

Time to generate:

ID	Scenario name
1	Access Homepage
2	Navigation Bar Visibility
3	Image Carousel Functionality
4	Search Panel Accessibility
5	User Testimonials Display
	Travel Tips and News
6	Section Access
	Newsletter Subscription
7	Section Access
8	Navigation to Flights Page
9	Navigation to Hotels Page
	Navigation to Car Rentals
10	Page
	Navigation to Special Offers
11	Page
	Navigation to Customer
12	Support Page
13	Responsive Design Check
	Navigation Bar Links
14	Functionality
15	Footer Accessibility
16	Load Time Performance
17	Cross-browser Compatibility
	Secure Connection
18	Verification
19	Secure Data Submission
20	Error Handling for Forms
	Load Performance Under
21	High Traffic
22	SEO Optimization Check
23	Accessibility Compliance
	Internationalization and
24	Localization
25	Link Integrity Check
	Image and Asset Loading
26	Efficiency

Mobile Navigation 27 Compatibility
Homepage Customization 28 Options
Security Certificates and 29 Encryption
User Session and Cookie 30 Management
Cache Behavior and 31 Efficiency
32 Ad Blocker Compatibility Privacy Policy and GDPR 33 Compliance
Homepage Feedback 34 Mechanism
35 Invalid URL Navigation
36 Navigation Bar Broken Links
Image Carousel Loading 37 Failure
38 Search Panel Validation SQL Injection Vulnerability in 39 Search Panel
Cross-Site Scripting (XSS) in 40 Testimonials
Newsletter Subscription 41 Email Validation
Performance Under High 42 Traffic
43 Secure Data Transmission Privacy Policy and Terms of 44 Service Access
Disabled JavaScript 45 Handling
Form Resubmission 46 Handling
47 Ad Blocker Interference Cookie Handling and 48 Consent
Responsiveness to Screen 49 Reader Software
User Testimonials Section 50 Overflow Handling

Newsletter Subscription 51 Confirmation Email
Carousel Auto-Rotation 52 Functionality
Manual Navigation 53 Functionality
54 Image Load Performance
Carousel Image Click- 55 through
56 Carousel Responsiveness
57 Image Description Visibility
Negative Scenario: Broken 58 Image Links
Accessibility for Screen 59 Readers
Touchscreen Navigation 60 Compatibility
Keyboard Navigation 61 Compatibility
Carousel Pause on Hover or 62 Focus
Performance under High 63 Traffic
64 Cross-browser Compatibility
Handling Non-Clickable 65 Images
Security Measures for Image 66 Source URLs
Carousel Load Time 67 Performance
Image Quality and 68 Optimization
Valid Booking Criteria 69 Submission
70 Dynamic Form Functionality
71 Form Submission Redirect
Input Validation and Error 72 Messaging
73 Date Range Validation
74 Performance Under Load
Security of Data 75 Transmission

76	Handling of Unavailable Services
77	Responsiveness on Various Devices
78	Negative: Invalid Destination Input
79	Negative: Past Date Selection
80	Load Time and Responsiveness
81	Session Timeout Handling
82	Cross-Site Scripting (XSS) Protection
83	SQL Injection Protection
84	Visibility of Newsletter Subscription Section
85	Email Address Input Validation
86	Interest Selection Functionality
87	Successful Subscription Process
88	Confirmation Email Receipt
89	Unsubscription Link Functionality
90	Responsive Design of Subscription Form
91	Negative Scenario: Duplicate Email Subscription
92	Security of Email Transmission
93	Invalid Interest Selection Handling
94	Performance of Subscription Submission
95	Cross-Site Scripting (XSS) Protection in Form
96	SQL Injection Protection in Subscription Form
97	Accessibility for Visually Impaired Users
98	Error Handling for Server Issues
99	Accessibility of Travel Tips and News Section
100	Display of Latest Articles
101	Article Selection Functionality

102 Article Loading Performance
Responsive Design for
103 Article Viewing
Negative Scenario: Broken
104 Links to Articles
105 Content Search Functionality
Comments Section
Accessibility and
106 Functionality
Social Media Sharing
107 Options
Article Bookmarking or Save
108 for Later Feature
Navigation Back to
109 Homepage
Accessibility for Visually
110 Impaired Users
Performance Under High
111 Traffic
112 Security of External Links
Error Handling for
113 Unavailable Content
Visibility and Accessibility of
114 Testimonials
115 Display of User Testimonials
Navigation Through
116 Testimonials
117 Load Time and Performance
Testimonials
118 Responsiveness
Negative Scenario: Missing
119 Testimonials
User Interaction with
120 Testimonials
Security of User-Submitted
121 Content
Accessibility for Visually
122 Impaired Users
Dynamic Update of
123 Testimonials
Feedback Mechanism for
124 Testimonials
Performance Under High
125 Traffic
Testimonials Pagination or
126 Load More Feature

10-15 minutes

Description

Verify that the GlobeTrotter homepage is accessible.

Check visibility and accessibility of the navigation bar.

Test the functionality of the image carousel on the homepage.

Ensure the search panel is accessible and functional.

Verify that user testimonials are correctly displayed on the homepage.

Check access to the travel tips and news section.

Ensure the newsletter subscription section is accessible and functional.

Test navigation from the homepage to the Flights page via the navigation bar.

Test navigation from the homepage to the Hotels page via the navigation bar.

Test navigation from the homepage to the Car Rentals page via the navigation bar.

Test navigation from the homepage to the Special Offers page via the navigation bar.

Test navigation from the homepage to the Customer Support page via the navigation bar.

Verify the homepage's responsiveness on different devices.

Ensure all links in the navigation bar work correctly and lead to the correct pages.

Test the accessibility of the footer section from the homepage.

Evaluate the load time and performance of the homepage.

Ensure the homepage displays and functions correctly across different web browsers.

Verify that the homepage is served over HTTPS and maintains a secure connection.

Ensure all forms on the homepage submit data securely.

Test the error handling capabilities of forms on the homepage.

Simulate high traffic conditions to test homepage load performance.

Verify that the homepage is optimized for search engines.

Test compliance with web accessibility standards (e.g., WCAG).

Test the homepage's ability to display content in different languages based on user preference or location.

Verify that all internal and external links on the homepage are working correctly without leading to 404s.

Test the loading efficiency of images and other assets on the homepage.

Ensure the navigation bar and other interactive elements work on mobile devices.

Test any user customization features available on the homepage.

Verify the validity of security certificates and the encryption of data in transit.

Test the management of user sessions and cookies for returning visitors.

Test the homepage's use of browser cache for repeat visits.

Test homepage functionality with common ad blockers enabled.

Ensure the homepage complies with privacy laws, including GDPR for European visitors.

Test the functionality and visibility of feedback mechanisms on the homepage.

Verify the system's response to navigation to an invalid or nonexistent URL.

Check for any broken links in the navigation bar.

Test the homepage's response when images in the carousel fail to load.

Verify input validation for the search panel.

Test for SQL injection vulnerabilities in the search panel.

Check for XSS vulnerabilities in user testimonials.

Ensure the newsletter subscription form validates email addresses correctly.

Evaluate homepage performance under simulated high traffic.

Verify that all data transmitted from the homepage is encrypted.

Test the accessibility and visibility of privacy policy and terms of service links.

Verify the homepage's functionality with JavaScript disabled.

Test the system's handling of form resubmissions on the homepage.

Assess homepage functionality with an ad blocker enabled.

Verify the homepage's cookie handling and user consent mechanism.

Test homepage accessibility with screen reader software.

Test the handling of an excessive number of user testimonials.

Verify the sending and content of the newsletter subscription confirmation email.

Verify the auto-rotation feature of the image carousel.

Test the manual navigation controls of the carousel.

Assess loading times for images in the carousel.

Verify that clicking an image navigates to the correct destination or offer page.

Evaluate the carousel's responsiveness on various devices and screen sizes.

Check the visibility of image descriptions or promotions.

Test the carousel's handling of broken image links.

Verify carousel accessibility with screen reader software.

Test carousel navigation on touchscreen devices.

Assess the carousel's compatibility with keyboard navigation.

Test if the carousel pauses rotation when hovered over or focused.

Evaluate the carousel's performance under simulated high traffic.

Ensure the carousel displays and functions correctly across different browsers.

Negative scenario where images are not clickable due to issues.

Test for security vulnerabilities through the image source URLs.

Test the initial load time of the carousel on the homepage.

Assess the quality and optimization of carousel images for web use.

Ensure users can successfully submit valid booking criteria.

Test dynamic form adjustments based on service selection.

Verify redirection to search results upon form submission.

Ensure input validation for all fields with clear error messaging.

Test validation of date ranges within the booking form.

Assess form submission performance under simulated high traffic.

Verify that booking criteria are transmitted securely.

Test system response when selected services are unavailable at the destination.

Ensure the booking process is fully responsive on desktops, tablets, and smartphones.

Verify system behavior with invalid or non-existent destination input.

Test system's handling of past date selections for bookings.

Evaluate load times and responsiveness of the booking form and search results.

Assess how the system handles session timeouts during the booking process.

Test for XSS vulnerabilities in the booking form.

Ensure the booking process is secure against SQL injection attacks.

Verify that the newsletter subscription section is easily accessible on the homepage.

Ensure the form validates email address inputs correctly.

Test the ability to select multiple interests.

Confirm the subscription process adds the user to the mailing list.

Verify that a confirmation email is sent upon subscription.

Test the functionality of the unsubscription link within the confirmation email.

Assess the responsiveness of the subscription form on various devices.

Verify the system's handling of duplicate email subscriptions.

Ensure the email address is transmitted securely to the server.

Test behavior when no interests are selected (if applicable).

Evaluate the performance and speed of the subscription process.

Test for XSS vulnerabilities in the subscription form.

Ensure the subscription form is secure against SQL injection attacks.

Verify the subscription section is accessible with screen readers.

Test system response to backend server issues during subscription.

Verify that the travel tips and news section is easily accessible from the homepage.

Ensure the latest articles, tips, and news items are displayed.

Test the functionality of selecting an article to read.

Evaluate the load time for articles.

Assess the readability and layout of articles on various devices.

Verify the handling of broken links or missing articles.

Test the ability to search within the travel tips and news section.

Assess the functionality and accessibility of the comments section (if applicable).

Verify the functionality of social media sharing options for articles.

Test the ability to bookmark or save articles for later reading.

Ensure users can easily navigate back to the homepage from an article.

Verify article section accessibility with screen readers.

Evaluate the section's performance under simulated high user traffic.

Ensure any external links within articles are secure and lead to trusted sites.

Test system response to attempts to access unavailable content.

Ensure the testimonials section is easily accessible and visible on the homepage.

Verify that testimonials from users are displayed correctly.

Test the functionality of navigating through multiple testimonials.

Evaluate the load time and performance of the testimonials section.

Assess the testimonials section's responsiveness on various devices.

Verify the handling of missing or unavailable testimonials.

Test any user interaction features within the testimonials section.

Ensure user-submitted testimonials are sanitized to prevent XSS attacks.

Verify the testimonials section is accessible to users with visual impairments.

Test the dynamic updating of the testimonials section with new entries.

Assess the presence and functionality of a feedback mechanism for testimonials.

Evaluate the section's performance under simulated high user traffic.

Test the functionality of pagination or a "load more" feature for displaying more testimonials.

Details

Ensure the homepage loads successfully with all elements visible: image carousel, navigation bar, search panel, user testimonials, travel tips and news, subscription CTA.

The navigation bar should be easily visible and accessible at the top of the homepage.

The image carousel should auto-rotate images. Manual navigation (next/previous) should also work.

The search panel should accept input and provide relevant form fields for different services (flights, hotels, etc.).

Testimonials should be visible, and navigation through them (if available) should function correctly.

This section should display the latest articles, tips, and news items correctly.

The form should prompt for an email address and interest selection, accept inputs, and submit correctly.

Selecting "Flights" from the navigation bar should redirect the user to the Flights page.

Selecting "Hotels" from the navigation bar should redirect the user to the Hotels page.

Selecting "Car Rentals" from the navigation bar should redirect the user to the Car Rentals page.

Selecting "Special Offers" from the navigation bar should redirect the user to the Special Offers page.

Selecting "Customer Support" from the navigation bar should redirect the user to the Customer Support page.

The homepage should render correctly and remain fully functional on desktops, tablets, and smartphones.

Every option in the navigation bar should be clickable and redirect to the corresponding page or section.

The footer should be accessible, and all links in it should function correctly, including social media links, privacy policy, terms of service, etc.

The homepage should load within a reasonable time frame, without any noticeable delays.

Test the homepage on major web browsers like Chrome, Firefox, Safari, and Edge to ensure compatibility.

The URL should begin with "https://" and there should be no security warnings from the browser.

Data entered in search panels and newsletter subscription forms should be submitted via HTTPS to protect user information.

Input validation for the search panel and newsletter subscription form: test with invalid data (e.g., incorrect email format) to ensure the system handles errors gracefully.

Use load testing tools to simulate multiple accesses to the homepage simultaneously to ensure it remains responsive under heavy load.

Ensure proper use of meta tags, alt attributes for images, and that the content is optimized for relevant keywords to improve search engine ranking.

Ensure the homepage is navigable and usable with assistive technologies like screen readers, and complies with accessibility guidelines (e.g., contrast ratios, alt text for images).

If applicable, ensure the homepage automatically detects user location to tailor content in the local language, and that manual language selection works correctly.

Test all links in the navigation bar, footer, travel tips and news section, and promotional banners to ensure they are not broken and lead to the correct pages.

Ensure images, CSS, and JavaScript files are optimized for quick loading, potentially using techniques like compression, minification, and lazy loading.

Test on various mobile devices to ensure touch-friendly navigation, especially the responsiveness of the navigation bar and the visibility of interactive elements.

If the homepage allows customization (e.g., theme selection, widget arrangement), ensure these features work correctly and persist across sessions.

Ensure the site's SSL/TLS certificate is valid and up-to-date, and that data transmission to and from the homepage is encrypted.

Ensure that user sessions are correctly initiated and terminated, cookies are securely set for session management, and that user preferences are remembered through cookies.

Ensure static assets are cached by the browser to reduce load times on subsequent visits, without affecting the freshness of dynamic content like news or special offers.

Ensure that essential homepage features and content display correctly even when users have ad blockers enabled, noting any compatibility issues.

Verify that the site provides clear privacy policy information, manages cookies appropriately, and obtains consent for data collection where required.

If there's a feature for users to give feedback on the homepage or report issues, test to ensure it's functional and user submissions are correctly handled.

Attempting to navigate to a URL that doesn't exist within the GlobeTrotter site (e.g., a typo in the URL) should redirect the user to a custom 404 page or the homepage, with an appropriate error message displayed.

Each link in the navigation bar should be tested to ensure it's not broken. If a link is broken (leads to a 404 page or an error), it should be logged as an issue.

Simulate a scenario where images in the carousel cannot load (due to network issues or missing files) to ensure the system provides a placeholder image or an error message without breaking the layout.

Entering invalid data (e.g., past dates for flights, special characters) should trigger appropriate error messages and prevent the form from being submitted.

Attempt to inject SQL code into search fields to check if the system is vulnerable. The system should sanitize inputs to prevent SQL injection attacks.

Attempt to inject JavaScript code into testimonials (if users can submit testimonials). The system should sanitize inputs to prevent XSS attacks.

Entering an invalid email address (e.g., without an "@" symbol) should trigger an error message and prevent form submission.

Use tools to simulate high traffic to the homepage and measure load times, responsiveness, and any potential downtime. The site should handle high traffic efficiently, without significant performance degradation.

Intercept the data transmitted during actions like form submissions to ensure it's encrypted using HTTPS, protecting against man-in-the-middle attacks.

These links should be easily accessible from the homepage, and clicking on them should take the user to the respective documents.

Disable JavaScript in the browser and assess how the homepage functions. Essential information should still be accessible, though dynamic features may be limited.

