 'příliš velkého', 'uzamčen']
    xpath = "//*[" + " or ".join([f"contains(text(), '{p}')" for p
in parts]) + "]"
    error_blocks = driver.find_elements(By.XPATH, xpath)

    print(f" Всего найдено элементов с ошибками:
{len(error_blocks)}")
    for block in error_blocks:
        print("-", repr(block.text))

    expected_errors = [
        "Zadali jste nesprávný e-mail nebo heslo.",
        "Z důvodu příliš velkého počtu chybných přihlášení byl
účet uzamčen na 60 minut."
    ]

    found = False
    for block in error_blocks:
        text = block.text.strip()
        for expected in expected_errors:
            if expected in text and block.is_displayed():
                found = True
                print(" Найдено сообщение об ошибке:", expected)
                break

```

```

        if not found:
            driver.save_screenshot("error_screen.png")
            print(" Скриншот сохрaнён: error_screen.png")

        assert found, " Ни одно из ожидаемых сообщений об ошибке не
найдено!"

    except Exception as e:
        print(f" Ошибка во время выполнения теста: {e}")
    finally:
        driver.quit()

```

```

✓ Tests passed: 1 of 1 test – 12 sec 909 ms

===== test session starts =====
collecting ... collected 1 item

incorrect_login.py::test_incorrect_login PASSED [100%]Навели на видимую кнопку входа
Кликнули на Přihlásit se
Ввели email
Ввели неправильный пароль
Клик по кнопке 'Přihlásit' выполнен
Всего найдено элементов с ошибками: 1
- 'Zadali jste nesprávný e-mail nebo heslo.'
Найдено сообщение об ошибке: Zadali jste nesprávný e-mail nebo heslo.

```

На основании выполненных тестов можно сделать вывод, что функциональность формы авторизации реализована корректно:

- при ошибочном вводе система сообщает о некорректных данных;
- при правильном вводе осуществляется успешный вход в систему.

Форма прошла функциональную проверку как на **корректную**, так и на **некорректную авторизацию**, что подтверждает её соответствие базовым требованиям.

Функциональное тестирование формы регистрации (cd.cz)

Проверить корректность валидации пользовательских данных в форме регистрации на сайте www.cd.cz, в частности, проверку на допустимые символы в полях «Имя» и «Фамилия».

Тест кейс

TestRail

České dráhy

Add

Search

Working On

Olena Iion

Admin

České dráhy

Details

Tests & Results

Defects

History

Test Runs & Results

Milestones

Reports

In section České dráhy.

People & Status

Unassigned Change

Created

Olena Iion

5/20/2025 7:22 PM

Updated

Olena Iion

5/21/2025 8:29 PM

C1 Функциональное тестирование формы регистрации

České dráhy

Successfully updated the test case.

| | | | | | | | |
|------------|-------|-----------------|--------|-------------|------|----------|------|
| Type | Other | Priority | Medium | Assigned To | None | Estimate | None |
| References | None | Automation Type | None | | | | |

Preconditions

URL <https://www.cd.cz/profil-uzivatele/registerce/> открыто в браузере

Steps

1. Перейти на страницу регистрации.

2. Ввести следующие значения:

Имя: 123!@#

Фамилия: 456\$\$%

Email: mail2319@gmail.com

Отправить форму регистрации.

Дождаться ответа от сервера и сообщения на экране.

Поставить галочку в соглашении пользователя

Отправить форму.

Expected Result

Сервер и/или клиентская часть должны отклонить регистрацию с сообщением: «Недопустимые символы в полях Имя и Фамилия. Допустимы только буквы.»

4 ?

[Ссылка на баг репорт](#)

- Нарушению структур целостности БД
- Ошибкам отображения имён
- Потенциальным XSS-уязвимостям при незранированных вводах

Был выполнен ручной функциональный тест регистрации. В результате найден баг, демонстрирующий отсутствие валидации входных данных. Попытка отследить регистрацию через API или DevTools показала, что запрос либо отправляется через JavaScript, либо уходит через внутреннюю защищённую систему. Тем не менее, визуальное поведение сайта и сетевой отклик подтверждают успешную регистрацию, несмотря на некорректный ввод.

Автоматизированное тестирование формы обратной связи

В рамках функционального тестирования сайта было проведено автоматизированное тестирование формы обратной связи. Первоначально планировалось протестировать основную форму по адресу:

<https://www.cd.cz/kontaktni-formular/?openform&t=D>

Однако в данной форме используется CAPTCHA, которая препятствует автоматической отправке формы с помощью инструментов автоматизации. В связи с этим было принято решение использовать альтернативную форму, доступную в разделе **eshop ČD**.

Проверить корректность работы формы:

- наличие обязательных полей;
- отображение ошибок при пустом вводе;
- успешная отправка формы при корректном заполнении.

[Ссылка на код автоматизации](#)

```
import pytest
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.service import Service
```

```
from selenium.webdriver.chrome.options import Options

@pytest.fixture
def driver():
    options = Options()
    options.add_argument("--window-size=1920,1080")
    service = Service()
    driver = webdriver.Chrome(service=service, options=options)
    yield driver
    driver.quit()

def test_send_contact_form(driver):
    driver.get("https://www.cd.cz/fanshop/kontaktni-formular")

    wait = WebDriverWait(driver, 10)

    try:
        cookie_btn = wait.until(
            EC.element_to_be_clickable((By.XPATH,
            "//a[@class='dm-cookie-popup-accept-cookies']"))
        )
        cookie_btn.click()
    except:
        print("Cookie knopka ne отображается или уже принята.")

    email_input =
wait.until(EC.presence_of_element_located((By.ID, "email")))
    email_input.clear()
    email_input.send_keys("test@example.com")

    message_input = driver.find_element(By.ID, "contactform-
message")
    message_input.send_keys("Toto je testovací zpráva pro
odeslání přes Selenium.")
```

```

checkbox = driver.find_element(By.ID,
"psgdp_r_consent_checkbox_2")
if not checkbox.is_selected():
checkbox.click()

submit_btn = driver.find_element(By.XPATH,
'//input[@value="Poslat"]')
submit_btn.click()

try:
confirmation = WebDriverWait(driver, 10).until(
EC.presence_of_element_located((By.XPATH,
'//li[contains(text(), 'úspěšně odeslán')]'))
)
assert "úspěšně odeslán" in confirmation.text.lower()

print("Форма успешно отправлена.")
except Exception as e:
driver.save_screenshot("form_error.png")
pytest.fail(f" Ошибка при проверке успешной отправки:
{e}")

```

```

✓ Tests passed: 1 of 1 test – 4 sec 328 ms

===== test session starts =====
collecting ... collected 1 item

forma.py::test_send_contact_form PASSED [100%]Форма успешно отправлена.

===== 1 passed in 10.61s =====

Process finished with exit code 0

```

- Заполнение всех обязательных полей формы (имя, e-mail, тема обращения, текст сообщения).
- Проверка, что при незаполненных полях отображаются ошибки.
- Проверка, что кнопка отправки активируется при заполнении всех обязательных полей.
- Проверка, что после отправки отображается сообщение об успешной отправке.

Форма в eShop прошла проверку на соответствие функциональным требованиям. Использование CAPTCHA на основной форме делает невозможным её автоматическое тестирование, что подтверждает актуальность выбора альтернативного варианта. Таким образом, были протестированы основные функции формы обратной связи в реальном сценарии.

Функциональное тестирование поиска маршрута

Один из ключевых функционалов сайта железных дорог — это система поиска маршрутов, которая позволяет пользователям находить доступные поездки между заданными станциями с учётом времени отправления, пересадок и дополнительных параметров .

Необходимо проверить, что функция поиска маршрутов работает корректно при различных вариантах ввода, а именно:

- система правильно находит маршруты между указанными станциями;
- фильтры и параметры учитываются при поиске;
- при некорректных или пустых данных отображаются соответствующие предупреждения;
- после поиска отображается таблица с результатами (время, поезд, пересадки и т.д.).

Тестовые параметры:

- Проверка валидности ввода (станции существуют).
- Проверка динамической подстановки (автозаполнение названий).
- Отображение ошибок при отсутствии маршрутов.
- Корректность сортировки маршрутов по времени.

[Ссылка на код автоматизации](#)

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.keys import Keys
```



```
import time
from config import URL

driver = webdriver.Chrome()
driver.get(URL)
wait = WebDriverWait(driver, 10)
driver.find_element(By.ID, 'consentBtnall').click()

time.sleep(3)

from_input = driver.find_element(By.ID, "connection-from")
from_input.send_keys("Ostrava")

to_input = driver.find_element(By.ID, "connection-to")
to_input.send_keys("Karlovy Vary")

calendar_input = wait.until(EC.element_to_be_clickable((By.NAME,
"calendar")))
calendar_input.click()
date_input = wait.until(EC.presence_of_element_located((By.NAME,
"calendar")))
date_input.clear()
date_input.send_keys("18/06/2025")
date_input.send_keys(Keys.TAB)

time_input =
wait.until(EC.presence_of_element_located((By.CSS_SELECTOR,
"input.timepicker")))
time_input.clear()
time_input.send_keys("15:30")
time_input.send_keys(Keys.TAB)

search_button = driver.find_element(By.CLASS_NAME, "btn.btn--
filled.btn--green.btn--with-icon.search-btn")
search_button.click()

time.sleep(5)
```

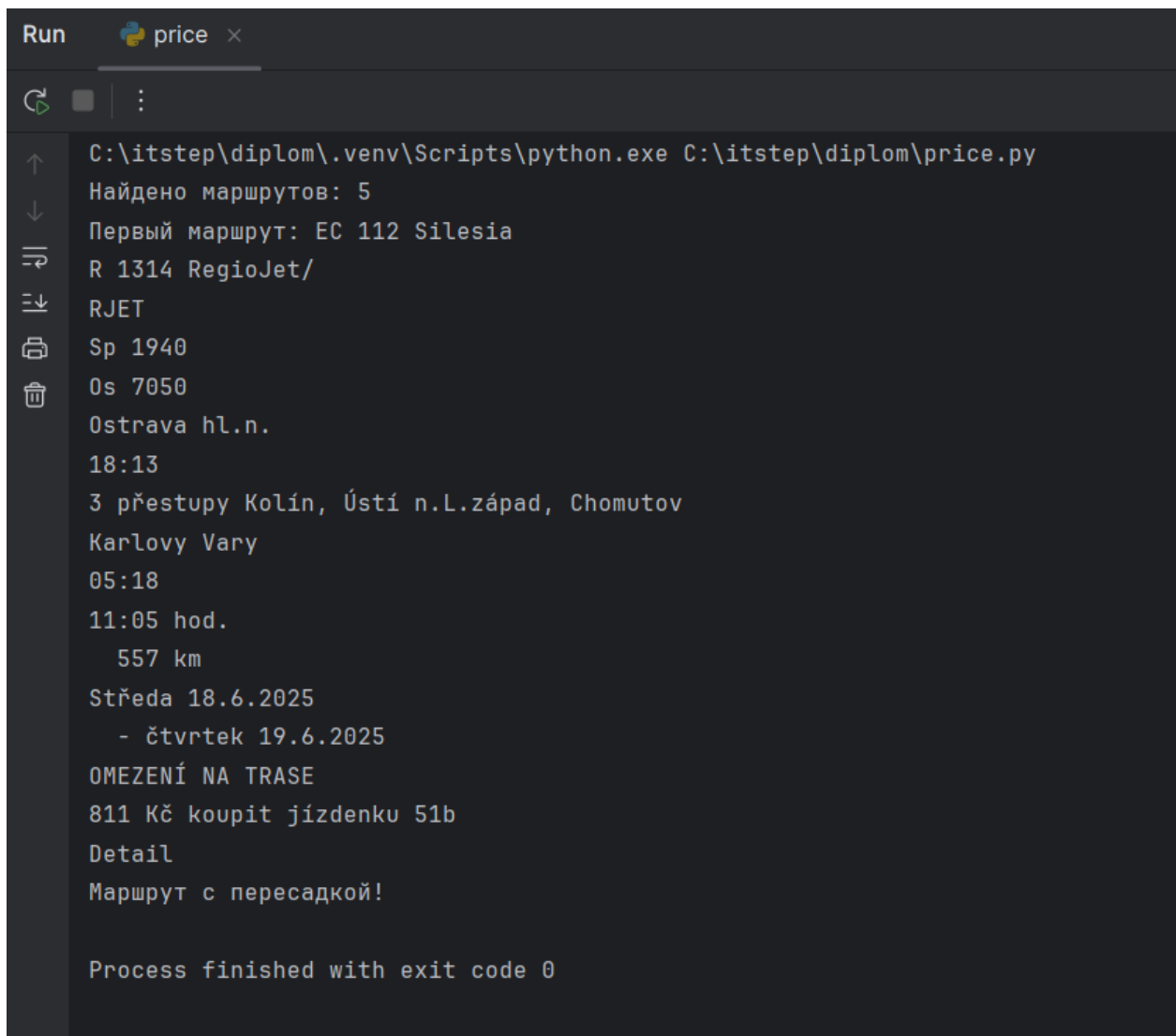
```
routes = driver.find_elements(By.CLASS_NAME, "overview-connection")

if routes:
    print(f"Найдено маршрутов: {len(routes)}")

    first_route = routes[0].text
    print("Первый маршрут:", first_route)

    if "přestup" in first_route.lower():
        print("Маршрут с пересадкой!")
    else:
        print("Прямой маршрут!")
else:
    print("Маршруты не найдены.")

driver.quit()
```



```
Run price x
C:\itstep\diplom\.venv\Scripts\python.exe C:\itstep\diplom\price.py
Найдено маршрутов: 5
Первый маршрут: EC 112 Silesia
R 1314 RegioJet/
RJET
Sp 1940
Os 7050
Ostrava hl.n.
18:13
3 přestupy Kolín, Ústí n.L.západ, Chomutov
Karlovy Vary
05:18
11:05 hod.
557 km
Středa 18.6.2025
- čtvrtek 19.6.2025
OMEZENÍ NA TRASE
811 Kč koupit jízdenku 51b
Detail
Маршрут с пересадкой!

Process finished with exit code 0
```

Функция поиска маршрутов является критически важной для пользователей. Проведённое тестирование подтверждает, что она работает корректно при типовых сценариях. Проверки показали устойчивость к ошибочному вводу, наличие логичных сообщений об ошибках и надёжную генерацию списка маршрутов при валидном запросе.

Функциональное тестирование добавления товаров в корзину

Раздел eShop сайта ČD позволяет пользователям приобретать сопутствующие товары и сувениры. Один из ключевых процессов интернет-магазина — это корректная работа корзины: добавление, отображение и пересчёт товаров. От надёжности этой функции зависит успешность завершения покупки.

Тест написан на проверку, что корзина работает корректно и товар добавляется в корзину.

[Ссылка на код автоматизации](#)

```
from selenium import webdriver
import time
import random
from selenium.common import TimeoutException
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

options = webdriver.ChromeOptions()
options.add_argument(
    "--disable-blink-features=AutomationControlled")
options.add_argument("--window-size=1920,1080")

driver =
webdriver.Chrome(service=Service(ChromeDriverManager().install()),
options=options)

def add_to_cart_test():
    driver.get("https://www.cd.cz/fanshop/")

    time.sleep(3)

    try:
        cookie_btn = WebDriverWait(driver, 10).until(
            EC.element_to_be_clickable((By.XPATH, "//a[@class='dm-
cookie-popup-accept-cookies']")))
        cookie_btn.click()
    except TimeoutException:
        print("Куки уже приняты или кнопка не найдена.")
        time.sleep(3)

    try:
        categories = WebDriverWait(driver, 10).until(
            EC.presence_of_all_elements_located((By.XPATH,
'//li[@class="nav-item parent dropdown leo-1"]')))

        if categories:
```

```

        random_category = random.choice(categories)
        random_category.click()
        print("Перешли в рандомную категорию.")
    else:
        print("Категории не найдены.")
        driver.quit()
        return

added_products = 0
total_product_count = 5

time.sleep(3)

while added_products < total_product_count:
    add_to_cards_buttons = WebDriverWait(driver, 10).until(
        EC.presence_of_all_elements_located((By.XPATH,
'//span[@class="leo-bt-cart-content"]'))
    )

    for add_to_card_button in add_to_cards_buttons:
        if added_products >= total_product_count:
            break

        try:
            time.sleep(1)

driver.execute_script("arguments[0].scrollIntoView({block:
'center'});",
                                add_to_card_button)
            driver.execute_script("arguments[0].click();",
add_to_card_button)
            time.sleep(3)

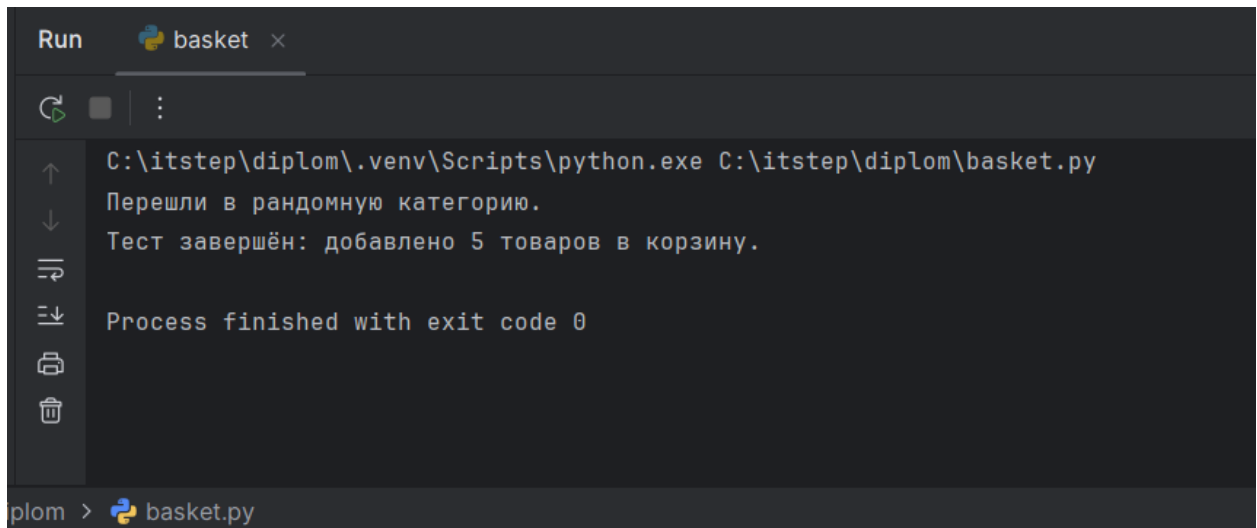
            added_products += 1
        except Exception:
            continue

    print(f"Тест завершён: добавлено {added_products} товаров в
корзину.")