Submitting a form (e.g., search panel, newsletter subscription) and then refreshing the page should not resubmit the form or create duplicate entries/subscriptions. An appropriate mechanism should prevent this.

Enable a popular ad blocker and access the homepage to ensure essential features (like the navigation bar, search panel, and image carousel) still work correctly.

The site should request consent for cookies per applicable regulations (e.g., GDPR). Rejecting cookies should not significantly degrade the user experience.

Use screen reader software to ensure all content and navigation elements on the homepage are accessible and properly labeled for visually impaired users.

Ensure the testimonials section properly handles and displays a large number of testimonials without layout issues. Pagination or scrolling mechanisms should work correctly.

Upon subscribing to the newsletter, the user should receive a confirmation email. This email should correctly reflect the subscription and contain no broken links or

Ensure that the image carousel automatically rotates through images at a set interval without user intervention.

Users should be able to manually navigate through the carousel using next/previous buttons or indicators for each image.

Images should load within a reasonable time to ensure a smooth user experience, even under slower internet connections.

Clicking on an image in the carousel should take the user to a detailed page about the destination or the offer, with the URL and page content matching the clicked image's context.

The image carousel should adjust appropriately for different screen sizes and devices, maintaining usability and visibility of images and navigation controls.

Descriptions or promotions associated with each image should be clearly visible and legible, enhancing the user's understanding of the destination or offer.

In case an image fails to load (e.g., due to a broken link), the carousel should display a placeholder image or a meaningful error message, without breaking the carousel functionality.

Ensure that the carousel and its navigation controls are accessible via screen readers, with each image having an alt text that describes the destination or offer.

On touchscreen devices, users should be able to swipe through the carousel images smoothly.

Users relying on keyboard navigation should be able to cycle through the carousel images using keyboard controls (e.g., arrow keys).

To enhance usability, the auto-rotation should pause when the user hovers the mouse over an image or when an image gains keyboard focus, resuming when the user moves away or focus changes.

Simulate high traffic to the website to ensure the carousel remains responsive and functional, with images loading correctly and navigation controls responsive.

Test the image carousel on major web browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure compatibility and consistent functionality.

Ensure that in scenarios where images cannot be clicked (due to loading errors or script failures), the system provides feedback or fails gracefully, potentially with a message or alternative navigation option.

Ensure that the image URLs cannot be manipulated to inject malicious code or redirect users to harmful sites. Inputs should be sanitized and validated to prevent security risks.

The carousel should load efficiently with the rest of the homepage, without causing significant delays in page load time.

Images should be of high quality without being unnecessarily large in file size, which could affect loading times. This balance is crucial for maintaining both aesthetic appeal and performance.

Users enter valid dates, destination, and other criteria into the search panel for flights, hotels, or car rentals. System should display appropriate results.

The form should dynamically adjust to display relevant fields for flights, hotels, or car rentals upon selection.

Upon submitting the search form, the system redirects the user to the search results page relevant to the selected service with options based on entered criteria.

The system validates input for each field (e.g., date ranges, destination validity) and displays an error message for invalid inputs.

Ensure that the departure date cannot be set after the return date and that past dates cannot be selected, with appropriate error messages displayed for invalid date ranges.

Simulate high user traffic and measure the response time of the system to process booking criteria submissions and display search results.

Data entered in the search panel should be encrypted, especially in transmissions to prevent data leakage or interception.

If a user selects a destination where certain services (e.g., car rentals) are not available, the system should inform the user with appropriate messaging or alternatives.

The dynamic form and all its elements should be easily navigable and usable across devices of various screen sizes.

Entering a non-existent destination should prompt an error message or suggestion, preventing form submission.

Attempting to select past dates for any service should be prevented or result in an error message, disallowing form submission.

The booking form should load quickly, and search results should appear within a reasonable timeframe without significant delays.

If the user's session times out during the booking process, the system should prompt the user to refresh or restart the booking process without losing previously entered data.

Attempt to inject JavaScript via form fields to check for XSS protection. The system should sanitize inputs to prevent such attacks.

Attempt to inject SQL commands into form fields to test for SQL injection vulnerabilities. The system should reject such inputs.

Users should be able to locate and scroll to the newsletter subscription section without issues.

The form should accept valid email addresses and reject invalid ones, with appropriate error messaging for invalid formats.

Users should be able to select one or multiple topics of interest from the options provided.

After submitting the form with a valid email and selected interests, the user should be added to the newsletter mailing list.

Users should receive a confirmation email shortly after subscribing, confirming their addition to the mailing list.

The confirmation email should include an unsubscription link that works correctly, allowing users to easily opt-out of future newsletters.

The subscription form should be fully functional and visually appealing on desktops, tablets, and smartphones.

Attempting to subscribe with an email already on the mailing list should result in an appropriate message indicating the email is already subscribed or offer a re-subscription.

The transmission of email data should be encrypted to protect against interception or unauthorized access.

If the form allows submission without selecting interests, it should either prompt the user to select at least one or handle the submission appropriately.

The submission of the subscription form should be processed quickly, without significant delays, enhancing user experience.

Attempt to inject JavaScript via the email input field to check for XSS protection. The system should sanitize inputs to prevent such attacks.

Attempt to inject SQL commands into the email input field to test for SQL injection vulnerabilities. The system should reject such inputs.

The subscription form, including input fields and selection options, should be accessible to visually impaired users using screen readers.

Simulate a server failure or disconnection to ensure the system gracefully handles errors, possibly by displaying a friendly error message or retry option.

Users should be able to navigate to the travel tips and news section without difficulties, using both menu navigation and direct links if available.

The section should show the most recent content prominently, allowing users to see the latest articles first.

Clicking on an article should lead the user to the full content of the article. Navigation to the article should work correctly.

Articles should load within a reasonable timeframe, ensuring a smooth user experience even on slower internet connections.

Articles should be easily readable and well-formatted on desktops, tablets, and smartphones, with images and text properly scaled.

If a user clicks on a link to an article that is missing or has been removed, the system should display a helpful error message or redirect to a related content page.

Users should be able to use a search function to find articles on specific topics. The search should return relevant results.

If articles have a comments section, it should be easily accessible and allow users to read and post comments. Validation and moderation mechanisms should be tested if present.

Social media sharing buttons (if available) should work correctly, allowing users to share articles on various platforms.

Users should have the option to save articles for later reading. This feature should be tested for functionality and usability.

After reading an article, there should be a clear and functional option for users to return to the homepage or the travel tips and news section.

The travel tips and news section, including article content, should be accessible to visually impaired users using screen readers.

Use tools to simulate high traffic to the travel tips and news section and measure load times, responsiveness, and any potential downtime.

External links, if present in articles, should be verified for security to prevent directing users to malicious sites.

Simulate access to content that has been removed or is temporarily unavailable to ensure the system provides a clear message or redirects to available content.

Users should be able to locate and scroll to the testimonials section without difficulty, and the section should be prominently displayed.

The section should show a selection of testimonials from users, including their experiences with GlobeTrotter services. The display should include text and possibly photos.

If the system allows navigating through testimonials (e.g., via arrows or a carousel), this functionality should work smoothly, showing different testimonials.

The testimonials should load within a reasonable timeframe, ensuring a smooth user experience, even on slower internet connections.

The testimonials section should be easily readable and well-formatted on desktops, tablets, and smartphones, with text and images properly scaled.

In the event that testimonials are missing or unavailable, the system should handle this gracefully, possibly with a message indicating the situation.

If users can interact with testimonials (e.g., "like," "share," or leave comments), these features should be tested for functionality.

If users can submit testimonials directly, the input should be sanitized to prevent cross-site scripting (XSS) and other security vulnerabilities.

The section, including navigation controls (if present), should be accessible via screen readers, and alternative text should be provided for images.

If the testimonials section is dynamically updated, ensure that new testimonials are added smoothly without disrupting the user experience.

If there is a mechanism for users to provide feedback on testimonials (e.g., reporting inappropriate content), it should be functional and easy to use.

Use tools to simulate high traffic to the testimonials section and measure load times, responsiveness, and any potential downtime.

If the section uses pagination or a "load more" feature to display additional testimonials, this functionality should work correctly and load additional content as expected.

Bug Num
#1
#2
#3
#4
#5
#6

Bug Name

Poškozené obrázky v karuselu

Sekce s uživatelskými recenzemi

Nefunkční odkazy v navigačním baru

Formulář pro přihlášení k odběru novinek

Po kliknutí na článek tipů na cestování stránka přesměruje jinam než má

Zastaralá knihovna způsobí pád aplikace

Homepage

Bug Description

Karusel na domovské stránce má integrovány obrázky s vysokým rozlišením prezentující populární destinace. Chyba se projevuje tak, že místo obrázků se uživateli zobrazují ikony poškozeného obrázku.

Sekce s uživatelskými recenzemi obsahuje textové bloky a obrázky, které se nekorektně překrývají kvůli chybám v CSS stylování nebo nedostatkům v responzivním designu.

Navigační bar obsahuje odkazy vedoucí na různé části webu. U některých odkazů dochází k chybě, kdy jsou uživatelé přesměrováni na neexistující stránky, což vyústí v chybu 404.

Uživatel může zadat svůj e-mail do formuláře pro odběr novinek. Formulář akceptuje e-mail, ale systém selže při odesílání potvrzovacího e-mailu uživateli.

Odkazy na články v sekci nejnovějších cestovních zpráv mají za následek zobrazení článku jenž není ten na který bylo kliknuto

Knihovna, jenž má za úkol spravovat rotaci karuselu způsobí pád aplikace po kliknutí na jeden z rotujících obrázků

Bug Type	Found	Test Scenario ID
Vizuální chyba	Y	68
Vizuální chyba	Y	1.13
Funkční chyba	Y	8-13
Funkční chyba	Y	88
Funkční chyba	Y	101
Konfigurační chyba	Y	55

Time to generate:

ID	Scenario name
1	Navigation to Flights Section
2	Display of Dynamic Search Form
3	Input Validation for Search Form
4	Search Button Visibility and Functionality
5	Flight Search Execution and Results Display
6	Error Handling for No Results
7	Search Functionality Under High Load
8	Security of Data Transmission
9	Responsive Design of Search Form
10	Negative Scenario: Invalid Date Range
11	Negative Scenario: Unavailable Destination
12	Session Timeout and Search Persistence
13	Autocomplete Functionality for Locations
14	Modification of Search Parameters Before Search
15	Accessibility for Visually Impaired Users
16	Access Flight Search from Homepage
17	Access Flight Search Directly
18	Valid Search Submission
19	Display of Search Button
20	Handling of Invalid Locations
21	Date Validation
22	Passenger Number Validation
23	Class of Service Selection
24	Performance Under High Load

Security - Injection
25 Vulnerability in Form
26 Search Results Accuracy
Negative Scenario: Network
27 Connectivity Issues
Negative Scenario: Service
28 Unavailability
Flight Search Form Reset
29 Functionality
30 Cross-Browser Compatibility
Mobile Responsiveness and
31 Usability
32 Automated Bot Detection
33 Basic Autofill Functionality
34 Accuracy of Suggestions
Performance of Suggestions
35 Display
36 Selection of Suggestion
Suggestions List Collapse on
37 Selection
Handling of Uncommon
38 Locations
Negative Scenario: No
39 Matches Found
40 Handling Special Characters
Responsiveness on Mobile
41 Devices
Keyboard Navigation
42 Support
Suggestions Visibility with
43 Long Input
Performance Under High
44 Load
Security of Data
45 Transmission in Autofill
Autofill Functionality with
46 Slow Network
Multilingual and Special
47 Location Names Handling
48 Flexible Dates Display
49 Date Selection Functionality
Update Search Form with
50 Selected Date

Performance of Flexible 51 Dates Feature
52 Usability on Various Devices
Negative Scenario: No Price 53 Differences Available
Accuracy of Price 54 Differences
Flexible Dates Option 55 Visibility
Handling of Date Selection 56 Outside Available Range
Interaction with Other Form 57 Elements
Keyboard and Mouse 58 Navigation
59 Refreshing Price Differences
Security of Data 60 Transmission
Flexible Dates Feature 61 Under High Traffic
Error Handling for Calendar 62 Loading Failures
Filter Display on Search 63 Results Page
Functionality of Stopover 64 Filter
65 Functionality of Airline Filter
Functionality of Flight 66 Duration Filter
67 Functionality of Price Filter
68 Multiple Filters Application
69 Filter Reset Functionality
Performance of Filters 70 Application
Usability of Filters on Mobile 71 Devices
Negative Scenario: No 72 Results After Filtering
73 Persistent Filter Selections
74 Filter Selection Feedback
75 Accessibility of Filter Options
Edge Case: Filtering with 76 Rare Criteria

Security of Filter Data 77 Transmission
78 Display of Flight Details
79 Accessibility of Flight Details Usability of Flight Details
80 Page on Mobile
Performance of Loading 81 Flight Details
82 Accuracy of Flight Details
83 Layover Details Clarity Availability of Amenities
84 Information
Negative Scenario: Missing 85 Flight Details
Navigation Back to Search 86 Results
87 Flight Details Link Sharing
88 Printing Flight Details
Comparison Feature 89 Integration
Security of Flight Details 90 Transmission
Multilingual Support for Flight 91 Details
Accessibility Features for 92 Viewing Details
Ease of Accessing 93 Modification Options
Accurate Update of Search 94 Parameters
Validation of New Search 95 Parameters
Responsiveness of Updated 96 Search Results
Retention of Modified 97 Parameters on Refresh
User Feedback on Search 98 Update
Negative Scenario: Invalid 99 Search Inputs
100 Usability on Various Devices

101 Performance Under Load
Clear Option to Reset
102 Search Criteria
103 Accessibility for All Users
Confirmation Before Search
104 Update
Basic Comparison
105 Functionality
Accuracy of Flight Details in
106 Comparison
Usability of Comparison
107 Feature
Performance of Loading
108 Comparisons
109 Selection Flexibility
Maximum Number of Flights
110 for Comparison
Negative Scenario: No
111 Differences Found
Comparing Across Different
112 Dates
113 Highlighting Key Differences
114 Mobile Responsiveness
Accessibility for Users with
115 Disabilities
Performance Under High
116 Load
Clear Option to Proceed
117 After Comparison
118 Basic Save Functionality
Confirmation Prompt
119 Visibility
120 Accuracy of Saved Searches
Usability of Save Feature on
121 Various Devices
Performance of Saving
122 Action
Negative Scenario: Save
123 Failure
124 User Feedback After Saving
Saved Search Access and
125 Management
Security of Saved Search
126 Data

127 Multi-Session Persistence
128 Limitations and Notifications
Accessibility for Users with
129 Disabilities
Edge Case: Saving with
130 Incomplete Criteria
131 Basic Sharing Functionality
Availability of Sharing
132 Options
133 Email Sharing Functionality
Social Media Sharing
134 Usability
Accuracy of Shared Flight
135 Details
Performance of Sharing
136 Feature
Negative Scenario: Sharing
137 to Invalid Recipient
Confirmation of Successful
138 Sharing
Security of Shared
139 Information
Direct Link Sharing
140 Functionality
141 Usability on Various Devices
Limitations and Restrictions
142 on Sharing
Accessibility for Users with
143 Disabilities
144 Accessing Booking Form
145 Completing Booking Form
Total Cost Calculation
146 Accuracy
Review and Confirmation of
147 Booking
Payment Information
148 Submission
Booking Confirmation and
149 Email Receipt
150 Usability on Various Devices
Negative Scenario: Payment
151 Failure

Security of Personal and
152 Payment Information

Modification and
153 Cancellation Options

Application of Promotional
154 Codes and Vouchers

Accessibility for Users with
155 Disabilities

Performance Under High
156 Load

Error Handling and User
157 Feedback

15-25 minutes

Description

Verify that users can navigate to the "Flights" section from the homepage.

Ensure the dynamic search form is displayed upon accessing the Flights section.

Test input validation for all fields in the search form.

Confirm the "Search" button is visible and functional after inputting search parameters.

Verify that clicking the "Search" button processes the request and displays results.

Test the system's response when no flights match the search criteria.