```

```
except Exception as e:
    print(f"Произошла ошибка: {e}")
finally:
    driver.quit()

add_to_cart_test()
```



В процессе автоматизации использовался скрипт на Python, который:

- автоматически переходит в eShop;
- выбирает доступный товар;
- имитирует нажатие кнопки добавления в корзину;
- проверяет наличие элемента в корзине

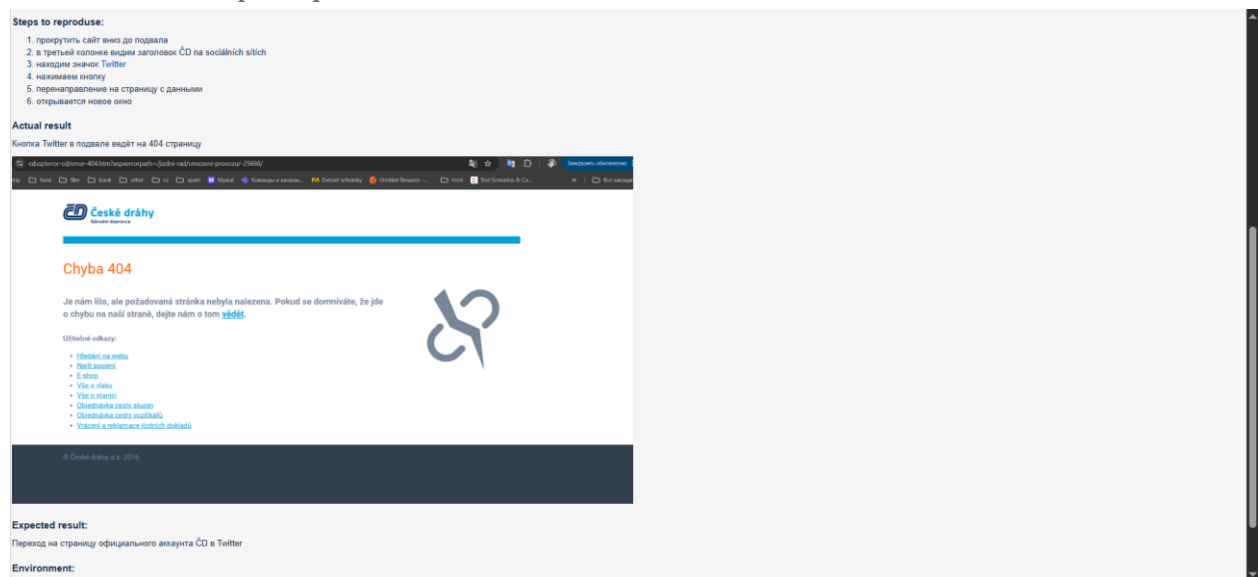
Проверка корзины подтверждает корректную реализацию e-commerce-функциональности на сайте. Пользовательский путь от выбора до добавления товара работает надёжно, отображение позиций соответствует ожидаемому поведению.

Неработающие ссылки на сайте ČD

В процессе ручного тестирования навигации сайта www.cd.cz были обнаружены **битые и неработающие ссылки** как во внутреннем контенте, так и в элементах главного меню.

Вот одна из них, это ссылка на иконке Twitter, размещённой в футере сайта www.cd.cz, ведёт на несуществующую страницу, в результате чего отображается ошибка 404.

[Ссылка на баг репорт](#)



При анализе кода страницы через инструменты разработчика было выявлено, что элемент футера, визуально обозначенный как ссылка на Twitter, содержит некорректный внутренний путь (/aplikace/omezeni-provozu/-25698/), не связанный с социальными сетями.

В результате пользователь, ожидая переход на официальный аккаунт ЪД в Twitter, попадает на нерелевантную или несуществующую страницу, что является **функциональным и навигационным дефектом**.

Юзабилити.

Во время ручного тестирования главной страницы сайта был зафиксирован **дефект отображения погодных данных**: секция, предназначенная для вывода информации о текущей погоде, присутствует в верстке, однако фактические данные не подгружаются.

Это можно классифицировать как **функциональный дефект**, так как элемент интерфейса не выполняет свою заявленную функцию. Одновременно, **с точки зрения юзабилити**, отсутствие погодной информации может затруднить планирование поездок пользователями, особенно при дальних маршрутах, и снизить общее удобство использования сайта.

Убедиться, что погодная информация (температура, погодные условия, иконка) корректно отображается на главной странице сайта, если она запрашивается через API.

Тест кейс ссылка

The screenshot shows the TestRail web interface for a project named "České dráhy". The left sidebar contains navigation links: "České dráhy", "Project Overview", "To-Do", "Test Cases" (with sub-links for Overview, Details, Tests & Results, Defects, and History), "Test Runs & Results", "Milestones", and "Reports". Below these are "People & Status" (Unassigned, Change) and a "Created" section listing "Olana lion".

The main content area displays a test case titled "C2 Отображение погодных данных на сайте". A green notification bar at the top states "Successfully added the new test case. Add another". Below this is a table with the following data:

| Type | Priority | Assigned To | Estimate |
|------------|-----------------|-------------|----------|
| Other | Medium | None | None |
| References | Automation Type | | |
| None | None | | |

The test case details include:

- Preconditions:** Открыт сайт <https://www.cd.cz> в браузере.
- Steps:**
 1. Открыть веб-браузер.
 2. Перейти на главную страницу сайта <https://www.cd.cz>.
 3. Проверить наличие видимого блока с погодной информацией (текст, иконка, температура, город).
 4. Если блок есть — сравнить отображаемые данные с API-ответом (например, используя Postman).
- Expected Result:**
 - Блок с актуальной погодой отображается.
 - Данные (температура, иконка, условия) соответствуют полученным из API (например, temperature: 16.4°C, weathercode: 3 → облачность).
 - При отсутствии данных с API — отображается fallback/заглушка ("Данные временно недоступны").

A green question mark icon with the number "4" is located in the bottom right corner of the test case details area.

Ссылка на баг репорт

Back to previous view

[SCRUM-15] Погода не отображается на главной странице Created: 19/May/25 Updated: 21/May/25

Status:To Do

Project:My Scrum Project

Components:None

Affects versions:None

Fix versions:None

Type:Bug

Reporter:Olga Malinova

Resolution:Unresolved

Labels:None

Remaining Estimate:Not Specified

Time Spent:Not Specified

Original estimate:Not Specified

Priority:Low

Assignee:Unassigned

Votes:0

Attachments:50 PNG

Rank:000030

Sprint:SCRUM Sprint 1

Severity:Trivial

Description

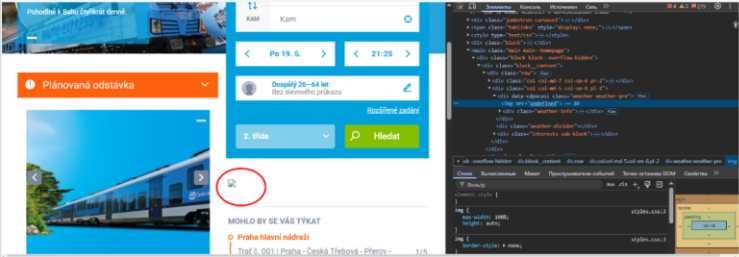
Preconditions: <https://www.cd.cz/> открыт в браузере

Steps to reproduce:

1. Открыть страницу сайта
2. Проверить наличие видимого блока с погодной информацией (текст, иконка, температура, город).
3. Проверить, отображается ли иконка погоды и температура.

Actual result

Иконка погоды не отображается, температура не отображается, API-ответ приходит с корректными погодными данными (код 200).



Expected result:

Должна быть видна иконка с погодой (например, солнце, облака), отображаться температура

Environment:

Windows 10, Google Chrome

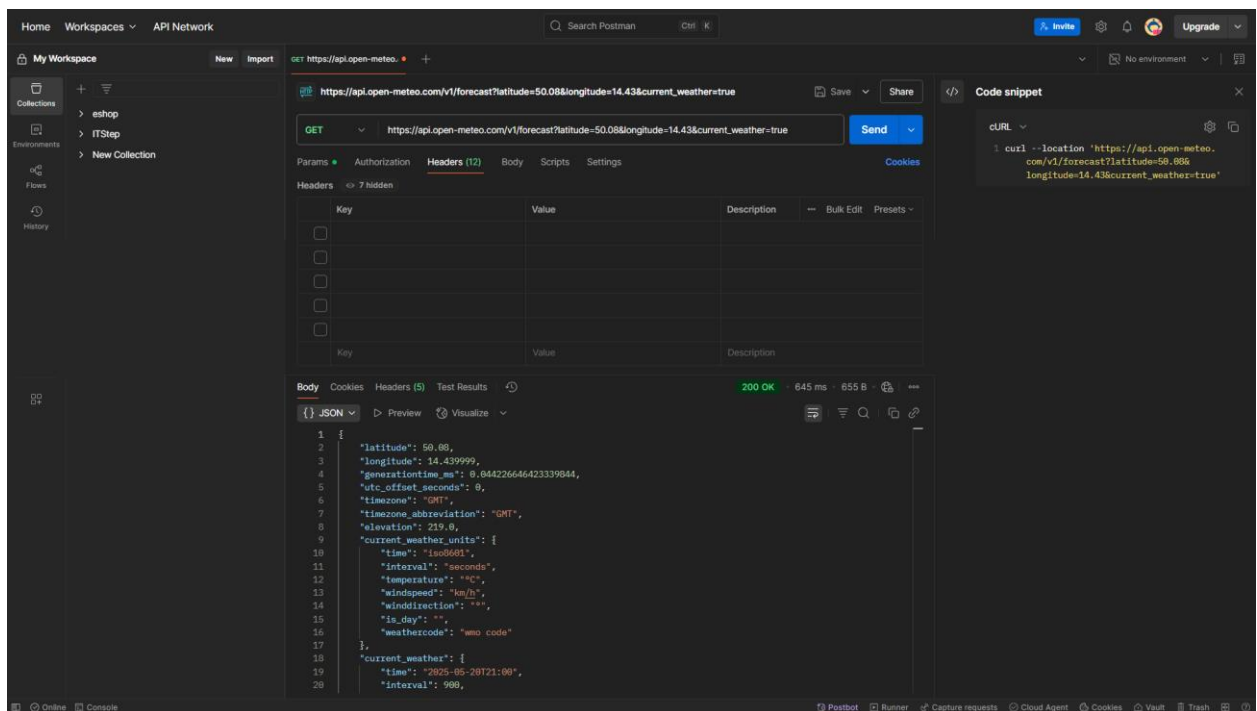
- Версия 135.0.7049.42

Generated at Wed May 21 21:30:27 UTC 2025 by Olga Malinova using Jira 1001.0.0-SNAPSHOT#100283-rev.2f8e83cea41399d10a38a401252998a899d4a41.

Описание: HTML-разметка содержит блок для отображения погоды (.weather-temperature), однако фактическая температура и другие данные отсутствуют. В атрибуте title установлено значение 'undefined'. При этом , прямой вызов API для получения погодных данных возвращает корректный JSON-ответ.

С помощью инструмента Postman был выполнен GET-запрос к открытому API Open-Meteo. В параметрах запроса были указаны координаты города Прага (широта: 50.08, долгота: 14.44). API возвращает информацию о текущей погоде в формате JSON.

Ссылка <https://.postman.co/workspace/My-Workspace~17e52269-83fb-487c-81cd-2cd9a875a6d9/request/39204869-f668c021-7b34-45cc-aa23-59d4bd16923a?action=share&creator=39204869&ctx=documentation>



На основе анализа можно сделать вывод, что проблема не связана с API, так как он возвращает корректные данные. Скорее всего, ошибка происходит при обработке данных на стороне фронтенда сайта. Блок погоды существует в DOM, но не заполняется содержимым, что указывает на потенциальный баг в JavaScript или отсутствии обработки ответа от API.

В процессе ручного тестирования RSS-ленты сайта ČD был зафиксирован дефект отображения символов, связанный с некорректной кодировкой, я бы так же отнесла этот дефект и к юзабилити. Текст ленты содержит искажённые символы вместо чешских букв, что делает контент трудночитаемым.

Это классифицируется как функциональный и совместимый дефект, поскольку:

- RSS является частью публичного сервиса сайта;
- он не может быть корректно прочитан в стандартных RSS-читалках и браузерах;
- нарушается восприятие информации конечным пользователем.

[Ссылка на баг репорт](#)