Assess search functionality under simulated high user traffic.

Ensure that search parameters are transmitted securely to the server.

Test the search form's responsiveness on various devices.

Verify the system's behavior when an invalid date range is entered.

Test the response when an unavailable or unrecognized destination is entered.

Assess how the system handles session timeouts during the search process.

Verify the autocomplete functionality for departure and arrival location fields.

Ensure users can modify search parameters before initiating the search.

Verify the flight search form's accessibility with screen readers.

Verify access to the flight search from the homepage.

Confirm direct access to the flight search page.

Test submission of a flight search with valid parameters.

Ensure the "Search" button is displayed after form completion.

Test search functionality with invalid departure or arrival locations.

Verify the system validates date inputs correctly.

Ensure passenger number inputs are validated.

Test the selection of different classes of service.

Evaluate system performance under high load.

Test for SQL injection and XSS vulnerabilities in the search form.

Verify accuracy and relevance of search results.

Test system behavior under poor network conditions.

Test system response when the flight search service is unavailable.

Verify the functionality of resetting the search form.

Ensure the flight search works across different web browsers.

Assess usability and responsiveness on mobile devices.

Test the system's ability to prevent automated bot submissions.

Verify the basic functionality of the destination autofill suggestions.

Ensure the suggestions are accurate and relevant to the typed input.

Assess the speed at which suggestions are displayed.

Test the functionality of selecting a suggestion.

Verify the suggestions list collapses after a selection is made.

Test how the system handles input of uncommon or less popular locations.

Verify the system's behavior when no matching suggestions can be found.

Assess autofill behavior when special characters are used in the input.

Ensure the autofill suggestion feature works well on mobile devices.

Verify that users can navigate through suggestions using keyboard arrows.

Test the visibility of suggestions with lengthy location names.

Evaluate the performance of the suggestion feature under high system load.

Ensure that data transmitted during the autofill process is secure.

Test the autofill suggestion feature under slow network conditions.

Verify autofill suggestions for locations with non-English characters.

Verify the calendar view displays price differences for dates.

Test the functionality of selecting a date based on displayed prices.

Confirm the search form updates correctly after selecting a date.

Evaluate the load time and responsiveness of the flexible dates calendar.

Ensure the flexible dates feature is usable on desktops, tablets, and smartphones.

Test behavior when no price differences are available for nearby dates.

Verify the accuracy of displayed price differences.

Check the visibility and clarity of the flexible dates option.

Ensure proper handling of dates selected outside of an available or sensible range.

Test the interaction between flexible dates and other search form elements.

Verify users can navigate and select dates using both keyboard and mouse.

Assess how the system refreshes or updates price differences.

Ensure secure transmission of data when using the flexible dates feature.

Test the feature's performance under simulated high user traffic.

Verify how the system handles failures to load the flexible dates calendar.

Verify filters are displayed correctly on the search results page.

Test the ability to filter results based on stopovers.

Test filtering search results by specific airlines.

Test the ability to filter results based on flight duration.

Test filtering search results based on price range.

Verify the application of multiple filters simultaneously.

Test the ability to reset filters to their default states.

Assess the performance impact of applying filters on search results.

Ensure filters are easily usable on mobile devices.

Verify the system's behavior when no flights match the selected filter criteria.

Test if filter selections persist during the same session.

Ensure there is immediate and clear feedback when filters are applied.

Test the accessibility of filter options for users with disabilities.

Test filtering functionality with rare or unusual criteria.

Ensure secure transmission of filter criteria and search results data.

Verify detailed flight information is displayed correctly.

Test the ease of accessing flight details from search results.

Ensure flight details are easily readable and navigable on mobile devices.

Assess the speed at which flight details load.

Verify the accuracy of the displayed flight details.

Test clarity and completeness of layover information.

Ensure information on flight amenities is available and accurate.

Handle missing or incomplete flight details appropriately.

Test the functionality of returning to the search results from flight details.

Verify the ability to share flight details.

Test the print functionality for flight details.

Verify integration with flight comparison features.

Ensure secure transmission of flight details data.

Test the display of flight details in multiple languages.

Ensure flight details are accessible to users with disabilities.

Verify users can easily access options to modify search parameters.

Ensure all search parameters can be accurately updated in the dynamic search form.

Verify validation of new inputted search parameters.

Assess the speed and responsiveness of displaying updated search results.

Test if the system retains modified search parameters after page refresh.

Ensure users receive clear feedback when search results are updated.

Test the system's response to invalid inputs for modified search criteria.

Ensure modifying search parameters is easy across different devices.

Evaluate the system's performance when modifying search parameters under high load.

Verify the presence and functionality of an option to reset search criteria.

Test the accessibility of modifying search parameters for users with disabilities.

Verify if a confirmation is needed/available before updating the search.

Verify the basic functionality of comparing selected flights.

Ensure all details in the comparison view are accurate.

Test the usability of the flight comparison feature.

Evaluate the performance of loading the comparison view.

Verify users can easily select and deselect flights for comparison.

Test the maximum number of flights that can be compared at one time.

Verify system behavior when selected flights have no significant differences.

Test the ability to compare flights across different departure or return dates.

Ensure key differences between flights are highlighted effectively.

Assess the comparison feature's responsiveness on mobile devices.

Test the comparison feature's accessibility for users with disabilities.

Evaluate how the comparison feature performs under high user traffic.

Verify the presence of a clear option to proceed after comparing flights.

Verify the basic functionality of saving search criteria or specific flights.

Ensure users receive a clear confirmation prompt when saving their search.

Test the accuracy of saved search criteria and specific flight details.

Ensure the save feature is easily accessible and usable across different devices.

Assess the response time when saving search criteria or flights.

Verify the system's behavior during a save operation failure.

Test for clear user feedback after a save operation is successful.

Verify users can easily access and manage their saved searches.

Ensure that saved search data is securely stored and accessed.

Test the persistence of saved searches across multiple sessions.

Assess any limitations on the number of saved searches and notify the user.

Verify the save feature's accessibility for users with disabilities.

Test saving functionality with incomplete search criteria.

Verify the basic functionality of sharing flight details.

Ensure all stated sharing options are available and functional.

Test the functionality of sharing flight details via email.

Assess the usability of sharing flight details on social media platforms.

Ensure that shared flight details are accurate and complete.

Evaluate the response time and performance of the sharing functionality.

Test system behavior when attempting to share flight details to an invalid address.

Verify that users receive confirmation of successful sharing.

Ensure that the sharing process is secure, protecting user privacy.

Test the functionality and accessibility of sharing flight details via direct link.

Ensure the sharing feature is easily accessible and usable across devices.

Assess any limitations or restrictions on the sharing functionality.

Verify the sharing feature's accessibility for users with disabilities.

Verify users can access the booking form for a flight.

Test the completion and submission of the booking form.

Ensure the total cost is calculated accurately.

Test the review and confirmation step before proceeding to payment.

Verify the payment page and submission process.

Confirm the booking confirmation and receipt of confirmation email.

Ensure the booking process is easily navigable on different devices.

Test system behavior in case of payment failure.

Verify the security of personal and payment information submitted during booking.

Assess options for modifying or cancelling the booking before final confirmation.

Test the application of promotional codes and vouchers on the payment page.

Ensure the booking process is accessible to users with disabilities.

Evaluate the performance of the booking system under high user traffic.

Verify error handling and user feedback throughout the booking process.

Details

Users should be able to easily find and access the Flights section from the GlobeTrotter homepage or direct navigation.

The search form should dynamically adjust to user inputs, offering fields for departure and arrival locations, dates, passenger numbers, and class of service.

The form should validate inputs for locations, dates (future dates only), passenger numbers (positive integers), and class of service. Invalid inputs should prompt descriptive error messages.

The Search button should be easily visible and clickable once all necessary search parameters have been entered.

After clicking Search, the system should display a loading indicator followed by the search results page, showing available flights based on entered parameters.

If no flights are available for the given search parameters, the system should display a relevant message or suggest alternative search parameters.

Simulate high traffic to test if the flight search functionality remains consistent in performance, ensuring search results are returned within a reasonable timeframe without errors.

Search requests should use HTTPS to encrypt data transmission, protecting user inputs from interception.

The search form should be fully functional and visually appealing across devices of various screen sizes, including desktops, tablets, and smartphones.

Entering a return date that is earlier than the departure date should trigger an error message and prevent the search from being executed.

If a user enters a destination that is not serviced or unrecognized, the system should prompt an error message or suggestions of valid destinations.

If the user's session times out before the search is initiated, the system should either retain the entered search parameters upon session renewal or prompt the user to re-enter them.

As the user begins typing a location, the system should offer autocomplete suggestions that match the input. Selecting a suggestion should populate the field with the chosen location.

Users should be able to change any search parameter (locations, dates, passenger numbers, class) without issues before clicking the "Search" button.

The search form, including all input fields and the Search button, should be accessible to visually impaired users using screen readers, with appropriate labels and instructions.

Ensure users can navigate to the "Flights" section from the GlobeTrotter homepage seamlessly.

Users accessing the Flights page directly (via bookmark or link) are presented with the dynamic search form.

Input valid departure and arrival locations, dates, passenger numbers, and class of service, then click "Search." Verify that search results are appropriately displayed.

After entering search criteria, the "Search" button should be visible and clickable.

Enter non-existent locations and attempt to initiate search. System should prompt with an error or not allow search execution.

Enter departure dates after return dates or past dates and attempt to search. The system should prevent search and display an error message.

Input invalid passenger numbers (e.g., 0, negative numbers) and try to initiate search. System should prevent search and display an error message.

Choose different classes of service (e.g., Economy, Business) and ensure the system processes the choice correctly.

Simulate high traffic and measure the response time and performance of the flight search functionality. Ensure system remains responsive and accurate.

Attempt to input malicious SQL commands or JavaScript into the form fields. The system should sanitize inputs and prevent execution.

Ensure that the search results are accurate based on the input parameters, including correct departure and arrival locations, dates, and class of service.

Simulate poor network connectivity and attempt to initiate a flight search. System should handle the situation gracefully, with appropriate error messaging or retry options.

Simulate a scenario where the backend service for flight search is down. The system should notify the user of the unavailability and possibly provide recommendations or retry options.

After inputting search criteria, use a form reset feature (if available) and ensure all fields are cleared correctly without retaining previous inputs.

Test the flight search functionality on major web browsers like Chrome, Firefox, Safari, and Edge to ensure compatibility and consistent behavior.

Ensure the flight search form is easily usable on smartphones and tablets, with readable text, accessible form fields, and clickable buttons without layout issues.

Implement CAPTCHA or similar mechanisms to differentiate between human and bot traffic, ensuring bots cannot easily automate search submissions.

Start typing a known location in the departure or arrival field. The system should display relevant location suggestions matching the input in real-time.

Type a partial location name, and check if the displayed suggestions accurately match the input, focusing on the relevance of the suggestions to the user's input.

Measure the response time from when a user starts typing to when suggestions are displayed. The suggestions should appear quickly, without noticeable lag, to ensure a smooth experience.

Use keyboard or mouse to select a suggestion from the list. The selected location should be populated in the input field correctly.

Once a suggestion is selected, the suggestions list should collapse automatically, indicating the user's selection has been accepted.

Type the name of an obscure or less commonly searched location. The system should still suggest relevant matches or the closest matches available.

Enter a non-existent location name. The system should handle this gracefully, possibly suggesting the closest matches or indicating no matches were found.

Include special characters in the location input (e.g., dashes, apostrophes). The system should still provide relevant suggestions or handle the input gracefully.

Test the feature on various mobile devices to ensure that suggestions are displayed correctly and can be selected easily, even on touch screens.

Use the keyboard's arrow keys to navigate up and down the list of suggestions and select a location. This should work seamlessly without issues.

Input a long location name to check if the suggestions are still visible and not truncated improperly. Suggestions should be fully visible or appropriately abbreviated.

Simulate a high number of concurrent users to see if the autofill suggestion performance degrades. The feature should still perform well under stress.

Intercept the data request made by the autofill feature to ensure that it is sent over HTTPS and that the data is not exposed to potential security risks.

Throttle the network speed and observe how the autofill feature behaves. It should either show a loading indicator or handle the delay gracefully.

Input locations with non-English characters or special linguistic features (e.g., accents). The system should correctly suggest locations and handle special characters appropriately.

Upon clicking the date field, ensure the calendar view shows price differences for nearby dates, allowing users to make cost-effective decisions.

Users should be able to select a date with displayed price differences, and the system should correctly update the search form with the selected date.

After selecting a date from the flexible dates calendar, the search form should reflect the chosen date accurately in the corresponding date field.

The calendar view with flexible dates and price differences should load quickly and remain responsive, even under slow network conditions or high server load.

The calendar view should be easily navigable and selectable across various devices, with clear visibility of price differences on dates.

If no price differences are available, the system should handle this gracefully, possibly by informing the user or suggesting alternative search strategies.

The displayed price differences for dates should accurately reflect actual prices available, without misleading the user.

The option to use flexible dates should be clearly visible and understandable to users, encouraging its use for finding better prices.

Attempt to select dates that are beyond the range of the displayed price differences or in the past. The system should prevent selection or advise against it.

Changing the date using the flexible dates feature should not adversely affect other already-filled form elements, like destinations or passenger numbers.

Ensure that users can navigate the flexible dates calendar and select dates using keyboard arrows for navigation and enter/space for selection, in addition to mouse clicks.

Determine how the system updates or refreshes price differences, either automatically over time or when the user initiates a new search, ensuring up-to-date information is presented.

Any data transmission related to fetching price differences should be secure, protecting against interception or manipulation.

Simulate high traffic to assess if the feature's performance degrades under load, ensuring that it remains functional and responsive.

Simulate scenarios where the calendar fails to load (e.g., due to network issues or backend errors). The system should inform the user and offer alternatives or retries.

Once a flight search is initiated and results are displayed, check if filters for stopovers, airlines, flight duration, and price are visible and accessible to the user.

Apply the stopover filter to select non-stop flights or flights with one or more stops. Verify that the search results update to only display flights that meet the stopover criteria.

Use the airline filter to select one or multiple airlines. The search results should update to only show flights operated by the selected airlines.

Apply the flight duration filter by selecting a maximum flight duration. Verify that the search results update to display only flights within the specified duration range.

Use the price filter to set a maximum or specific price range. The search results should update to only show flights that fall within the specified price range.

Apply a combination of filters (e.g., stopovers, airlines, price) at the same time. The search results should update to reflect flights that meet all selected filter criteria.

After applying filters, use a reset or clear filters option, if available, and verify that the search results revert to showing all flights without filter criteria applied.

Measure the time taken for the search results to update after applying filters. The update should be quick, ensuring a smooth user experience.

Test the filters on various mobile devices to ensure they are easily accessible, readable, and usable on smaller screens.

Apply filters in such a way that no available flights meet the criteria. The system should inform the user there are no matching results and possibly suggest removing or adjusting filters.

After applying filters and possibly navigating away from the search results page, returning to the page should retain the previously selected filter criteria, assuming the user hasn't initiated a new search.

Upon selecting a filter, there should be an immediate visual or auditory feedback indicating that the filter has been applied, such as the filter section updating or a loading indicator appearing.

Filters should be accessible through keyboard navigation and screen readers, with all text labels and interactive elements properly described for assistive technologies.

Apply filters that are less commonly used or combine multiple specific filters to see if the system can handle edge cases appropriately and still return accurate results.

Any data transmission related to applying filters and retrieving filtered results should be encrypted to prevent interception or manipulation, maintaining user privacy and data integrity.

Upon selecting a flight from the search results, ensure that detailed information (departure/arrival times, duration, layovers, amenities) is displayed accurately and comprehensively.

Confirm that users can easily select a flight from the search results to view its details, with minimal clicks or taps required.

The flight details page should be responsive and provide a good user experience on smartphones and tablets, with text, buttons, and interactive elements sized appropriately for touch interaction.

Flight details should load quickly after a flight selection is made, ensuring a smooth and efficient user experience.

The information displayed in the flight details section should be accurate and match the data of the selected flight, including any layovers and amenities offered.

For flights with layovers, detailed information about the layover(s) – including duration and layover airport – should be clearly presented and easy to understand.

Check if amenities such as in-flight entertainment, meals, and Wi-Fi are listed correctly for the selected flight.

In cases where certain flight details are missing or incomplete, the system should inform the user accordingly, rather than leaving blank spaces or showing incorrect information.

Users should be able to easily navigate back to their search results from the flight details view without losing their place or having to initiate a new search.

Check if users can share flight details through social media, email, or a direct link, and that the shared link directs correctly to the flight details.

Ensure that users can print flight details directly from the page, with the print layout optimized for clarity and readability.

If a flight comparison feature is available, selected flight details should integrate seamlessly, allowing users to compare this flight against others based on the detailed information provided.

Data related to flight details should be securely transmitted, especially if personalization or user-specific data is involved, to prevent interception and ensure privacy.

If the platform supports multiple languages, ensure that flight details are accurately and appropriately displayed in the selected language, including any specific terminologies related to flights.

Flight details should be accessible through screen readers, with proper alt texts for any icons or images, and the page layout should be navigable using keyboard controls for users with mobility impairments.

Users should find it straightforward to locate and use options for modifying search criteria (locations, dates, passenger numbers) from the search results page.

Test updating each parameter individually and in combination (departure and arrival locations, dates, passenger numbers) to ensure the form reflects these changes accurately before initiating a new search.

Input new search criteria, including edge cases like past dates or unrealistic passenger numbers, to test if the system validates these inputs correctly and provides feedback or error messages where appropriate.

After modifying search criteria and initiating a new search, check the speed and efficiency of the updated results display. The system should quickly refresh the results to match the new criteria without unnecessary delay.

After modifying search criteria and initiating a search, refresh the page to see if the search form retains the newly inputted parameters or reverts to the original search.

After modifying search criteria and initiating a new search, the system should provide clear visual or textual feedback that the search has been updated and new results are based on the latest criteria.

Attempt to input invalid search criteria, such as non-existent locations or past dates, to see if the system prevents the search and advises the user to correct the input.

Test the process of modifying search parameters on desktops, tablets, and smartphones to ensure usability is maintained across devices, especially on smaller screens.

Simulate high user traffic to assess if there's any degradation in performance when users modify search parameters and initiate a new search, ensuring the system remains responsive.

Ensure there's a clear and easily accessible option for users to reset search parameters to their defaults, providing a way to start a new search from scratch if desired.

Ensure the functionality to modify search parameters is accessible, supporting keyboard navigation and screen readers, allowing all users to easily update their search criteria.

Test if the system provides a prompt or confirmation option before initiating the search with new parameters, ensuring users have a chance to review changes before proceeding.

Select multiple flights from search results and initiate a comparison. The system should display a side-by-side comparison highlighting differences in price, duration, layovers, and amenities.

The comparison should accurately reflect each flight's departure and arrival times, total duration, layover details, price, and amenities, with no discrepancies from the search results.

The comparison view should be user-friendly, with clear distinctions between flights. It should be easy to navigate and make selections for a detailed comparison on various devices.

The comparison view should load quickly after selecting flights for comparison, without significant delays, ensuring a smooth user experience.

Users should be able to modify their selection of flights for comparison easily, adding or removing flights from the comparison view without starting over.

Determine and verify the maximum number of flights that can be selected for a single comparison. The system should limit selections appropriately and inform users of this limit.

If selected flights for comparison are very similar or identical (e.g., same flight different booking class), the system should still display the comparison or suggest reviewing identical flights.

If the system allows, compare flights on different dates side-by-side to assess how date changes affect price, duration, and other factors.

In the comparison view, significant differences (e.g., price, duration) should be highlighted or emphasized to help users make informed decisions.

The flight comparison view should be easily navigable and readable on smartphones and tablets, with all relevant information accessible.

Ensure that the comparison feature is accessible, supporting keyboard navigation and screen readers, and that all textual and visual information is conveyed appropriately.

Simulate high traffic to see if there's any degradation in the performance of the comparison feature, ensuring that it remains functional and responsive.

After reviewing a comparison, users should have a clear and straightforward option to select a flight and proceed with booking or return to search results.

Perform a flight search, select the option to save the search criteria or specific flights, confirm the action, and ensure the information is saved to the user's profile.

Upon choosing to save search criteria or specific flights, a confirmation prompt should appear, asking the user to confirm their decision to save for future access.

After saving, verify that the saved search criteria or specific flight details accurately reflect the user's initial selection and are stored correctly in their profile.

The option to save search criteria or flights should be clearly visible and functional on desktops, tablets, and smartphones, with intuitive access and confirmation steps.

The process of saving search criteria or specific flights should be quick and efficient, with minimal delay between user confirmation and the system saving the information.

Simulate conditions that could lead to a failure in saving the search (e.g., network issues, server errors) and ensure the system provides clear feedback or retry options to the user.

Upon successfully saving search criteria or flights, the user should receive clear feedback indicating the save was successful, possibly with a link or direction to where they can view it.

Users should be able to view their saved searches or flights in their profile, with options to review, delete, or possibly modify and search again based on saved criteria.

Check that the saved search information is stored securely within the user's profile, with proper authentication required for access and manipulation to protect user privacy.

Saved search criteria or flights should remain accessible in the user's profile across different sessions, ensuring that users can log out and back in without losing their saved information.

If there's a limit to the number of searches or flights a user can save, test that this limit is enforced and that users are notified when reaching it.

Ensure the functionality to save searches is accessible, supporting keyboard navigation and screen readers, and providing all necessary textual descriptions for assistive technologies.

Attempt to save a search when some criteria are not fully specified or are ambiguous to see how the system handles partial information.

Select a flight to share, choose a sharing method (email, social media, direct link), and complete the sharing process. Verify that the flight details are correctly formatted and sent.

The options to share flight details via email, social media, and direct link should be easily accessible and work as intended, guiding the user through the necessary steps for each method.

Choose to share flight details via email, input recipient email addresses, and send. Verify that the email is received with the flight details accurately included.

Select the option to share flight details on social media, choose a platform (if multiple options are provided), and share. Ensure the process is seamless and the details are posted correctly.

Regardless of the sharing method, the flight details shared should match exactly what the user intended to share, including all relevant flight information.

Sharing actions should be processed quickly, without significant delays, ensuring a smooth user experience.

Attempt to share flight details via email to an incorrectly formatted email address. The system should validate the address and prompt the user to correct it before proceeding.

After sharing flight details, the user should receive confirmation that the information has been successfully sent or posted, providing reassurance that the action was completed.

When sharing flight details, especially via email or direct link, ensure that the information is transmitted securely and that user privacy is maintained.

Ensure that the direct link generated for sharing flight details is functional, accessible to the recipient, and correctly displays the flight information.

The feature to share flight details should be straightforward and user-friendly on desktops, tablets, and smartphones, with a consistent experience across platforms.

If there are any limitations on the number of times flight details can be shared or restrictions based on the recipient's location or platform, these should be clearly communicated.

Ensure that the functionality to share flight details is accessible, supporting keyboard navigation and screen readers, with all necessary textual descriptions for assistive technologies.

After selecting a flight to book, the system should display the booking form including fields for passenger details, seat and meal preferences, and additional services.

Fill out the booking form with all necessary details and additional services, then submit. The system should validate the input and proceed to the total cost calculation.

After selecting additional services, verify the system calculates the total cost correctly, including the flight's base price and added services. The total cost should be presented clearly before proceeding to payment.

Users should be able to review the total cost and details of their booking before confirming they wish to proceed to payment. There should be an option to go back and edit the booking if necessary.

On the payment page, input payment information (credit card details, billing address) and any promotional codes or vouchers. Submit the payment and verify that the system processes it correctly.

After payment submission, the system should display a booking confirmation page and send a confirmation email to the user, including flight details, booking reference, and check-in instructions.

The booking form, payment page, and confirmation steps should be user-friendly and accessible on desktops, tablets, and smartphones, without any layout or functionality issues.

Simulate payment failures (e.g., declined credit card, incorrect billing information) to ensure the system provides clear feedback and options to retry or change payment method.

All personal and payment information entered during the booking process should be encrypted and securely transmitted to prevent unauthorized access or data breaches.

Before final payment submission, verify that users have the option to modify or cancel their booking, and check the ease and clarity of accessing these options.

Apply promotional codes or vouchers during the payment process and verify that discounts are correctly applied to the total cost. Ensure the system rejects invalid or expired codes with appropriate feedback.

The booking form, payment process, and confirmation steps should be accessible, supporting keyboard navigation and screen readers, with all form fields and instructions clearly labeled.

Simulate high traffic to assess if there's any degradation in the performance of the booking process, particularly in form submission and payment processing, ensuring the system remains responsive and functional.

Throughout the booking process, simulate various errors (e.g., missing mandatory fields, server errors) to test the system's error handling and ensure clear, constructive feedback is provided to the user for correction.

10

10

10

10

10

10

Bug Num
15
16
17
18
19
20
21

Bug Name

Autofill nesprávných letišť

Výběr data v minulosti

Filtr přímých letů

Filtr cenového rozmezí

Skryté detaily letů

Tlačítko pro rezervaci letu

Ztráta uložených letů

Bug Description

Vyhledávací formulář poskytuje možnost autofill pro vstupní pole letišť. Chyba se projevuje tím, že návrhy někdy obsahují nesprávná nebo neexistující letiště.

Kalendář umožňuje uživatelům vybrat datum odletu, které je v minulosti, což by mělo být omezeno na aktuální a budoucí data.

Filtr pro vyhledávání přímých letů někdy zobrazí lety s mezipřistáními, což je v rozporu s očekáváním uživatele.

Cenový posuvník nesprávně filtruje lety podle ceny, ať už nezobrazuje nejlevnější nabídky nebo nerespektuje nastavený cenový limit.

Rozbalovací sekce s detaily letů má chybu, která někdy způsobí, že se obsah nezobrazí a zůstane skrytý před uživatelem.

Tlačítko „Rezervovat let“ občas nereaguje nebo způsobuje pád stránky, což uživatele znemožňuje dokončit rezervaci.

Funkce pro ukládání vyhledaných letů je nespolehlivá a někdy dojde ke ztrátě informací o uložených letech kvůli problémům se správou relací.

Bug Type	Found	Test Scenario ID
Funkční chyba	Y	33,34
Funkční chyba	Y	56
Funkční chyba	Y	64
Funkční chyba	Y	67
Vizuální chyba	Y	78
Funkční chyba	Y	145
Konfigurační chyba	Y	118,120

Comment

Time to generate:

ID	Scenario name
1	Accessing Hotel Search
2	Input Validation for Search Criteria
3	Successful Hotel Search
4	Search Performance under Load
5	No Results Scenario
6	Security of Search Data Transmission
7	Usability of Hotel Search on Various Devices
8	Accessibility of Hotel Search Form
9	Search Results Sorting Default
10	Interactive Elements in Search Results
11	Refresh and Persistence of Search Criteria
12	Negative: Special Characters in Search Criteria
13	Filtering Options for Search Results
14	Error Handling for Failed Searches
15	Multilingual Support for Hotel Searches
16	Accessing Interactive Map
17	Display and Clarity of Hotel Locations on the Map
18	Interaction with Map Markers
19	Exploring Nearby Attractions
20	Performance of the Interactive Map
21	Map Functionality on Various Devices
22	Information Accuracy on Map Selections
23	Negative Scenario: Map Loading Failure
24	Security of Data in Interactive Maps

25	User Preferences for Map View
26	Accessibility of Interactive Maps
27	Filtering Options on the Map
28	Saving and Sharing Map Views
29	Multilingual Map Information
30	Real-time Updates on Map
31	Accessing Hotel Details
32	Accuracy of Hotel Details
33	Usability of Hotel Details Page
34	Performance of Loading Hotel Details
35	Interactive Elements in Hotel Details
36	Negative Scenario: Incomplete Hotel Details
37	Security of Hotel Details Page
38	Accessibility of Hotel Details Information
39	Linking to Booking From Hotel Details
40	Error Handling for Hotel Details Retrieval
41	Multilingual Support for Hotel Details
42	Real-time Availability and Pricing
43	Hotel Contact Information Accessibility
44	Review and Ratings Display in Hotel Details
45	Comparison and Wishlist Features from Details Page
46	Sorting by Price
47	Sorting by Star Rating
48	Sorting by Guest Reviews
49	Usability of Sorting Options
50	Performance of Sorting Function

51	Negative Scenario: Tied Sorting Criteria
52	Persistence of Sorting Preference
53	Sorting and Filtering Interaction
54	Real-time Updates in Sorted Results
55	Accessibility of Sorting Features
56	Sorting Option Memory Across Searches
57	Error Handling for Sorting Functionality
58	Visual Indicators for Active Sorting
59	Custom Sorting Preferences
60	Security of Sorting Requests Room Selection and
61	Services
62	Display of Total Cost
63	Usability of Booking Process on Various Devices
64	Payment Information Submission and Validation
65	Successful Booking Confirmation
66	Performance of the Booking System
67	Negative Scenario: Booking with Expired Availability
68	Security of Payment and Personal Information
69	Modification and Cancellation Options Post- Booking
70	Error Handling During Booking
71	Accessibility of the Booking Interface
72	Application of Promotional Codes or Discounts
73	Real-time Room Availability Check
74	Booking Process Interruption and Recovery
75	Multilingual Support for Booking Process

76	Access Booking Details
	Modify Booking Options
77	Display
	Modification Process
78	Usability
	Cancellation Process
79	Confirmation
	Real-time Availability Checks
80	for Modifications
	Performance of Modification
81	and Cancellation
	Negative Scenario:
82	Modification Limitations
	Negative Scenario:
83	Cancellation Fees
	Security of Modification and
84	Cancellation
	Accessibility of Modification
85	and Cancellation
	Error Handling in
	Modification and
86	Cancellation
	Refund Process for
87	Cancellations
	Update Notifications for
88	Modifications
	Verification of User Identity
89	for Changes
	User Feedback for
90	Unsuccessful Changes
91	Accessing Profile Settings
	Setting Preferences for Hotel
92	Amenities
	Choosing Preferred
93	Locations
	Specifying Budget Range
94	Preferences
	Confirmation of Saved
95	Preferences
	Usability of Preference
96	Settings on Various Devices
	Performance of Saving
97	Preferences
	Negative Scenario:
	Incomplete Preference
98	Settings
99	Security of Preference Data
	Accessibility of Preference
100	Settings

101	Error Handling in Preference Management
102	Preference Influence on Search Results
103	Modifying and Deleting Preferences
104	Profile Preferences Memory Across Sessions
105	User Feedback for Preference Application
106	Accessing Booking History
107	Review Option Visibility
108	Submitting Hotel Reviews and Ratings
109	Moderation and Display of Reviews
110	Usability of Review Submission Interface
111	Performance of Review Submission Process
112	Negative Scenario: Incomplete Review Submission
113	Security of Review Data
114	Accessibility of Review Features
115	Error Handling in Review Process
116	Confirmation of Review Moderation Outcome
117	Editing and Deleting Submitted Reviews
118	Impact of Reviews on Hotel Ratings
119	Review Submission Deadlines
120	User Feedback for Unsuccessful Review Actions
121	Accessing Personalized Recommendations
122	Accuracy of Recommendations Based on Past Activity
123	Usability of Recommendations Interface
124	Option to Book or Learn More from Recommendations

125	Real-time Availability and Pricing in Recommendations
126	Performance of Recommendation System
127	Negative Scenario: No Recommendations
128	Security of Recommendation Data
129	Accessibility of Recommendations Section
130	Error Handling for Recommendations Feature
131	Feedback Mechanism for Recommendations
132	Updating Preferences Based on Recommendations
133	Influence of New Searches and Bookings on Recommendations
134	Opt-out Option for Personalized Recommendations
135	Promotion and Deals in Recommendations
136	Accessing Special Offers Section
137	Display of Offers and Deals
138	Offer Details and Restrictions
139	Booking Process with Special Offers
140	Usability of Offers Interface on Various Devices
141	Performance of Loading Offers
142	Negative Scenario: Expired or Invalid Offers
143	Security of Promotional Data
144	Accessibility of Special Offers Section
145	Error Handling in Offers Selection
146	Confirmation of Discounted Booking
147	Modification and Cancellation of Discounted Bookings
148	Real-time Offer Updates

Feedback Mechanism for
149 Offer Relevance

150 Comparison of Offers

15-25 minutes

Description

Verify the user can access the hotel search form.