Скрин актуального результата

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0">
  <channel>
    <title>Řesek dráhy</title>
    <link>http://www.cd.cz/link</link>
    <description>RSS 2.0</description>
    <language>cs</language>
    <ttl>
    <managingEditor>
    <webMaster>
    <pubDate>15 May 2025 03:00:04 +0100</pubDate>
    <lastBuildDate>15 May 2025 03:00:04 +0100</lastBuildDate>
    <item>
      <guid>40518</guid>
      <title>Novinky u vlaků linky Praha - Berlín v letní sezóně</title>
      <description><p><strong>Vlakem do Německa na jaře a v létě: Seznámte se s vlaky EC z Prahy do Berlína a cestování jen s rezervacemi.</strong></p></description>
      <link>http://www.cd.czhttps://ceskedrahy.cz/tiskove-centrum/tiskove-zpravy/vlakem-do-nemecka-na-jare-v-lete-sezonni-par-vlaku-ec-z-prahy-do/link</link>
      <author>
        Řesek dráhy
      </author>
      <source>new.cd.cz</source>
      <pubDate>06 May 2025 11:29:26 +0100</pubDate>
    </item>
    <item>
      <guid>40517</guid>
      <title>Pochod Praha â€˜ Práche</title>
      <description><p><strong>Obujte toulavé boty a vyhlášte si do Práche!</strong></p></description>
      <link>http://www.cd.czhttps://www.ceskedrahy.cz/tiskove-centrum/tiskove-zpravy/ceske-drahy-posili-dopravu-na-pochod-praha-prcice-nabidnou-0</link>
      <author>
        Řesek dráhy
      </author>
      <source>new.cd.cz</source>
      <pubDate>06 May 2025 09:01:19 +0100</pubDate>
    </item>
  </channel>
</rss>
```

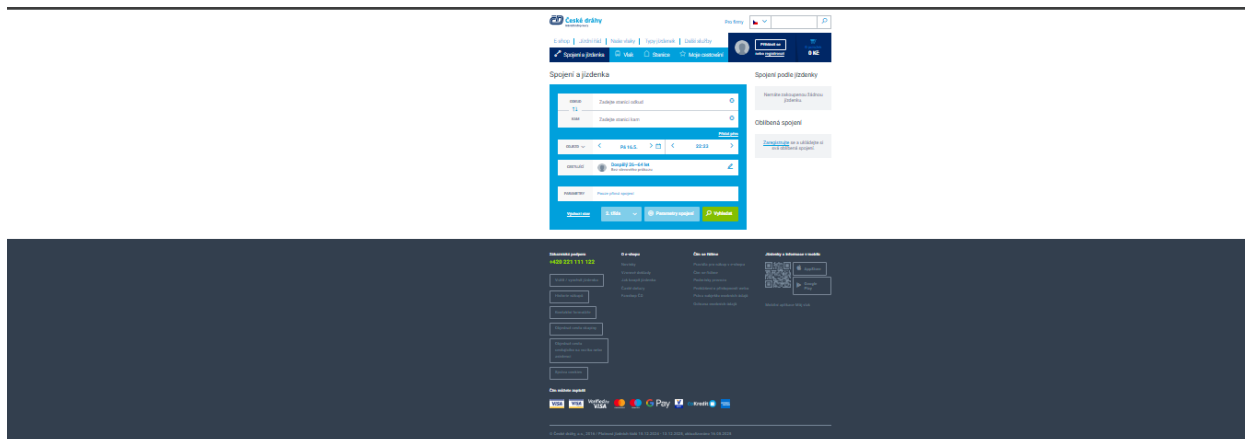
Юзабилити и нефункциональные замечания

В ходе дополнительного осмотра сайта были обнаружены отдельные страницы с отсутствующим содержимым. Хотя они не влияют напрямую на основные сценарии (поиск маршрута, покупка билета, авторизация), такие страницы могут снижать общее восприятие качества ресурса и дезориентировать пользователей.

Баг репорт [ссылка](#)

Следующий баг из этой же категории **Нефиксированный подвал** (footer), который поднимается к основному контенту при недостатке его высоты, — это **нарушение визуального и структурного стандарта интерфейса**.

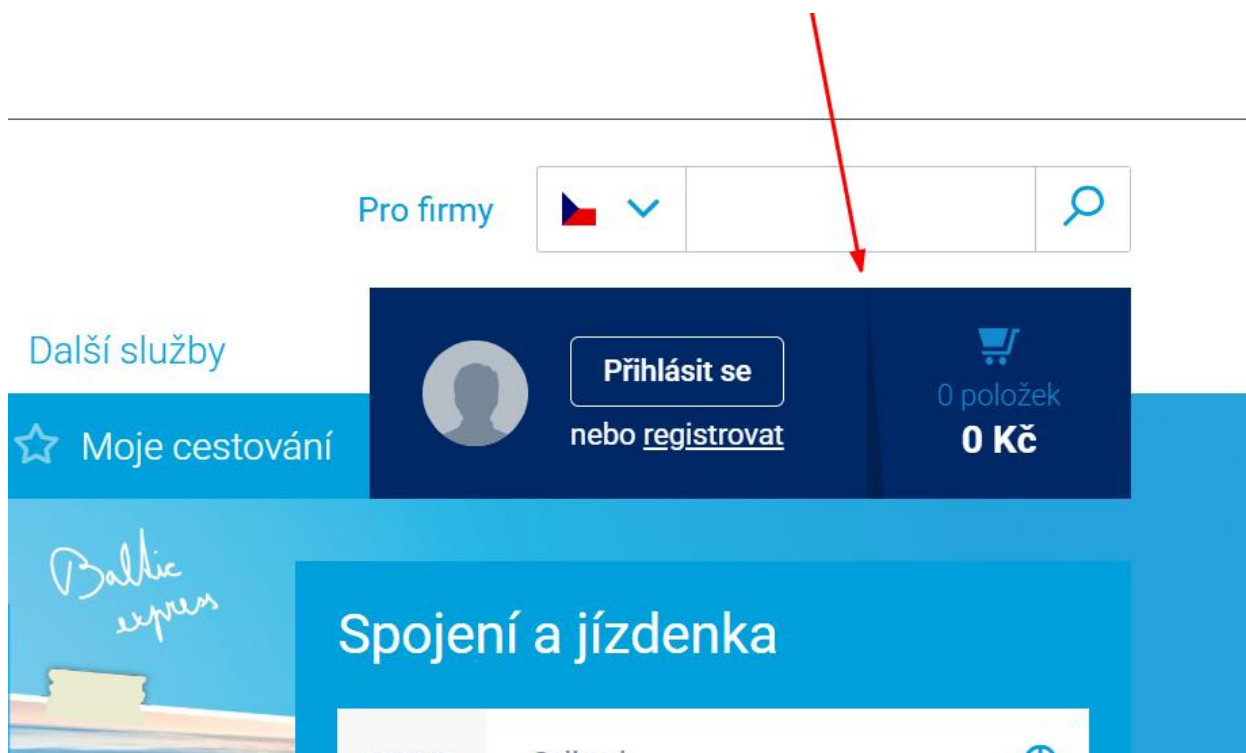
Баг репорт [ссылка](#)



При ручном тестировании отображения страницы **Spojení a jízdenka | České dráhy** было выявлено, что футер сайта не фиксируется в нижней части окна браузера при недостатке основного контента. Это приводит к тому, что нижняя часть интерфейса визуально поднимается вверх, нарушая баланс макета и создавая ощущение незавершённости страницы. Это баг уровня HTML/CSS — отсутствие или неправильное применение `position: fixed` или `flex-структуры`.

Следующий баг репорт будет посвящен тому что на главной странице сайта České dráhy, в верхней части страницы, между иконкой корзины и кнопкой входа появляется диагональная наклонная тёмная полоса. Она выглядит как графический артефакт или элемент разметки, не предусмотренный дизайном.

[Баг репорт ссылка](#)



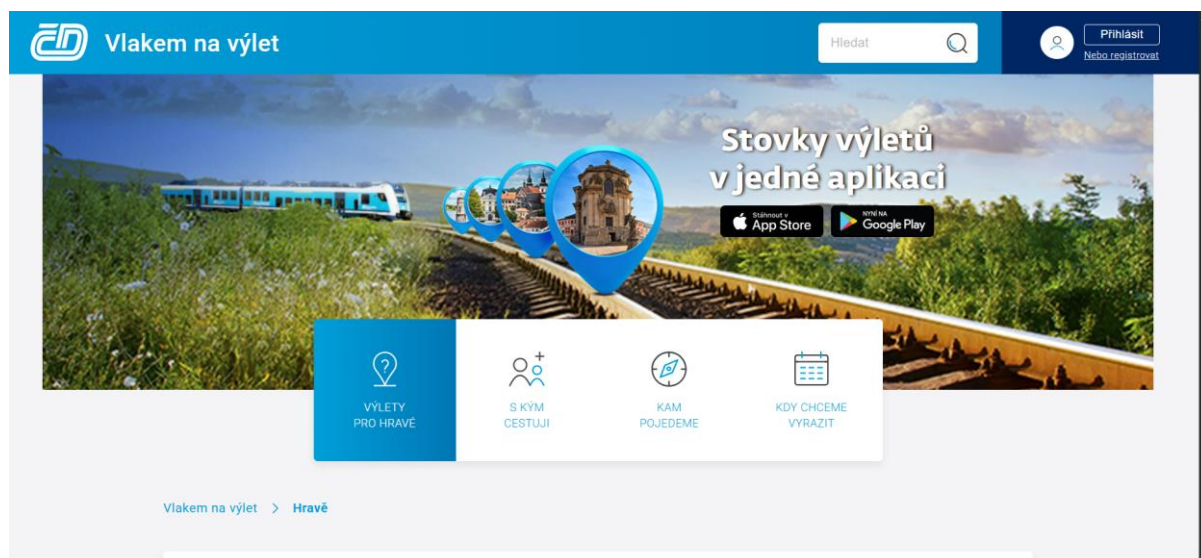
Это нарушает визуальную целостность интерфейса и может вызывать ощущение недоработанности дизайна.

В дальнейшем ходе визуального анализа сайта České dráhy были выявлены следующие интерфейсные недостатки, влияющие на восприятие пользователем и удобство взаимодействия:

- 1. Низкое качество изображения (страница "Vlakem na výlet")**

При загрузке страницы отображается растянутое и размытое изображение, что создаёт ощущение технической недоработки и снижает визуальную привлекательность ресурса.

[Ссылка на баг репорт](#)



2. Карусель баннеров не переключается автоматически (главная страница)

Баннеры на главной странице не листаются самостоятельно, несмотря на визуальные признаки наличия автопереключения. Пользователь должен листать вручную, что ухудшает восприятие промо-контента.

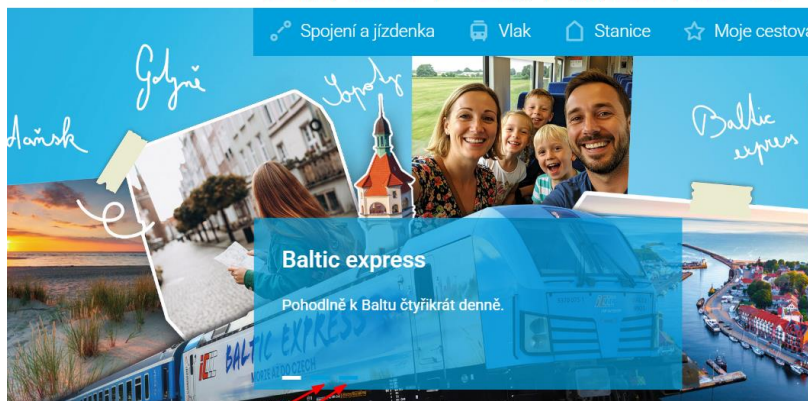
[Ссылка на баг репорт](#)

3. Плохая видимость стрелок управления баннерами

Кнопки прокрутки баннеров сливаются с фоном, что затрудняет их обнаружение. Это мешает пользователю понять, как перелистнуть содержимое карусели.

[Ссылка на баг репорт](#)



Скриншоты:





Baltic express
Pohodlné k Baltu čtyřikrát denně.


Handwritten notes: Gdyně, Yopohy, Baltic expres

Spojení a jízdenka

| | | |
|-------|-------|---|
| ODKUD | Odkud |  |
| KAM | Kam |  |

< Čt 15. 5. > < 22:35 >

 **Dospělý 26–64 let**
Bez slevového průkazu 



Čas na luxusní kolekci ČD

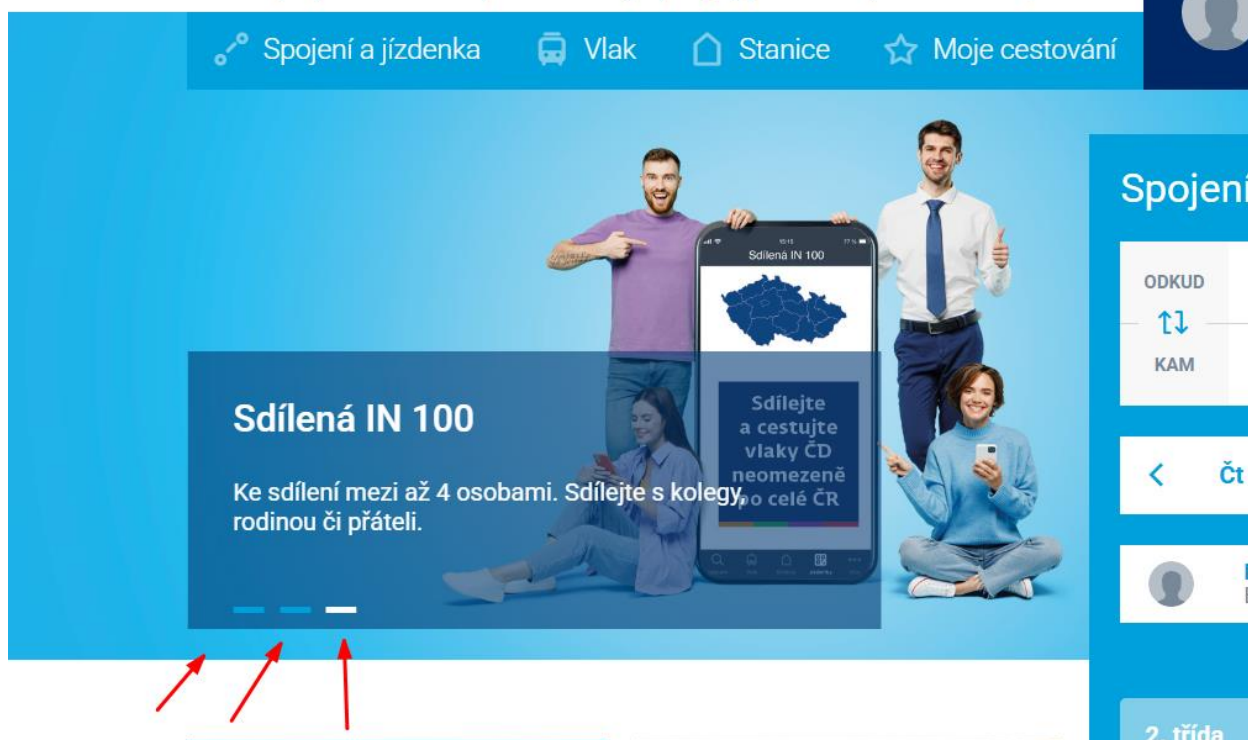
Limitovanou edici 68 ks hodinek PRIM najdete na našem fanshopu.

Jen 68 kusů

Spojení a jízdenka

| | |
|-------|--|
| ODKUD | |
| KAM | |

< Čt



Эти UI/UX баги влияют на восприятие сайта, делают использование менее комфортным. Относятся к эргономике интерфейса и качеству пользовательского опыта.