Test validation of input fields in the hotel search form.

Confirm successful search functionality based on given criteria.

Evaluate search functionality performance under high load.

Test system response when no hotels match the search criteria.

Ensure search criteria are transmitted securely over the network.

Assess the usability of the hotel search form across different devices.

Verify the hotel search form is accessible to users with disabilities.

Verify the default sorting of search results.

Test the functionality of interactive elements in search results.

Verify search criteria are retained after a page refresh.

Test the system's handling of special characters in search criteria.

Ensure filtering options are available and functional for hotel search results.

Test the system's response to a failed search operation.

Verify hotel search functionality supports multiple languages.

Verify users can access the interactive map from hotel search results.

Ensure hotel locations are clearly marked and identifiable on the map.

Test the interaction with hotel location markers on the map.

Verify users can explore nearby attractions in addition to hotel locations.

Assess the loading time and responsiveness of the interactive map.

Ensure the interactive map is functional and usable across different devices.

Confirm the accuracy of information displayed for hotels and attractions on the map.

Test system behavior if the interactive map fails to load.

Verify the security of data transmission within the interactive map feature.

Test if user preferences for map view (e.g., satellite, terrain) are respected.

Ensure the interactive map is accessible to users with disabilities.

Test the application of filtering options directly on the interactive map.

Verify users can save or share their current map view and selections.

Test the multilingual support for information displayed on the map.

Ensure the map displays real-time availability and prices for hotels.

Verify users can view detailed information for a selected hotel from search results.

Ensure all displayed hotel details are accurate and comprehensive.

Assess the usability and layout of the hotel details page across various devices.

Evaluate the loading time for the hotel details page.

Test the functionality of interactive elements within the hotel details page.

Verify system behavior when certain hotel details are missing or incomplete.

Ensure secure transmission of data on the hotel details page.

Verify hotel details page is accessible to users with disabilities.

Test the transition from viewing hotel details to initiating a booking.

Assess the system's error handling when hotel details cannot be retrieved.

Verify multilingual support for the hotel details page.

Ensure hotel details include real-time room availability and pricing information.

Confirm accessibility of hotel contact information for inquiries.

Test the display of user reviews and ratings on the hotel details page.

Verify users can compare hotels or add them to a wishlist directly from details page.

Verify users can sort hotel search results by price.

Test sorting functionality based on hotel star ratings.

Confirm hotels can be sorted based on guest review scores.

Assess the usability and accessibility of sorting options across devices.

Evaluate the performance and speed of sorting operations.

Test system behavior when multiple hotels have identical sorting criteria values.

Verify if the user's sorting preference is retained during the session.

Assess how sorting interacts with applied filters.

Ensure sorted results reflect real-time availability and pricing.

Test the sorting feature's accessibility for users with disabilities.

Verify if the sorting option selected is remembered across different searches.

Test error handling and user feedback for sorting functionality.

Confirm active sorting criteria are visually indicated.

Explore the possibility of allowing users to create custom sorting preferences.

Ensure the security of data transmission during sorting operations.

Verify the process of selecting a room and adding services.

Ensure total cost is correctly calculated and displayed.

Assess the booking process's usability across devices.

Test the payment information submission and validation process.

Confirm the user receives a booking confirmation.

Evaluate the booking system's performance, especially during payment processing.

Verify system behavior when selected room is no longer available.

Ensure secure handling of all personal and payment information.

Test the availability of modification or cancellation options after booking.

Assess the system's error handling capabilities during the booking process.

Verify the booking interface is accessible to users with disabilities.

Test the application and validation of promotional codes or discounts.

Ensure system checks and displays real-time room availability during selection.

Test the system's recovery from interruptions during the booking process.

Verify the booking process supports multiple languages.

Verify users can access their booking details.

Ensure options to modify or cancel are clearly displayed.

Test the usability of the booking modification process.

Confirm the user receives confirmation upon cancellation.

Verify availability is checked in real-time during booking modifications.

Assess the response time for booking modifications and cancellations.

Test limitations on booking modifications.

Verify the system informs users of any cancellation fees.

Ensure all modification and cancellation requests are securely processed.

Verify modification and cancellation processes are accessible.

Assess error handling during booking changes.

Test the refund process for cancellable bookings.

Confirm users receive updated notifications for any booking modifications.

Ensure the system verifies user identity before allowing booking changes.

Provide clear user feedback for unsuccessful booking changes.

Verify users can easily access their profile settings to manage preferences.

Test the ability to set preferences for hotel amenities.

Confirm users can select preferred locations for hotel searches.

Enable users to define their budget range for hotel searches.

Ensure users receive confirmation after saving their search preferences.

Assess the usability of setting preferences across different devices.

Evaluate the performance and response time when saving preferences.

Test system behavior when preferences are incomplete or contradictory.

Ensure the security of personal preference data.

Verify the accessibility of preference settings for users with disabilities.

Assess the system's error handling during preference management.

Test how saved preferences influence future hotel search results.

Confirm users can modify or delete their saved search preferences.

Verify if the system remembers saved preferences across different sessions.

Provide user feedback indicating when preferences are applied to searches.

Verify users can easily access their booking history to find past stays.

Ensure the option to review past bookings is clearly visible.

Test the process of submitting a review and rating for a hotel stay.

Confirm submitted reviews are moderated and displayed on the hotel's page.

Assess the usability of the review submission form across devices.

Evaluate the performance and response time of the review submission process.

Verify system behavior when review submission is incomplete.

Ensure the security of data transmission during review submission.

Verify the review features are accessible to users with disabilities.

Test error handling and user feedback during the review process.

Confirm users receive notification about the outcome of review moderation.

Test the ability for users to edit or delete their submitted reviews.

Assess how individual reviews impact overall hotel ratings.

Verify if there are deadlines for submitting reviews after a hotel stay.

Provide clear user feedback for unsuccessful review actions (edit, delete).

Verify users receive personalized hotel recommendations upon visiting the site.

Ensure recommendations are relevant based on the user's past searches and bookings.

Assess the usability of the recommendations section across devices.

Verify users can directly book or learn more about recommended hotels.

Confirm that recommendations reflect real-time availability and current pricing.

Evaluate the loading time and responsiveness of the recommendation system.

Test system behavior when no personalized recommendations can be generated.

Ensure the security of user data used in generating personalized recommendations.

Verify the recommendations section is accessible to users with disabilities.

Assess error handling and user feedback within the recommendation feature.

Test if users can provide feedback on the relevance of recommendations.

Confirm users can update their preferences based on recommendations received.

Verify new searches and bookings influence future recommendations.

Test the availability of an opt-out option for users who prefer not to receive recommendations.

Ensure promotions and special deals are highlighted in personalized recommendations.

Verify users can easily find and access the Special Offers section.

Ensure all available offers and deals are displayed clearly.

Confirm users can view detailed information about offers, including any restrictions.

Test the booking process when a special offer is selected.

Assess the usability of the Special Offers section across devices.

Evaluate the loading time and responsiveness of the Special Offers section.

Verify system behavior when an offer is expired or invalid.

Ensure the security of data transmission in the offers and booking process.

Verify the Special Offers section is accessible to users with disabilities.

Test error handling and user feedback when selecting offers.

Confirm users receive confirmation of bookings made with special offers.

Assess options for modifying or cancelling bookings made with special offers.

Ensure the system updates special offers in real-time.

Test if users can provide feedback on the relevance of special offers.

Verify users can compare multiple special offers.

Details

User navigates to the "Hotels" section. The system should display the hotel search form.

Enter invalid data (e.g., past dates for check-in/check-out, special characters in city/landmark/hotel name) to ensure the system validates inputs properly.

Input valid search criteria and submit. Verify that the list of hotels displayed matches the criteria.

Simulate high traffic to assess if search functionality slows down or fails under heavy load, ensuring performance is scalable.

Enter search criteria that match no hotels to ensure the system provides a meaningful response or suggestions.

Monitor the data transmission when submitting the hotel search form to ensure it is encrypted and secure (HTTPS).

Test the hotel search functionality on desktop, tablet, and mobile devices to ensure form is responsive and usable on all screen sizes.

Use screen readers and keyboard navigation to ensure all elements of the hotel search form are accessible.

Without applying any manual sorting, check the default order of hotel search results (e.g., by relevance, price, or rating).

Ensure that links, buttons, or filters in the hotel search results are clickable and lead to the correct actions or pages.

After performing a search, refresh the page to check if the search criteria and results persist.

Enter special characters in search fields to check if the system sanitizes inputs to prevent SQL injection or XSS attacks.

After receiving search results, apply various filters (e.g., price range, star rating) and verify that results update accordingly.

Simulate a backend failure (e.g., database unreachable) to ensure the system gracefully handles errors and informs the user.

Change the website language and perform a hotel search to ensure search functionality works correctly across supported languages.

After performing a hotel search, users should be able to choose to view the interactive map showing hotel locations.

Hotels should be marked on the map with clear icons or markers, making it easy to identify different hotels.

Clicking or tapping on a hotel marker should display detailed information about the hotel, such as name, rating, and brief amenities.

The map should not only show hotel locations but also mark nearby attractions, allowing users to explore what's around the hotels.

The interactive map should load promptly and allow for smooth zooming and panning without lag or glitches.

Test the map on desktops, tablets, and smartphones to ensure it is responsive and interactive elements work properly on touch screens.

The information provided when a hotel or attraction is selected on the map should be accurate and match the details listed in search results.

Simulate scenarios where the map might fail to load (e.g., network issues) to check if the system provides useful feedback or alternative options.

Ensure that any data requests or submissions through the interactive map (e.g., viewing hotel details) are securely handled.

Users should be able to change the map view according to their preferences, and the system should remember these settings during the session.

The map feature should be navigable and usable with keyboard shortcuts and screen readers, providing accessible information for all users.

Users should be able to apply filters (e.g., price range, star rating) and see the map update to reflect these filters.

The system should offer options to save the current map view or share it with others, retaining the user's zoom level and selected locations.

Information about hotels and attractions on the map should be available in multiple languages, matching the user's website language preference.

The map should reflect real-time changes in hotel availability and pricing, providing users with up-to-date information.

After selecting a hotel from the search results, ensure that the detailed information page loads with all relevant hotel information.

Check that room amenities, dining options, available services, and cancellation policies are accurately listed and match the hotel's offer.

The hotel details page should be easily navigable and visually appealing on desktop, tablet, and mobile devices.

The hotel details should load quickly, ensuring a smooth user experience without significant delays.

Interactive elements like photo galleries, booking buttons, or maps should function correctly and enhance the user's understanding of the hotel.

In cases where some hotel details are unavailable, the system should notify the user appropriately, without leaving them confused.

Any data transmission, especially for booking or inquiry forms on the hotel details page, should be encrypted and secure.

The hotel details page should support screen readers and keyboard navigation, ensuring all information is accessible.

Users should be able to easily navigate from the hotel details page to the booking interface with the selected hotel pre-loaded.

Simulate a failure to retrieve hotel details (e.g., due to server issues) to check if the system provides useful feedback or alternatives.

The hotel details should be available in multiple languages, aligning with the user's language preferences on the website.

The details page should display up-to-date information on room availability, pricing, and any special offers or discounts.

Hotel contact details (e.g., phone, email) should be readily available for users who have questions or need further information.

User reviews and ratings should be visible, providing insights into guest experiences and helping inform user decisions.

Features allowing users to compare hotels or save them to a wishlist for future reference should function seamlessly from the details page.

Users should have the option to sort hotels from low to high price and vice versa, and the search results should update accordingly.

Select the option to sort hotels by star rating (highest to lowest and vice versa), ensuring the results reflect the chosen order.

Users should be able to sort hotels from highest to lowest review scores and vice versa, with results updating to display accordingly.

Sorting options should be clearly visible and easily selectable on desktop, tablet, and mobile interfaces.

Applying any sorting criterion should quickly reorder the search results without significant delay.

When hotels have the same price, star rating, or guest review score, verify the system sorts them in a consistent secondary order.

After sorting the search results and navigating away, returning to the results should preserve the user's sorting preference.

Apply filters (e.g., location, amenities) and then sort the results to ensure both sorting and filtering criteria are respected.

Sorted hotel lists should update to show the most current availability and pricing, especially when sorted by price.

Sorting options should be fully accessible with keyboard navigation and screen readers, with clear labels for all options.

After conducting a search, sorting the results, and performing a new search, check if the system remembers the previously selected sorting option.

In case of an error during sorting (e.g., server issue), the system should provide clear feedback and allow users to retry.

The currently active sorting criterion should be highlighted or otherwise visually indicated to inform users of the active sort order.

Test if users can combine sorting criteria (e.g., price and then by rating) to customize how search results are ordered.

Data involved in sorting requests, especially if user-specific preferences are saved, should be securely transmitted and handled.

User selects a hotel, chooses a room type, and adds additional services (e.g., breakfast, parking). The selections should be accurately reflected.

After selecting room and services, the system should display the total cost, including all additional services and taxes.

The booking process, including room selection and payment, should be user-friendly and functional on desktop, tablet, and smartphone screens.

User enters payment information which the system validates. Errors should prompt correct user feedback for correction.

Upon completing payment, the system should confirm the booking and send a confirmation email with details and a reference number.

The booking and payment process should be smooth and efficient, without significant delays, ensuring a positive user experience.

Attempt to book a room that has become unavailable or sold out during the selection process. System should notify and offer alternatives.

All user data and payment information entered during booking should be encrypted and securely processed to protect against data breaches.

After booking, users should have clear options to modify or cancel their booking, with any changes or cancellations accurately processed.

Simulate errors (e.g., payment failure, network issues) during booking to ensure the system provides clear, actionable feedback to the user.

The booking process should be fully accessible, with support for screen readers, keyboard navigation, and clear labels for all form fields.

Users should be able to apply promotional codes at checkout, with the system correctly applying the discount to the total cost.

When selecting a room type, the system should verify and display current availability to prevent overbooking or selection of unavailable rooms.

Simulate accidental page refresh or network disconnection to check if the system saves the booking progress or allows easy recovery.

The booking process, including form fields and instructions, should be available in the user's selected language to ensure clarity and ease of use.

Users should easily navigate to their booking via their profile or a direct link in the confirmation email and view their booking details.

The booking details page should clearly present users with options to modify or cancel their booking.

Users selecting to modify their booking should find the process intuitive, with the ability to change dates, services, or guest details.

Upon canceling a booking, users should receive immediate confirmation on-screen and via email.

When a user attempts to modify their booking, the system should check and display real-time room availability to ensure changes are viable.

Modifications and cancellations should be processed quickly, reflecting changes without delay to ensure a smooth user experience.

Attempt to modify a booking past any set deadlines or limits to ensure the system restricts changes appropriately and informs the user.

If cancellation fees apply, the system should clearly inform the user before they confirm the cancellation.

Data transmission during booking modifications or cancellations should be encrypted to protect user information.

The processes for modifying or canceling a booking should be accessible, supporting screen readers and keyboard navigation.

If errors occur during modification or cancellation (e.g., network issues), the system should provide clear feedback and recovery options.

For bookings eligible for refunds upon cancellation, verify the refund process is initiated and completed as expected.

After a successful booking modification, users should receive an updated confirmation email detailing the changes.

To modify or cancel a booking, the system should require users to verify their identity, such as logging in or providing booking details.

If a modification or cancellation attempt is unsuccessful, the system should clearly explain the reason and suggest next steps.

Users should navigate to their profile settings with ease, finding the section dedicated to search preferences.

Users should be able to select their preferred hotel amenities (e.g., free Wi-Fi, gym access) and save these preferences to their profile.

Allow users to specify preferred locations or destinations and ensure these preferences are saved and reflected in future searches.

Users should be able to set a preferred budget range for hotel searches, influencing the search results to match their budget preferences.

Upon saving new search preferences, users should receive on-screen confirmation and possibly an email notification confirming the update.

The process of setting and saving search preferences should be user-friendly and fully functional on desktops, tablets, and smartphones.

Saving search preferences should be quick and efficient, with immediate feedback provided to the user indicating successful saving.

Attempt to save incomplete or logically contradictory preferences (e.g., high luxury in low budget range) to see how the system handles errors.

All personal preference data saved to the user's profile should be securely stored and transmitted, protecting user privacy.

Setting and saving search preferences should be accessible, supporting keyboard navigation and screen readers for all users.

If errors occur when users manage their preferences (e.g., network issues, validation errors), the system should provide clear feedback.

Conduct a hotel search after setting preferences to confirm that the search results are influenced by the saved preferences.

Users should have the option to change or completely remove their saved search preferences from their profile settings.

Log out and then back into the website to check if the system retains and applies the user's saved search preferences to hotel searches.

When conducting a search, the system should indicate that user preferences are being applied, possibly through a notification or filter summary.

Users should navigate to their booking history through their profile or dashboard, finding a list of past hotel bookings.

For each completed stay listed in the booking history, there should be a visible and accessible option to leave a review and rating.

Users should be able to choose a past booking, submit a numerical rating and written feedback, and successfully submit this review.

After submitting a review, check that it undergoes a moderation process and is subsequently displayed on the hotel's page for other users to see.

The interface for submitting reviews, including rating scales and text inputs, should be user-friendly and functional on desktop, tablet, and mobile screens.

Submitting a review should result in quick and efficient processing, with immediate feedback provided to the user indicating successful submission.

Attempt to submit a review without filling all required fields (e.g., missing rating or feedback) to ensure the system prompts for completion.

Data involved in submitting hotel reviews, including written feedback and ratings, should be securely transmitted and stored.

The process of accessing booking history, selecting a hotel to review, and submitting feedback should be accessible, supporting screen readers and keyboard navigation.

Simulate errors (e.g., network issues, server errors) during review submission to check if the system provides clear, actionable feedback to the user.

After review submission, users should receive notification (e.g., email or dashboard update) indicating whether their review was accepted or needs revision.

Users should have the option to edit or delete their reviews after submission, before and possibly after moderation, with appropriate system support.

Submitted reviews should contribute to the hotel's overall rating in a transparent and predictable manner, reflecting users' experiences accurately.

Check if the system enforces any deadlines by which users must submit their reviews following their stay, and inform users accordingly.

If a user attempts to edit or delete a review and the action is unsuccessful, the system should provide clear reasons and suggest next steps.

Users with a history of searches or bookings should see personalized hotel recommendations on the homepage or Hotels section.

Recommendations should closely match the user's preferences indicated by their past activity, such as preferred locations, amenities, and budget.

The section displaying personalized recommendations should be easily navigable and visually appealing on desktop, tablet, and mobile devices.

Each recommendation should include clear options for users to either book the hotel directly or learn more about it.

Recommended hotels should display up-to-date availability and pricing to ensure users receive accurate and useful information.

The system should quickly generate and display personalized recommendations without significant delays, ensuring a smooth user experience.

For users with minimal or no past activity, the system should handle the absence of recommendations gracefully, possibly suggesting popular hotels.

Data analysis for generating recommendations should be performed securely, protecting user privacy and ensuring data is not misused.

Users relying on screen readers and keyboard navigation should be able to access and interact with the hotel recommendations.

In the event of errors during recommendation generation (e.g., server issues), the system should provide clear feedback to the user.

Users should have the option to provide feedback on recommended hotels, helping to refine the accuracy of future recommendations.

Users should be able to adjust their search preferences directly from the recommendations section to refine the types of hotels suggested to them.

User actions like new searches or recent bookings should dynamically update the algorithm's basis for future recommendations.

Users should have the option to opt-out of personalized recommendations, with the system respecting this choice in subsequent visits.

When applicable, recommended hotels with special offers or promotions should be clearly marked, providing added value to the user's choices.

Users should locate the Special Offers section with ease from the homepage or Hotels section, showcasing current hotel deals and promotions.

The system should list available hotel offers and deals clearly, providing essential details such as discount rates, validity, and terms.

Users selecting an offer should see detailed information, including any restrictions or conditions attached to the deal (e.g., blackout dates, minimum stay).

When a user proceeds to book a hotel with a special offer, the booking process should reflect the discounted rate and apply the deal correctly.

The interface for browsing and selecting hotel offers should be user-friendly and functional on desktops, tablets, and smartphones.

The system should display available offers and deals promptly, without significant delays, ensuring a smooth user experience.

Attempt to select an expired or otherwise invalid offer to see if the system notifies the user and prevents booking at the outdated terms.

Data related to promotional offers and the user's booking information should be securely transmitted, especially during the booking and payment process.

The Special Offers section should support screen readers and keyboard navigation, allowing all users to access and take advantage of hotel deals.

If errors occur while selecting or applying an offer (e.g., network issues), the system should provide clear feedback and recovery options to the user.

Upon completing a booking with a special offer, users should receive confirmation, including details of the offer applied and the final discounted rate.

Test the process for modifying or cancelling a booking that included a special offer, verifying that any changes or cancellations are processed correctly.

The Special Offers section should reflect real-time updates to ensure users see the most current deals available, including last-minute discounts.

Users should have the option to provide feedback on special offers, helping to refine the relevance and presentation of future promotions.

Users interested in multiple offers should be able to compare them side-by-side to determine the best value based on their preferences and booking criteria.

2
2
2
2
2
2
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
3
4
4
4
4
4

10
10

Bug Num
7
8
9
10
11
12
13
14

Bug Name

Nesprávné výsledky vyhledávání

Zpomalení při načítání mapy

Nesprávné zobrazení informací o pokojích

Exponovaná platební data

Oprávnění při přidávání recenzí

Problémy s galerií obrázků hotelu

Problém s načítáním recenzí z externího API

Problém s aktualizací dostupnosti pokojů

Bug Description

Při vyhledávání hotelů podle města, data příjezdu a odjezdu a počtu osob někdy systém zobrazuje výsledky, které neodpovídají zadaným kritériím.

Interaktivní mapa zobrazující polohu hotelů je velmi pomalá při načítání a někdy se dokonce nenačte vůbec, což komplikuje proces výběru hotelu na základě jeho umístění.

Po výběru typu pokoje a pokračování k rezervaci se občas stane, že detaily a cena pokoje neodpovídají původně vybraným specifikacím.

Na stránce pro zadání platebních informací může dojít k odhalení citlivých dat v URL adrese, což představuje vážné bezpečnostní riziko.

Na stránce rezervace hotelů mají uživatelé možnost přidávat recenze na hotely, ve kterých se ubytovali. Implementovaná chyba spočívá v tom, že určitá skupina uživatelů narazí na problém s oprávněními, když se pokusí odeslat recenzi. Systém jim oznámí, že nemají dostatečná oprávnění pro přidání recenze, přestože mají být přihlášení a měli by mít tuto možnost jako součást svých uživatelských práv. Tento problém může být způsoben nesprávnou konfigurací serveru nebo chybou v logice aplikace, která ověřuje oprávnění uživatelů.

Galerie obrázků hotelu trpí problémy s načítáním a někdy zobrazuje obrázky v nesprávném poměru stran nebo vůbec nezobrazuje náhledy.

Stránka hotelu zahrnuje sekci s uživatelskými recenzemi, které jsou načítány prostřednictvím externího API. Chyba se projevuje jako neschopnost stránky správně zobrazit recenze kvůli problémům s timeoutem nebo přerušením síťového připojení. Tento problém simuluje reálné situace, kdy externí služby nejsou dostupné nebo je jejich odezva pomalá, což ovlivňuje celkovou funkcionální a uživatelskou zkušenost na stránce.

Po provedení rezervace by měl systém automaticky aktualizovat dostupnost pokojů v daném hotelu. V tomto případě chyba v databázi způsobí, že ačkoliv uživatel dokončí proces rezervace a obdrží potvrzení, stav dostupnosti pokojů v databázi se neaktualizuje správně. To může vést k situaci, kdy je stejný pokoj nabídnut více uživatelům, což způsobuje konflikty v rezervacích a možné přebookování.

Bug Type	Found	Scenario	Closest Scenario
Funkční chyba	Y	3	
Výkonnostní chyba	Y	20	
Funkční chyba	Y	62	
Bezpečnostní chyba	A	68	
Problém s oprávněním			
	Y	108	
Vizuální chyba	Y	35	
Síťová chyba			
	N		
Databázová chyba			
	Y	42	

Time to generate:

ID	Scenario name
1	Accessing Car Rental Section
2	Input Validation for Rental Details
3	Usability of Date Pickers
4	Performance of Rental Search Interface
5	Enabling Car Preferences Selection
6	Negative Scenario: No Cars Available
7	Security of Data Transmission in Rental Search
8	Accessibility of Rental Search Form
9	Error Handling for Search Interface
10	Saving Search Criteria
11	Performance Under High Traffic
12	Multilingual Support for Rental Search
13	Real-time Availability Checks
14	User Feedback for Successful Search Completion
15	Customization of Rental Search Results
16	Displaying Matched Car Options
17	Filtering Cars by Preferences
18	Selecting Additional Add-ons
19	Usability of Car Selection and Add-ons Interface
20	Updating Booking Summary with Selected Options
21	Performance of Options Selection and Summary Update
22	Negative Scenario: Add-ons Not Available

Security of Data Transmission in Add-ons 23 Selection
Accessibility of Selection 24 Process
Error Handling in Preferences and Add-ons 25 Selection
Confirmation Before 26 Finalizing Selection
Modification of Preferences 27 and Add-ons
Price Adjustment with Add- 28 on Selection
Review of Terms for 29 Selected Add-ons
User Feedback for 30 Unavailable Preferences
Accessing Car Model 31 Descriptions
Completeness of Car Model 32 Information
Usability of Car Description 33 Interface
Performance of Description 34 Loading
35 Visual Aids in Descriptions
Comparison Feature for Car 36 Models
Accuracy and Up-to-date 37 Information
Negative Scenario: Missing 38 Information
Security of Information 39 Display
User Feedback for 40 Descriptions
Accessibility of Descriptions 41 for Disabled Users
Language Support for 42 Descriptions
Error Handling for 43 Descriptions Access
Sharing Option for Car 44 Descriptions
Updating Preferences Based 45 on Descriptions
46 Initiating Price Comparison
Accuracy of Price 47 Information

Comprehensive Provider 48 Coverage
Usability of Price 49 Comparison Interface
Performance of Price 50 Comparison Feature
Negative Scenario: Identical 51 Pricing
Security of Price Comparison 52 Data
Accessibility of Price 53 Comparison
Error Handling in Price 54 Comparison
Provider Ratings and 55 Reviews in Comparison
Updating Comparison Based 56 on Preferences
Sharing Price Comparison 57 Results
58 Saving Price Comparisons
Filter and Sort Options in 59 Price Comparison
Direct Booking from Price 60 Comparison
61 Accessing Rental Summary
Accuracy of Rental Summary 62 Details
Usability of Summary Page 63 on Various Devices
Performance of Summary 64 Generation
Detailed Breakdown of Costs 65 in Summary
Modification Options from 66 Summary
Confirming Accuracy Before 67 Proceeding
Security of Data in Rental 68 Summary
Accessibility of Rental 69 Summary Information
Error Handling for Summary 70 Display
Language Support for Rental 71 Summary
Review of Terms and 72 Conditions in Summary
Saving Rental Summary for 73 Future Reference

Feedback Option for Rental
74 Summary Content
Real-time Updates to Summary for Availability
75 Changes
76 Initiation of Payment Process
Payment Information
77 Submission and Validation
Security Measures in
78 Payment Process
Usability of Payment
79 Interface on Various Devices
Performance of Payment
80 Processing
81 Handling Payment Failures
Confirmation of Booking and
82 Payment
Error Handling in Payment
83 Process
Accessibility of Payment
84 Process
Application of Promotional
85 Codes or Discounts
Modification Options Before
86 Final Payment
Multilingual Support for
87 Payment Process
Receipt and Record of
88 Transaction
Customer Support Access
89 During Payment
Real-time Updates and Final
90 Price Verification
Accessing Booking History
91 for Modification
Display of Modification and
92 Cancellation Options
Usability of Modification
93 Interface
Processing Booking
94 Modifications
Processing Booking
95 Cancellations
Security of Modification and
96 Cancellation Requests
Performance of Modification
97 and Cancellation Actions
Error Handling in
Modification and
98 Cancellation

15-25 minutes

Description

Verify users can easily access the Car Rental section.

Test validation of rental location and date fields.

Assess the usability of date pickers for selecting rental dates.

Evaluate loading time and responsiveness of the car rental search interface.

Confirm car preferences can be selected after initial details are input.

Test system response when no cars are available for selected details.

Ensure all personal and search data is transmitted securely.

Verify the car rental search form is accessible to users with disabilities.

Assess error handling and user feedback within the search interface.

Test if the system allows users to save their rental search criteria.

Evaluate the car rental search functionality under high user traffic.

Verify the rental search interface supports multiple languages.

Ensure the system checks and displays real-time car availability.

Provide clear user feedback upon successful search completion.

Test the ability to customize and filter search results.

Verify that cars matching the user's preferences are displayed.

Test the effectiveness of filtering cars based on user preferences.

Confirm users can choose additional rental options and services.

Assess the usability of the interface for selecting car preferences and add-ons.

Ensure the booking summary updates in real-time to reflect chosen options.

Evaluate the performance and response time of the selection process and summary update.

Test system behavior when certain add-ons are not available at the chosen location.

Ensure the security of data transmission during add-ons selection.

Verify the car preferences and add-ons selection process is accessible.

Test error handling during car preferences and add-ons selection.

Confirm users receive a prompt to review selections before finalizing.

Test the flexibility to modify car preferences and add-on selections.

Ensure price adjustments reflect the selection of additional add-ons accurately.

Verify users can review terms and conditions for selected add-ons.

Provide clear user feedback for unavailable car preferences.

Verify users can access detailed descriptions of car models.

Ensure all relevant details are provided for each car model.

Assess the usability and accessibility of the car model descriptions interface.

Evaluate the loading time and responsiveness when accessing car model descriptions.

Confirm the presence of visual aids (e.g., images, diagrams) in car descriptions.

Test if users can compare different car models directly.

Ensure that car model descriptions are accurate and reflect current offerings.

Verify system behavior when information on certain car models is incomplete.

Ensure the security of data transmission for car model descriptions.

Assess if users can leave feedback or ask questions about car models.

Verify car model descriptions are accessible to users with disabilities.

Test the multilingual support for car model descriptions.

Assess error handling when accessing car model descriptions.

Test if users can share car model descriptions with others.

Verify users can update their rental preferences based on car model descriptions.

Verify users can easily initiate a price comparison for selected cars.

Ensure the price comparison accurately reflects current rates from providers.

Test the breadth of rental providers included in the price comparison.

Assess the usability and readability of the price comparison interface.

Evaluate the loading time and responsiveness of the price comparison feature.

Test system behavior when multiple providers offer identical pricing.

Ensure secure transmission of data within the price comparison feature.

Verify the price comparison feature is accessible to users with disabilities.

Assess error handling and user feedback within the price comparison feature.

Include provider ratings and reviews in the price comparison.

Test if users can adjust preferences and see updated comparisons in real-time.

Verify users can share their price comparison results with others.

Test if users can save their price comparison results for later reference.

Assess the availability of filtering and sorting options within the comparison.

Confirm users can book directly from the price comparison results.

Verify users can access a detailed summary before booking confirmation.

Ensure the rental summary accurately reflects user selections.