В рамках нефункционального тестирования были проведены базовые проверки удобства и доступности сайта. Основное внимание было уделено логике пользовательского пути, видимости элементов интерфейса, возможности навигации с клавиатуры, а также соответствию визуальных элементов принципам доступности. Проверена навигация с клавиатуры (Tab, Enter) и можно ли пройти формы без мыши.

Для анализа производительности использовался инструмент **Lighthouse**, встроенный в браузер Chrome. С его помощью были получены ключевые метрики загрузки страниц в мобильном и десктопном режимах. Для эмуляции мобильной версии применялся DevTools Chrome с выбором устройства Moto G4. Это позволило выявить возможные узкие места в производительности сайта как на стационарных, так и на мобильных устройствах

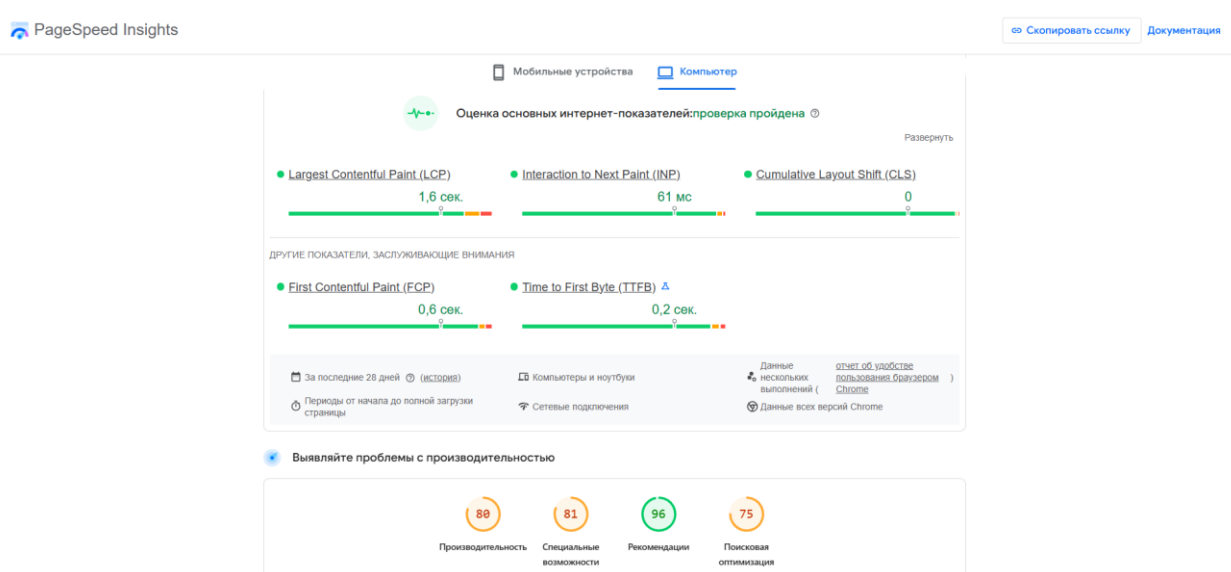
The image displays two screenshots of the TestRail web interface, showing test cases for a project named "České dráhy".

Top Screenshot (C3): The test case is titled "Проверка скорости загрузки главной страницы (десктоп)". It is categorized as "Type: Other", "Priority: Medium", "Assigned To: None", and "Estimate: None". The "References" and "Automation Type" are both set to "None". The "Preconditions" section states "Lighthouse в Chrome DevTools". The "Steps" section lists three steps: 1. Открыть сайт <https://www.cd.cz> в режиме десктоп, 2. Запустить Lighthouse с настройками «Performance», «Desktop», 3. Дождаться завершения анализа. The "Expected Result" section lists three criteria: "Загружается за < 3 сек", "Не более 100 сетевых запросов", and "Нет блокирующих JS (>300ms)".

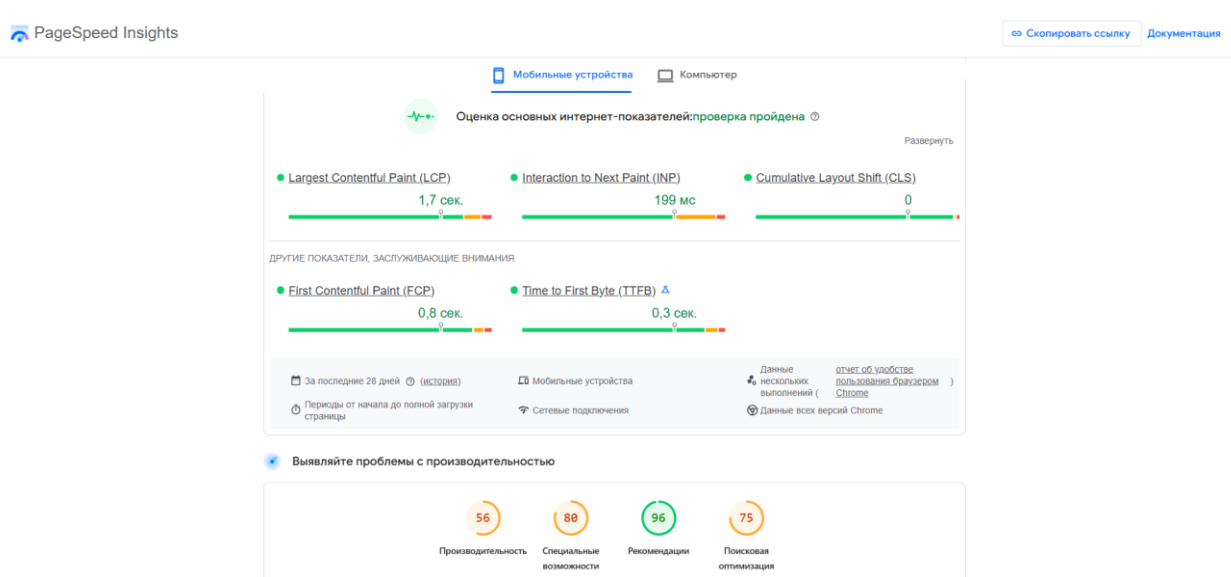
Bottom Screenshot (C4): The test case is titled "Проверка скорости загрузки главной страницы (мобильная версия)". It has the same metadata as C3. The "Preconditions" section states "Эмуляция мобильного устройства (например, Moto G4) включена в DevTools". The "Steps" section lists three steps: 1. Открыть сайт <https://www.cd.cz> в режиме эмуляции мобильного устройства, 2. Запустить Lighthouse с настройками «Performance», «Mobile», 3. Дождаться завершения анализа. The "Expected Result" section lists three criteria: "Метрика «Performance» > 60", "Time to Interactive < 4 сек", and "Cumulative Layout Shift (CLS) < 0.1".

[ссылка на тест кейс десктоп](#) и [тест кейс мобильная версия](#)

Ссылка на тест десктоп в https://pagespeed.web.dev/analysis/https-www-cd-cz/zez6mctcvj?form_factor=desktop



Ссылка на тест мобильная версия https://pagespeed.web.dev/analysis/https-www-cd-cz/zez6mctcvj?form_factor=mobile



Сайт <https://www.cd.cz> показывает среднюю производительность (74/100) на десктопных устройствах по данным Google PageSpeed Insights

Мобильная версия сайта <https://www.cd.cz> демонстрирует низкий показатель производительности (32/100)

Core Web Vitals показывает проблемы с Largest Contentful Paint (LCP), загрузкой ресурсов и работой погодного виджета, этот баг я регистрировала выше в своей работе. Используются устаревшие форматы изображений (JPEG/PNG). Нет использования `font-display: swap` у шрифтов. Некоторые

ресурсы не кэшируются эффективно. Неиспользуемые JavaScript и CSS присутствуют на странице.

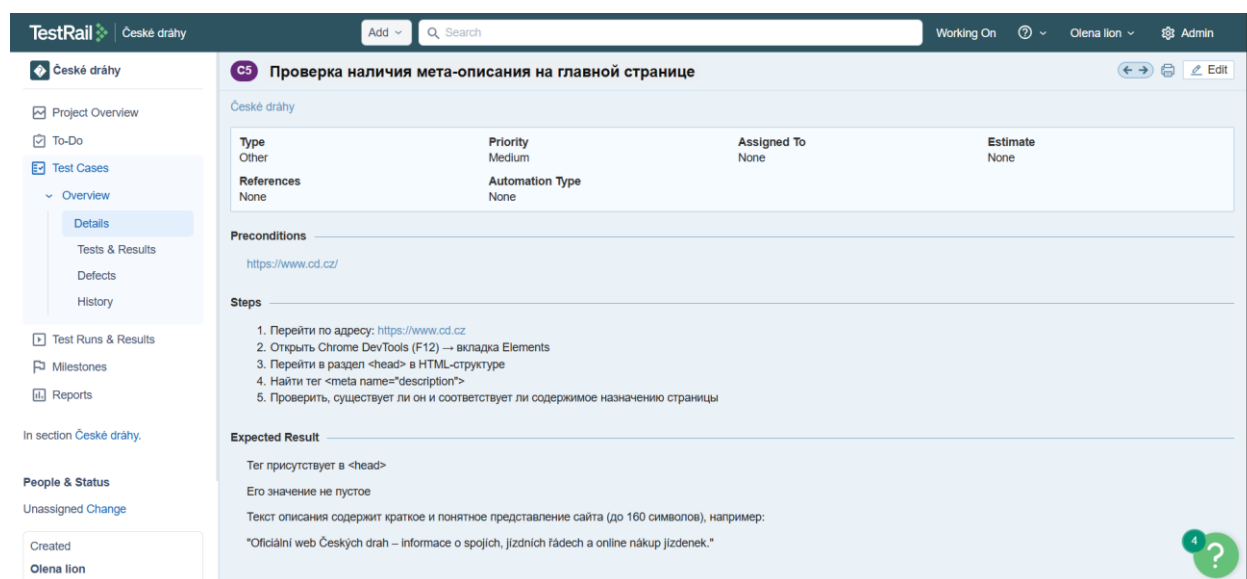
Сайт демонстрирует недостаточную оптимизацию для быстрой загрузки на всех устройствах.

[Ссылка на баг репорт десктоп](#)

[Ссылка на баг репорт мобильная версия](#)

С помощью инструмента **Lighthouse**, также выявился ошибка которая указывает на то, что нет метаописания которое должно содержать общие сведения о контенте страницы и могут быть показаны в результатах поиска. Это негативно влияет на SEO-оптимизацию сайта и может ухудшить отображение сайта в результатах поисковых систем.

[Тест кейс ссылка](#)



я разработала тест автоматизации который тоже подтверждает ошибку

[Ссылка на код автоматизации](#)

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
from config import URL
from selenium.common.exceptions import NoSuchElementException

options = webdriver.ChromeOptions()
```

```

options.add_argument(
    "--disable-blink-features=AutomationControlled")
options.add_argument("--window-size=1920,1080")

driver =
webdriver.Chrome(service=Service(ChromeDriverManager().install()),
options=options)

def check_og_tags():
    og_props = ["og:title", "og:description", "og:image"]
    for prop in og_props:
        try:
            meta = driver.find_element(By.CSS_SELECTOR,
f'meta[property="{prop}"]')
            content = meta.get_attribute("content")
            print(f" Найден {prop}: {content}")
        except NoSuchElementException:
            print(f" МЕТА-тег {prop} не найден на странице")

driver.get(URL)
check_og_tags()

```

МЕТА-тег og:title не найден на странице

МЕТА-тег og:description не найден на странице

МЕТА-тег og:image не найден на странице

Наличие автоматизированной проверки позволяет оперативно отслеживать ключевые SEO-проблемы. Это особенно важно при внесении изменений в шаблоны страниц, где такие баги могут появиться как **регрессии**.

[Ссылка на баг репорт](#)

[SCRUM-7] Отсутствует изображение предпросмотра ссылки (OG image) при отправке сайта в мессенджерах. Created: 15/May/25 Updated: 16/May/25

| | | | |
|-------------------|------------------|--|--|
| Status: | To Do | | |
| Project: | My Scrum Project | | |
| Components: | None | | |
| Affects versions: | None | | |
| Fix versions: | None | | |

| | | | |
|---------------------|------------------|-----------|------------|
| Type: | Bug | Priority: | Medium |
| Reporter: | Olena Malinowska | Assignee: | Unassigned |
| Resolution: | Unresolved | Votes: | 0 |
| Labels: | None | | |
| Remaining Estimate: | Not Specified | | |
| Time Spent: | Not Specified | | |
| Original estimate: | Not Specified | | |

| | |
|--------------|----------------|
| Attachments: | 19.PNG |
| Rank: | 0j0001r |
| Sprint: | SCRUM Sprint 1 |
| Severity: | Minor |

Description

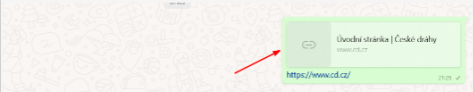
Preconditions: <https://www.cd.cz/> открыт в браузере

Steps to reproduce:

1. открываем страницу
2. копируем адрес страницы
3. посылаем ссылку через соц сети

Actual result

Изображение отсутствует, отображается иконка по умолчанию.



Expected result:

Ссылка содержит заголовок, описание и изображение (указанное в og:image), как это реализовано на большинстве современных сайтов.

Environment:

Windows 10, Google
Chrome

Стресс-тестирование

С целью демонстрации работы инструмента нагрузочного тестирования был проведён стресс-тест с использованием **Apache JMeter**. Так как доступ к тестовой среде сайта [cd.cz](https://www.cd.cz/) отсутствует, тест проводился в ограниченных условиях с числом виртуальных пользователей, равным 10. Это позволило визуализировать работу инструмента и принцип многопоточной нагрузки.

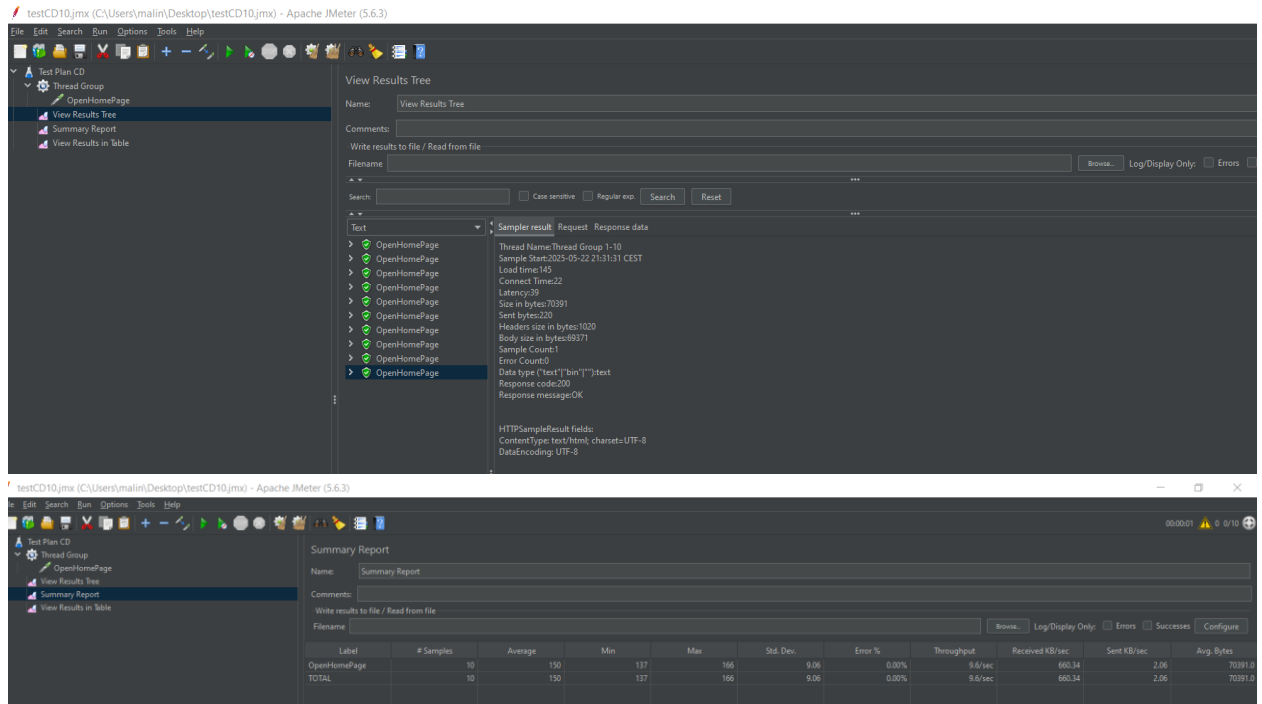
Настройки теста:

- Инструмент: Apache JMeter 5.6.3
- Цель: обращение 10 пользователей к домашней странице (GET-запрос)
- Название сценария: OpenHomePage
- Количество потоков (пользователей): 10
- Контроллеры: Thread Group, HTTP Request, View Results Tree, Summary Report

Результаты теста:

- Общее количество запросов: 10
- Среднее время отклика: **145 мс**
- Минимальное время: 137 мс
- Максимальное время: 166 мс

- Стандартное отклонение: 9.06 мс
- Ошибок: **0%**
- Пропускная способность (Throughput): 9.6 запросов/сек
- Передано данных: ~70391 байт
- Код ответа сервера: 200 OK
- Время подключения: 22 мс
- Ответ сервера корректный, кодировка UTF-8, MIME-тип text/html



Вывод:

Стресс-тест показал, что при обращении 10 пользователей одновременно сайт корректно обрабатывает все входящие запросы и возвращает правильный ответ. Ответы сервера приходят без ошибок, а средняя задержка составляет всего 150 мс. Это говорит о том, что при небольшой нагрузке сайт работает надёжно. Несмотря на то, что такой объём нагрузки не позволяет делать выводы о предельной производительности, он демонстрирует базовую работоспособность сайта под нагрузкой и подтверждает эффективность инструмента Apache JMeter.

Безопасность

SSL-сертификат и защита данных

SSL (Secure Sockets Layer) и его преемник TLS (Transport Layer Security) — это криптографические протоколы, предназначенные для защиты данных, передаваемых между пользователем и веб-сервером. Использование SSL/TLS обеспечивает шифрование передаваемой информации, а также подтверждает подлинность сервера.