Assess the rental summary page's usability across different devices.

Evaluate the loading time and responsiveness when generating the rental summary.

Test if the rental summary provides a detailed breakdown of all costs.

Confirm users can easily modify their rental choices from the summary page.

Verify that users review and confirm the accuracy of the rental summary.

Ensure the security of personal and rental data in the summary phase.

Verify the rental summary information is accessible to users with disabilities.

Assess error handling and user feedback when the rental summary fails to load.

Test the multilingual support for the rental summary page.

Ensure terms and conditions are included or linked in the rental summary.

Test if users can save or print the rental summary for their records.

Verify users can provide feedback on the rental summary content or layout.

Confirm summary updates for any real-time changes in car availability or pricing.

Verify the transition to the payment process after rental summary confirmation.

Test the submission and validation of payment information.

Ensure all payment transactions are secure and encrypted.

Assess the payment interface's usability across different devices.

Evaluate the speed and efficiency of payment processing.

Test system behavior and user guidance during payment failures.

Confirm users receive immediate booking and payment confirmation.

Assess error handling capabilities within the payment process.

Verify the payment process is accessible to users with disabilities.

Test the application and validation of promotional codes during payment.

Confirm users can modify rental details or payment information before finalizing.

Ensure the payment process supports multiple languages.

Verify users receive a receipt or record of the transaction.

Test accessibility of customer support options during the payment process.

Ensure final price verification and any real-time updates are handled in payment.

Verify users can access their booking history to modify or cancel bookings.

Ensure options for modification and cancellation are clearly presented.

Assess the usability of the booking modification interface.

Test the processing and confirmation of booking modifications.

Confirm the system processes and confirms booking cancellations accurately.

Ensure modification and cancellation requests are processed securely.

Evaluate performance and response time for modifications and cancellations.

Test system behavior and user guidance when errors occur.



Details

Users navigate to the "Car Rentals" section from the homepage or navigation menu. The car rental search interface is displayed.

Enter invalid data (e.g., past dates for pick-up/drop-off, special characters in location) to ensure the system validates inputs properly.

Date pickers for pick-up and drop-off should be easy to use, allowing users to select dates without confusion or errors.

The interface, including location input and date selection, should load promptly and respond quickly to user inputs, ensuring a smooth experience.

Once a user inputs rental location and dates, the system should enable selection of car preferences (e.g., car size, type, model).

Enter details leading to no available cars (e.g., a remote location, high-demand dates) to check if the system offers alternatives or notifications.

Data transmission during the car rental search process, especially personal data and search criteria, should be encrypted.

The search form, including location and date inputs, should be fully accessible, supporting screen readers and keyboard navigation.

In case of errors during search (e.g., server issues, invalid inputs), the system should provide clear, actionable feedback to the user.

Users should have the option to save their rental search criteria for future visits, improving ease of use and efficiency.

Simulate high traffic to assess if search functionality slows down or fails, ensuring the system is robust and scalable.

The car rental search interface, including all labels and instructions, should be available in the user's selected language.

When selecting rental details, the system should verify real-time availability of cars to prevent overbooking or selection of unavailable options.

After entering all necessary details and conducting a search, the system should indicate that the search has been successfully completed.

After completing a search, users should be able to customize and filter results based on additional preferences like car features or rental company.

After specifying car type or model preferences, the system should display a list of available cars that match these preferences.

The filtering mechanism should accurately narrow down car options based on selected preferences such as car type, size, or model.

Users should be able to select add-ons like rental car insurance, GPS, child seats, or additional drivers, and have these reflected in the booking summary.

The interface for choosing car preferences and additional services should be user-friendly, clearly categorized, and easy to navigate on all devices.

As users select their car and add-ons, the booking summary should dynamically update to include all selected options and the total price.

Selecting car preferences and add-ons should result in immediate updates to the booking summary without significant delays or performance issues.

Attempt to select add-ons that are unavailable for the specific rental location to check if the system provides alternatives or notifications.

Data involved in selecting car preferences and add-ons, especially personal choices and modifications, should be securely transmitted.

The process of selecting car preferences and add-ons should support accessibility tools like screen readers and keyboard navigation.

If errors occur when selecting preferences or add-ons (e.g., server issues), the system should provide clear feedback and recovery options.

Before proceeding to payment, users should be prompted to review their car and add-on selections, ensuring all choices are intentional.

Users should be able to easily modify their car type, model preferences, and selected add-ons before finalizing the booking.

The total price in the booking summary should adjust accordingly when users select or remove add-ons, accurately reflecting the final cost.

Users should have the option to review detailed terms, conditions, and costs for any chosen add-ons before finalizing their selection.

If selected car preferences are unavailable, the system should offer feedback and suggest available alternatives.

Users exploring available car models should be presented with an option to view detailed descriptions including specs like passenger capacity, and fuel efficiency.

Detailed descriptions for car models should include all necessary information such as passenger capacity, luggage space, fuel efficiency, and any additional features.

The interface for viewing car model descriptions should be user-friendly, easily navigable, and accessible on various devices.

Accessing detailed descriptions of car models should be quick and efficient, with minimal delays in displaying the information.

Car model descriptions should include high-quality images or diagrams to help users visualize the car's features and capacity.

Users should have the option to compare detailed descriptions of multiple car models side by side to aid in decision-making.

The information presented in car model descriptions should be up-to-date, accurately reflecting the features and specifications of each model.

Attempt to access descriptions for car models with missing information to check if the system provides a notification or fills in the gaps with available data.

Data transmission involved in fetching and displaying car model descriptions should be secure, protecting against unauthorized access.

Users should be able to easily provide feedback or inquire more about certain car models directly through the description interface.

The car model descriptions, including textual and visual content, should be fully accessible, supporting screen readers and keyboard navigation.

Car model descriptions should be available in multiple languages, catering to a diverse user base.

If errors occur while accessing descriptions (e.g., server issues), the system should provide clear feedback and alternatives to the user.

Users interested in specific car models should have the option to share these descriptions via social media or email.

After viewing car model descriptions, users should be able to easily adjust their rental preferences or selections based on the information provided.

After selecting a car model and rental period, users should have a clear option to compare prices across different rental providers.

The displayed prices in the comparison should be up-to-date and match the rates provided by each car rental provider for the specified period.

The comparison should include a wide range of providers, offering users a comprehensive view of available options and pricing.

The interface for comparing prices should be user-friendly, with clear distinctions between providers and prices, accessible on all devices.

Accessing the price comparison should be quick and efficient, with minimal delays in displaying the information, ensuring a smooth experience.

If several providers list the same price for a rental, verify the system presents these options equally or provides additional criteria for comparison.

Data involved in fetching and displaying price comparisons, especially if user-specific preferences are included, should be securely handled.

The price comparison feature should be accessible, supporting screen readers and keyboard navigation, with all elements properly labeled.

If errors occur while fetching price data (e.g., server issues, provider API failures), the system should provide clear feedback to the user.

Alongside price information, user ratings and reviews of each provider should be displayed to help users make more informed decisions.

After initially viewing a price comparison, users should be able to change preferences (e.g., rental period, car type) and see updated results.

Users should have the option to share price comparison results via social media, email, or other platforms.

Users interested in reviewing comparison results at a later time should have the option to save these results within their profile.

Users should be able to filter and sort price comparison results based on criteria like price, provider rating, or car features.

After finding the best deal through price comparison, users should have a straightforward path to proceed with booking the selected offer.

After selecting a car and any additional options, users should be guided to a summary page detailing their rental choices, including location, dates, and total price.

The summary should accurately list all user selections, including car model, rental period, chosen add-ons, and correctly calculate the total price.

The rental summary page should be easily readable and navigable on desktop, tablet, and smartphone screens, with clear option to proceed or modify the booking.

Generating the rental summary should be quick and efficient, with minimal delays, ensuring a smooth transition towards booking confirmation.

The summary should include a detailed cost breakdown, including base rental rate, charges for additional options, taxes, and any applicable discounts.

Users should have the option to return to previous steps and modify their car selection or add-ons directly from the summary page.

Before proceeding to payment, users should be prompted to review the rental summary for accuracy and confirm that all details are correct.

Data displayed and transmitted during the summary review, especially personal information and rental details, should be securely handled.

The rental summary, including details and cost breakdown, should be fully accessible, supporting screen readers and keyboard navigation.

In case of errors preventing the summary from displaying correctly (e.g., server issues), the system should provide clear feedback and options to retry.

The rental summary should be available in multiple languages, aligning with the user's website language settings, to ensure clarity and understanding.

Users should be able to review or access the rental agreement terms and conditions directly from the rental summary page before proceeding.

Users should have the option to save or print the detailed rental summary for future reference, including booking confirmation or proof of reservation.

Users should be able to easily provide feedback on the rental summary, such as clarity of information or suggestions for improvement.

If there are any real-time changes affecting car availability or pricing, the summary should dynamically update to reflect these changes before final confirmation.

After confirming the rental details, users should be seamlessly guided to a secure payment gateway for processing the payment.

Users should be able to enter payment details (credit card, billing address) with the system validating this information accurately.

The payment process should implement strong encryption and security measures to protect user payment information and prevent data breaches.

The payment interface should be user-friendly and fully functional on desktops, tablets, and smartphones, ensuring easy payment submission.

Payment processing should be quick and efficient, with users experiencing minimal delays and receiving prompt feedback on the payment status.

In case of payment failure (e.g., due to card decline or network issues), the system should provide clear feedback and options to retry.

Upon successful payment, users should receive on-screen confirmation and a detailed booking confirmation email including rental details.

If errors occur (e.g., server error, timeout), the system should provide users with clear, actionable feedback and recovery options.

The payment process, including form fields and instructions, should be accessible, supporting screen readers and keyboard navigation.

Users should be able to apply valid promotional codes at payment, with the system correctly applying the discount and updating the total cost.

Before completing the payment, users should have the option to go back and modify rental details or correct payment information if needed.

The payment interface and instructions should be available in the user's selected language, ensuring clarity and understanding throughout.

Along with booking confirmation, users should receive a receipt or transaction record, either on-screen or via email, for their records.

Users should have easy access to customer support or help options during payment in case of questions or issues.

Before payment submission, the system should verify the final price, including any last-minute changes or applied discounts, ensuring accuracy.

Users should easily navigate to their booking history, identifying the specific rental booking they wish to modify or cancel.

Upon selecting a booking, the system should display clear options for either modifying the booking details or proceeding with cancellation.

The interface for modifying booking details, such as dates or car preferences, should be user-friendly and straightforward across various devices.

After a user confirms changes to their booking, the system should process these modifications efficiently, providing confirmation to the user.

Upon user confirmation of a booking cancellation, the system should process the cancellation and update the booking status accordingly.

All user requests to modify or cancel bookings should be handled securely, with user authentication and data encryption as necessary.

Modifications and cancellations should be processed promptly, with minimal wait times for the user, ensuring a smooth experience.

If errors occur during modification or cancellation (e.g., network issues), the system should provide clear instructions or alternatives to the user.



Bug Num
22
23
24
25
26
27

Bug Name
Výběr místa vyzvednutí a vrácení auta
Výběr data a času vyzvednutí a vrácení
Resetování preferencí vozidla
Kalkulace celkové ceny
Text s podmínkami pronájmu
Potvrzovací e-mail

Bug Description

Pole pro výběr místa vyzvednutí a vrácení vozidla někdy nezaznamenává zadané informace, nebo se informace resetují po odchodu uživatele ze stránky

System umožňuje uživatelům vybrat čas vrácení vozidla, který je před časem vyzvednutí, což by nemělo být možné.

Preferované nastavení vozidla, jako je model nebo přídavné vybavení, se resetuje po navigaci uživatele pryč ze stránky a návratu zpět.

Na stránce pro zadání platebních informací může dojít k odhalení citlivých dat v URL adrese, což představuje vážné bezpečnostní riziko.

Text s podmínkami pronájmu se překrývá s ostatními prvky na stránce, což znemožňuje uživatelům jejich čtení a pochopení.

Po dokončení rezervace systém selhává při odesílání potvrzovacího e-mailu, což uživatele zbavuje jistoty, že byla jejich rezervace úspěšně zaznamenána.

Bug Type	Found	Scenario	Closest Scenario
Funkční chyba	Y	10,2	
Funkční chyba	Y	3	
Konfigurační chyba	N		10
Bezpečnostní chyba	A	20	
Vizuální chyba	N		
Funkční chyba	Y	82	

ID chyby
1
2
3
4
5
6
7
8
9

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

Jméno chyby

Poškozené obrázky v karuselu

Sekce s uživatelskými recenzemi

Nefunkční odkazy v navigačním baru

Formulář pro přihlášení k odběru novinek

Po kliknutí na článek tipů na cestování
stránka přesměruje jinam než má

Zastaralá knihovna způsobí pád aplikace

Nesprávné výsledky vyhledávání

Zpomalení při načítání mapy

Nesprávné zobrazení informací o pokojích

Exponovaná platební data

Oprávnění při přidávání recenzí

Problémy s galerií obrázků hotelu

Problém s načítáním recenzí z externího API

Problém s aktualizací dostupnosti pokojů

Autofill nesprávných letišť

Výběr data v minulosti

Filtr přímých letů

Filtr cenového rozmezí

Skryté detaily letů

Tlačítko pro rezervaci letu

Ztráta uložených letů

Výběr místa vyzvednutí a vrácení auta

Výběr data a času vyzvednutí a vrácení

Resetování preferencí vozidla

Kalkulace celkové ceny

Text s podmínkami pronájmu

Potvrzovací e-mail

Popis chyby

Karusel na domovské stránce má integrovány obrázky s vysokým rozlišením prezentující populární destinace. Chyba se projevuje tak, že místo obrázků se uživateli zobrazují ikony poškozeného obrázku.

Sekce s uživatelskými recenzemi obsahuje textové bloky a obrázky, které se nekorektně překrývají kvůli chybám v CSS stylování nebo nedostatkům v responzivním designu.

Navigační bar obsahuje odkazy vedoucí na různé části webu. U některých odkazů dochází k chybě, kdy jsou uživatelé přesměrováni na neexistující stránky, což vyústí v chybu 404.

Uživatel může zadat svůj e-mail do formuláře pro odběr novinek. Formulář akceptuje e-mail, ale systém selže při odesílání potvrzovacího e-mailu uživateli.

Odkazy na články v sekci nejnovějších cestovních zpráv mají za následek zobrazení článku jenž není ten na který bylo kliknuto

Knihovna, jenž má za úkol spravovat rotaci karuselu způsobí pád aplikace po kliknutí na jeden z rotujících obrázků

Při vyhledávání hotelů podle města, data příjezdu a odjezdu a počtu osob někdy systém zobrazuje výsledky, které neodpovídají zadaným kritériím.

Interaktivní mapa zobrazující polohu hotelů je velmi pomalá při načítání a někdy se dokonce nenačte vůbec, což komplikuje proces výběru hotelu na základě jeho umístění.

Po výběru typu pokoje a pokračování k rezervaci se občas stane, že detaily a cena pokoje neodpovídají původně vybraným specifikacím.

Na stránce pro zadání platebních informací může dojít k odhalení citlivých dat v URL adrese, což představuje vážné bezpečnostní riziko.

Na stránce rezervace hotelů mají uživatelé možnost přidávat recenze na hotely, ve kterých se ubytovali. Implementovaná chyba spočívá v tom, že určitá skupina uživatelů narazí na problém s oprávněními, když se pokusí odeslat recenzi. Systém jim oznámí, že nemají dostatečná oprávnění pro přidání recenze, přestože mají být přihlášení a měli by mít tuto možnost jako součást svých uživatelských práv. Tento problém může být způsoben nesprávnou konfigurací serveru nebo chybou v logice aplikace, která ověřuje oprávnění uživatelů.

Galerie obrázků hotelu trpí problémy s načítáním a někdy zobrazuje obrázky v nesprávném poměru stran nebo vůbec nezobrazuje náhledy.