Согласно анализу сайта www.cd.cz, выполненному с помощью онлайн-инструмента SSL Labs (Qualys), по состоянию на 24 мая 2025 года сайт получил рейтинг «В». Это означает, что соединение с сайтом зашифровано, однако присутствуют некоторые проблемы, снижающие общую оценку. В частности, сайт не поддерживает самый современный уровень защиты (например, отсутствие поддержки Perfect Forward Secrecy или старые версии TLS).

Для улучшения рейтинга безопасности до уровня А или А+ рекомендуется:

- Внедрить поддержку современных версий протокола TLS (например, TLS 1.3)
- Отключить устаревшие протоколы (TLS 1.0 и 1.1)
- Обеспечить поддержку механизмов Perfect Forward Secrecy (PFS)
- Настроить корректный порядок шифров (cipher suites)

Ссылка на актуальный анализ SSL-сертификата

<https://www.ssllabs.com/ssltest/analyze.html?d=www.cd.cz>

Анализ HTTP-заголовков безопасности (SecurityHeaders.com)

SecurityHeaders.com — это онлайн-инструмент, разработанный экспертом в области кибербезопасности Скоттом Хелмом (Scott Helme), который проверяет наличие и корректность HTTP-заголовков, влияющих на безопасность веб-приложения. Эти заголовки выполняют функции защиты от XSS, clickjacking, MIME-sniffing и других распространённых атак.

Согласно анализу сайта www.cd.cz, проведённому 24 мая 2025 года, сайт получил оценку «А». Это свидетельствует о высоком уровне реализации заголовков безопасности. Однако для достижения максимального рейтинга «А+» необходимо добавить два отсутствующих заголовка:

- Referrer-Policy

- Permissions-Policy

Заголовок Referrer-Policy позволяет контролировать, какие данные из URL передаются сторонним сайтам при переходах. Permissions-Policy управляет доступом к API браузера (например, камера, микрофон, геолокация), что уменьшает поверхность атаки.

Ссылка на актуальный анализ Security Headers :

<https://securityheaders.com/?q=https%3A%2F%2Fwww.cd.cz%2F&hide=on&followRedirects=on>

Контент

Мультимедийные элементы (изображения, иконки, графики, видео и анимации) играют ключевую роль в восприятии информации пользователями. Они способствуют лучшему визуальному представлению контента, эмоциональному вовлечению и навигации. В случае транспортных порталов, таких как www.cd.cz, фотографии поездов, схемы маршрутов, визуализация станций и интерфейсов приобретают особую значимость.

Обнаруженные особенности на сайте www.cd.cz

- Изображения в новостных блоках, на главной странице и в разделах услуг;
- Схемы и графики маршрутов;
- Информационные иконки (например, про багаж, билеты, доступность);
- Фотографии поездов и инфраструктуры Чешских железных дорог.

Однако в процессе тестирования были выявлены следующие проблемы:

- Некоторые изображения не сопровождаются альтернативным текстом (alt);

Адаптация для мобильной версии

Рекомендации по улучшению

- Добавить alt-атрибуты ко всем изображениям;
- Внедрить lazy-loading и оптимизацию изображений;

- Проверить корректность отображения мультимедиа на всех типах устройств;
- Применять векторные изображения для иконок (формат SVG);
- Интегрировать описание мультимедиа для пользователей с ограниченными возможностями.

Проблемы локализации мультимедийного и текстового контента

В процессе тестирования англоязычной версии сайта <https://www.cd.cz/en/eshop/o-eshopu/-28522/> была зафиксирована проблема с некорректной локализацией интерфейса. При переходе на англоязычную версию страницы (установлена английская локаль в URL), часть текста и интерфейса остаётся на чешском языке. Это создает путаницу для иностранных пользователей и снижает общее качество восприятия информации.

[Ссылка на баг репорт](#)



Přihlásit se
nebo registrovat

0 položek
0 Kč

Rules for purchasing in the ČD e-shop and the Můj vlak (My Train) application

Platnost od 10. 12. 2023

Version 6.1

Soubory ke stažení



(EN)Pravidla pro nákup v e-shop ČD a aplikaci Můj vlak

645 kB (pdf) [Stáhnout](#)
(only CZ version)

RULES FOR SHOPPING IN THE ČD E-SHOP AND MŮJ VLAK APP

A. General Part, Common for Domestic and International Transport

For the use of documents purchased through the ČD e-shop, the following apply:

- The Contractual Transport Conditions of Czech Railways for Public Railway Passenger Transport (hereinafter ČD SPPO).
- The Tariff of Czech Railways for Domestic Passenger and Baggage Transport (hereinafter ČD tariff).

9.8.2. For seat reservations on SC Pendolino in ČR – SR transport, documents can be returned by the passenger without deductions no later than 15 minutes before departure from the passenger's originating station. Later, a 100% deduction applies.

9.9. International Bicycle Ticket

9.9.1. An international bicycle ticket can be returned by the passenger without deductions until 23:59 on the day preceding the first day of validity, on the day of validity or later a 100% deduction applies.

[Úvod](#) > [E-shop](#) > [O e-shopu](#)

Zákaznická podpora

+420 221 111 122

Vrátit / vyměnit jízdenku

Kontaktní formulář

Časté dotazy

Objednat cestu skupiny

Objednat cestu cestujícího
na vozíku nebo asistenci

Nápověda

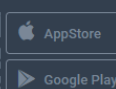
Správa cookies

Další webové stránky

Omezení provozu
Skupina České dráhy
Fanshop ČD
ČD Nostalgie
ČD Muzeum Lužná
S ČD bez překážek

📍 Mapa stránek
📡 RSS
✉ Kontakty

Jízdenky a informace v mobilu



Mobilní aplikace Můj vlak

ČD na sociálních sítích

✖ České dráhy
✖ Mimofádnosti
📺 YouTube
📘 Facebook
📷 Instagram

Čím se řídíme

Pravidla pro nákup v e-shopu
Čím se řídíme
Podmínky provozu
Prohlášení o přístupnosti webu
Práva subjektu osobních údajů
Ochrana osobních údajů

O firmě

Aktuality
Veřejné zakázky
Nemovitosti
Tiskové zprávy
Kariéra

Такая недоработка может нарушать пользовательский опыт, особенно для туристов, не владеющих чешским языком, и негативно сказывается на международном имидже цифровых сервисов.

Кроссбраузерное тестирование

Во время тестирования сайта www.cd.cz важно было проверить, как он работает в разных браузерах. Пользователи заходят не только с Chrome, но и с Firefox, Safari, Edge и другими. Поэтому сайт должен везде выглядеть и работать одинаково хорошо.

Для проверки кроссбраузерной совместимости использовались следующие версии браузеров:

- Google Chrome — 134.0.6998.166 (64-bit)
- Firefox — 136.0.2 (64-bit)
- Safari — 5.1.7 (7534.57.2)
- Microsoft Edge — 134.0.3124.85 (64-bit)

Это особенно удобно при ручном тестировании и при проверке кроссбраузерной совместимости.

Также в процессе изучила такой инструмент как BrowserStack — это облачная платформа для тестирования веб-сайтов и мобильных приложений в реальных условиях. Она позволяет разработчикам и тестировщикам проверять, как работает сайт или приложение:

- на разных браузерах (Chrome, Firefox, Safari, Edge и др.),
- на разных операционных системах (Windows, macOS, Android, iOS),
- на реальных устройствах, а не только в эмуляторах.

Можно протестировать сайт, чтобы убедиться, что он одинаково хорошо работает в Safari на iPhone и в Chrome на Windows 10 — без необходимости иметь все эти устройства у себя. К сожалению демоверсия имеет ограничения по тесту, но я изучила панель инструментов, настройки и возможности.

Заключение

В рамках данной дипломной работы было проведено всестороннее функциональное и нагрузочное тестирование веб-сайта чешской железнодорожной компании České dráhy (CD.cz). Исследование охватило все ключевые аспекты качества цифрового продукта: функциональность, производительность, юзабилити, безопасность, корректность контента и кроссбраузерную совместимость.

На основании ручного и автоматизированного тестирования были выявлены как корректно работающие участки сайта, так и конкретные дефекты, способные повлиять на пользовательский опыт, безопасность или восприятие ресурса. Были оформлены баг-репорты, приведены доказательства отсутствия валидации, зафиксированы ошибки отображения, слабые места SEO и уязвимости интерфейса. Также проведён анализ сетевой безопасности с использованием SSL Labs и SecurityHeaders.

Для демонстрации инструментов тестирования были использованы Selenium, Postman, JMeter, Lighthouse, DevTools, а также проведён анализ с помощью платформ BrowserStack, SSL Labs и SecurityHeaders.com. Это показало, как современный тестировщик может сочетать ручную и автоматизированную практику в рамках реального проекта.

Результаты тестирования подтвердили работоспособность основных функций сайта, но также выявили **ряд критичных и некритичных багов**, особенно в областях локализации, адаптивности и визуального оформления. Некоторые ошибки могут быть устранены относительно быстро, другие требуют системного подхода к улучшению архитектуры или клиентской части.

Работа над этим проектом позволила мне не только освоить ключевые инструменты тестирования, но и развить практические навыки анализа, формализации и документирования дефектов. Я на практике убедилась в важности тестирования как части жизненного цикла программного обеспечения и поняла, какую роль оно играет в обеспечении качества цифровых сервисов, особенно в такой чувствительной сфере, как транспорт.

[Ссылка на проект в Github](#)