Stránka hotelu zahrnuje sekci s uživatelskými recenzemi, které jsou načítány prostřednictvím externího API. Chyba se projevuje jako neschopnost stránky správně zobrazit recenze kvůli problémům s timeoutem nebo přerušením síťového připojení. Tento problém simuluje reálné situace, kdy externí služby nejsou dostupné nebo je jejich odezva pomalá, což ovlivňuje celkovou funkcionalitu a uživatelskou zkušenost na stránce.

Po provedení rezervace by měl systém automaticky aktualizovat dostupnost pokojů v daném hotelu. V tomto případě chyba v databázi způsobí, že ačkoliv uživatel dokončí proces rezervace a obdrží potvrzení, stav dostupnosti pokojů v databázi se neaktualizuje správně. To může vést k situaci, kdy je stejný pokoj nabídnut více uživatelům, což způsobuje konflikty v rezervacích a možné přebookování.

Vyhledávací formulář poskytuje možnost autofill pro vstupní pole letišť. Chyba se projevuje tím, že návrhy někdy obsahují nesprávná nebo neexistující letiště.

Kalendář umožňuje uživatelům vybrat datum odletu, které je v minulosti, což by mělo být omezeno na aktuální a budoucí data.

Filtr pro vyhledávání přímých letů někdy zobrazí lety s mezipřistáními, což je v rozporu s očekáváním uživatele.

Genový posuvník nesprávně filtruje lety podle ceny, ať už nezobrazuje nejlevnější nabídky nebo nerespektuje nastavený cenový limit.

Rozbalovací sekce s detaily letů má chybu, která někdy způsobí, že se obsah nezobrazí a zůstane skrytý před uživatelem.

Tlačítko „Rezervovat let“ občas nereaguje nebo způsobuje pád stránky, což uživatele znemožňuje dokončit rezervaci.

Funkce pro ukládání vyhledaných letů je nespolehlivá a někdy dojde ke ztrátě informací o uložených letech kvůli problémům se správou relací.

Pole pro výběr místa vyzvednutí a vrácení vozidla někdy nezaznamenává zadané informace, nebo se informace resetují po odchodu uživatele ze stránky

Systém umožňuje uživatelům vybrat čas vrácení vozidla, který je před časem vyzvednutí, což by nemělo být možné.

Preferované nastavení vozidla, jako je model nebo přídavné vybavení, se resetuje po navigaci uživatele pryč ze stránky a návratu zpět.

Jak uživatelé vybírají své auto a doplňky, souhrn rezervace by se měl dynamicky aktualizovat, aby zahrnoval všechny vybrané možnosti a celkovou cenu.

Text s podmínkami pronájmu se překrývá s ostatními prvky na stránce, což znemožňuje uživatelům jejich čtení a pochopení.

Po dokončení rezervace systém selhává při odesílání potvrzovacího e-mailu, což uživatele zbavuje jistoty, že byla jejich rezervace úspěšně zaznamenána.

Typ chyby	Nalezeno	ID testovacího scénáře	Testovaná stránka	ID nejbližší scénář
Vizuální chyba	A	68	Domovská stránka	
Vizuální chyba	A	1.13	Domovská stránka	
Funkční chyba	A	8-13	Domovská stránka	
Funkční chyba	A	88	Domovská stránka	
Funkční chyba	A	101	Domovská stránka	
Konfigurační chyba	A	55	Domovská stránka	
Funkční chyba	A	3	Rezervace Hotelu	
Výkonnostní chyba	A	20	Rezervace Hotelu	
Funkční chyba	A	62	Rezervace Hotelu	

Bezpečnostní chyba	A	68	Rezervace Hotelu	
Problém s oprávněním	A	108	Rezervace Hotelu	
Vizuální chyba	A	35	Rezervace Hotelu	
Síťová chyba	N		Rezervace Hotelu	
Databázová chyba	A	42	Rezervace Hotelu	
Funkční chyba	A	33,34	Rezervace letu	
Funkční chyba	A	56	Rezervace letu	
Funkční chyba	A	64	Rezervace letu	
Funkční chyba	A	67	Rezervace letu	
Vizuální chyba	A	78	Rezervace letu	
Funkční chyba	A	145	Rezervace letu	
Konfigurační chyba	A	118,120	Rezervace letu	
Funkční chyba	A	10,2	Půjčení Auta	
Funkční chyba	A	3	Půjčení Auta	
Konfigurační chyba	N		Půjčení Auta	10
Funkční chyba	A	20	Půjčení Auta	
Vizuální chyba	N		Půjčení Auta	
Funkční chyba	A	82	Půjčení Auta	

Přeložený nejbližší scénář

Obrázky by měly být vysoce kvalitní, aniž by byly zbytečně velké, což by mohlo ovlivnit dobu načítání. Tato rovnováha je klíčová pro udržení estetického působení a výkonu.

Zajistěte, aby se domovská stránka úspěšně načetla se všemi viditelnými prvky: obrázkovým karuselem, navigační lištou, vyhledávacím panelem, uživatelskými recenzemi, tipy na cestování a novinkami, výzvou k odběru.

Výběr možnosti "Speciální nabídky" z navigační lišty by měl uživatele přeměřovat na stránku Speciální nabídky.

Uživatelé by měli krátce po přihlášení k odběru obdržet potvrzovací e-mail, potvrzující jejich přidání do mailingového seznamu.

Kliknutí na článek by mělo vést uživatele k plnému obsahu článku. Navigace k článku by měla fungovat správně.

Kliknutím na obrázek v karuselu by měl být uživatel přenesen na detailní stránku o destinaci nebo nabídce, přičemž URL adresa a obsah stránky by měly odpovídat kontextu kliknutého obrázku.

Formulář by měl ověřovat vstupy pro místa, data (pouze budoucí data), počet cestujících (kladná celá čísla) a třídu služby. Neplatné vstupy by měly vyvolat chybové zprávy.

Interaktivní mapa by se měla rychle načíst a umožnit plynulé přiblížení a posun bez záseků nebo chyb.

Po výběru pokojů a služeb by systém měl zobrazit celkové náklady včetně všech dodatečných služeb a daní.

Všechna uživatelská data a platební informace zadané během rezervace by měly být šifrovány a zabezpečeně zpracovány, aby se chránily před únikem dat.

Uživatelé by měli mít možnost vybrat si minulou rezervaci, odeslat numerické hodnocení a písemnou zpětnou vazbu a úspěšně odeslat tuto recenzi.

Interaktivní prvky, jako jsou fotogalerie, tlačítka pro rezervaci nebo mapy, by měly správně fungovat a přiblížit daný hotel uživateli.

Detailní stránka by měla zobrazovat aktuální informace o dostupnosti pokojů, cenách a jakýchkoli speciálních nabídkách nebo slevách.

Začněte psát známé místo do pole odjezdu nebo příjezdu. Systém by měl v reálném čase zobrazit relevantní návrhy odpovídající vstupu.

Zkuste vybrat data, která jsou mimo rozsah zobrazených cenových rozdílů nebo v minulosti. Systém by měl výběr zabránit nebo proti tomu radit.

Použijte filtr mezipřistání k výběru letů bez mezipřistání nebo s jedním či více mezipřistáními.

Ověřte, že výsledky vyhledávání se aktualizují tak, aby zobrazovaly pouze lety, které splňují kritéria

Použijte cenový filtr pro nastavení maximální nebo konkrétní cenové hladiny. Výsledky vyhledávání by se měly aktualizovat tak, aby zobrazovaly pouze lety, které spadají do stanoveného cenového

Po výběru letu z výsledků vyhledávání se ujistěte, že jsou přesně a komplexně zobrazeny podrobné informace (časy odletu/příletu, doba trvání, mezipřistání, vybavení).

Vyplňte formulář rezervace všemi potřebnými údaji a doplňkovými službami a poté odeslat. Systém by měl vstup ověřit a pokračovat k výpočtu celkových nákladů.

Pokud existuje limit pro počet vyhledávání nebo letů, které si uživatel může uložit, ověřte, že tento limit je vynucen a uživatelé jsou informováni, když ho dosáhnou. Po uložení ověřte, že uložená kritéria vyhledávání nebo konkrétní údaje o letech přesně odrážejí počáteční výběr uživatele a jsou správně uloženy v jejich profilu.

Uživatelé by měli mít možnost uložit si kritéria vyhledávání pronájmu pro budoucí návštěvy, což zlepšuje snadnost použití a efektivitu. Zadejte neplatná data (např. minulá data pro vyzvednutí/vrácení, speciální znaky na místě) pro ověření, že systém správně ověřuje vstupy.

Výběry datumů pro vyzvednutí a vrácení by měly být snadno použitelné, umožňující uživatelům vybrat data bez zmatku nebo chyb.

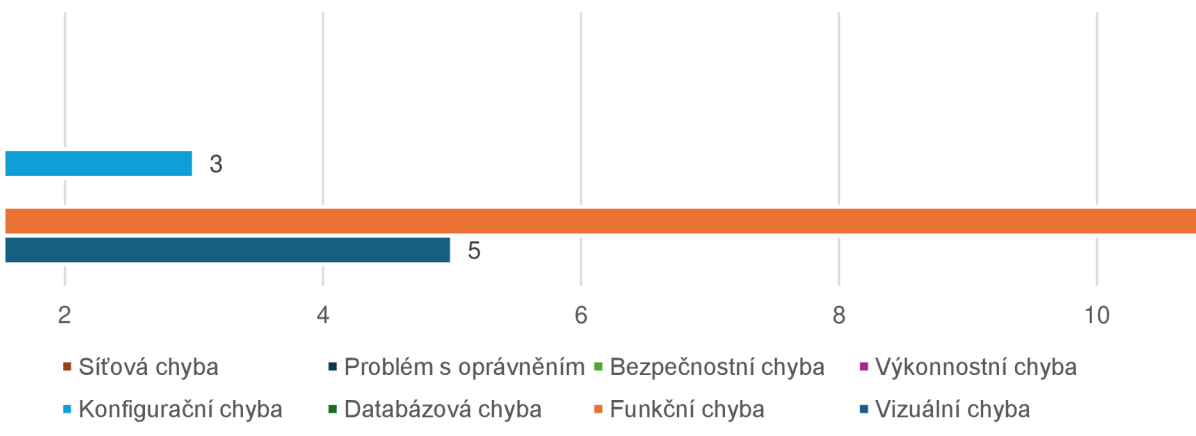
Uživatelé by měli mít možnost uložit si kritéria vyhledávání pronájmu pro budoucí návštěvy, což zlepšuje snadnost použití a efektivitu.

Jak uživatelé vybírají své auto a doplňky, souhrn rezervace by se měl dynamicky aktualizovat, aby zahrnoval všechny vybrané možnosti a celkovou cenu.

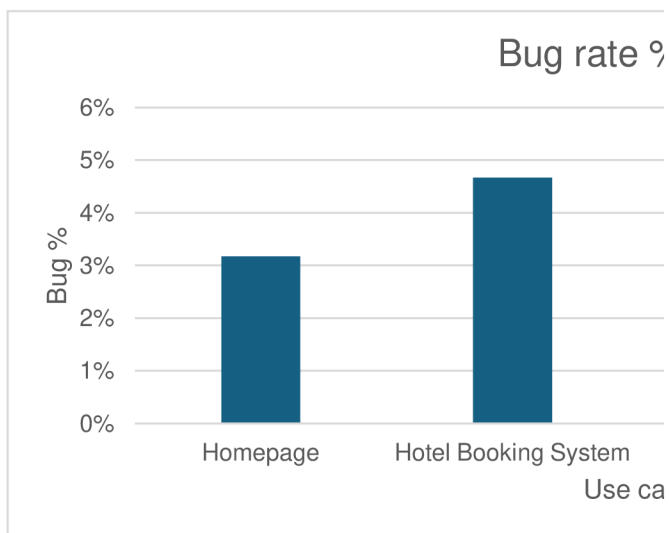
Po úspěšné platbě by uživatelé měli obdržet potvrzení na obrazovce a podrobný e-mail s potvrzením rezervace včetně údajů o pronájmu.

Vizuální chyba	Funkční chyba	Databázová chyba
2	1	0
1	1	1
1	5	0
1	4	0
5	11	1
Vizuální chyba	Funkční chyba	Databázová chyba
2	1	0
1	1	1
1	5	0
0	4	0
4	11	1
80%	100%	100%

Cekové chyby dle typu

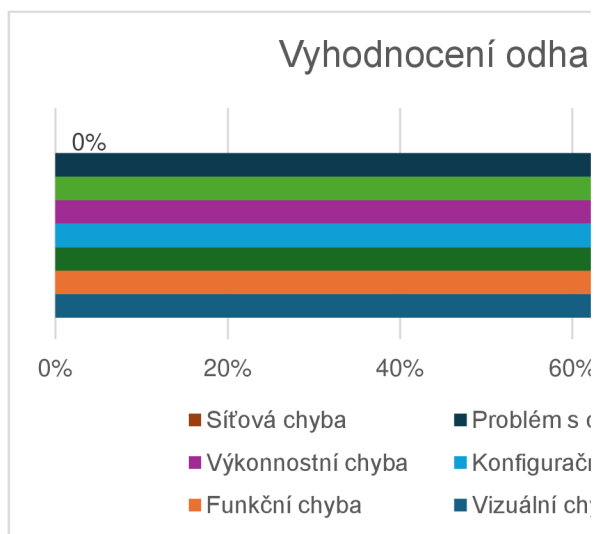
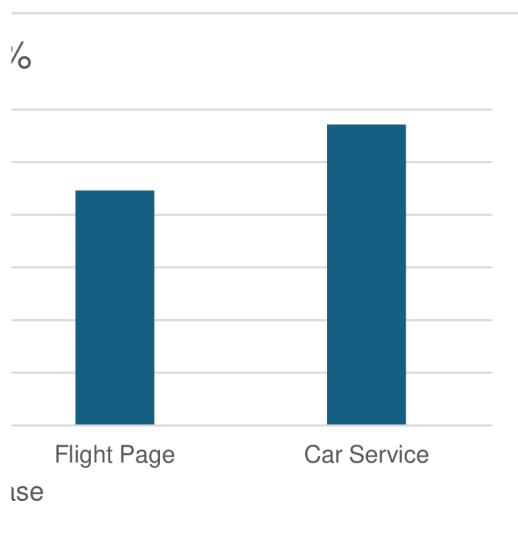


Konfigurační chyba	Výkonnostní chyba	Bezpečnostní chyba
1	0	0
0	1	1
1	0	0
1	0	0
3	1	1
Discovered		
Konfigurační chyba	Výkonnostní chyba	Bezpečnostní chyba
1	0	0
0	1	1
1	0	0
0	0	0
2	1	1
67%	100%	100%



Problém s oprávněním	Sít'ová chyba	Total bugs	Total scenarios
0	0	4	126
1	1	7	150
0	0	7	157
0	0	6	105
1	1	24	538

Problém s oprávněním	Sít'ová chyba	Total bugs	Total discovered
0	0	4	4
1	0	7	6
0	0	7	7
0	0	6	4
1	0	24	21
100%	0%		

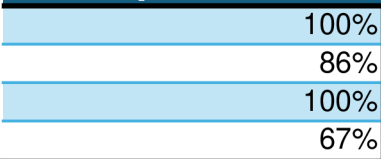


já moc nevim jak to funguje,
ale je to prostě procentuální
poměr scénárií vůči bugům

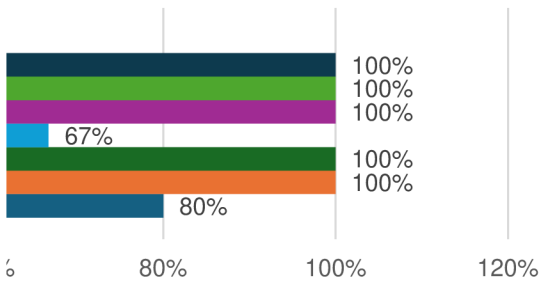
Bug rate



Discovery rate



lení chyb dle typu



oprávněním ■ Bezpečnostní chyba
ní chyba ■ Databázová chyba
yba