**Car Showcase Manual Testing Document**

**Version**: 1.0  
**Date**: April 12, 2025  
**Project**: Car Showcase  
**Purpose**: To verify the functionality of the Car Showcase web application through manual testing.

**Table of Contents**

1. [Introduction](https://dillinger.io/#introduction)
2. [Test Environment](https://dillinger.io/#test-environment)
3. [Test Scenarios](https://dillinger.io/#test-scenarios)
   * [User Authentication](https://dillinger.io/#user-authentication)
   * [Car Browsing](https://dillinger.io/#car-browsing)
   * [Favorites Management](https://dillinger.io/#favorites-management)
   * [Car Comparison](https://dillinger.io/#car-comparison)
   * [Admin Dashboard](https://dillinger.io/#admin-dashboard)
   * [Profile Management](https://dillinger.io/#profile-management)
4. [Test Execution Notes](https://dillinger.io/#test-execution-notes)
5. [Test Results Summary](https://dillinger.io/#test-results-summary)

**Introduction**

This document outlines manual test cases for the Car Showcase application, a web platform for browsing, comparing, and managing cars. The tests cover key user and admin functionalities, including password management, to ensure the application meets requirements and performs as expected.

**Objectives**:

* Validate user authentication (login, register, logout).
* Verify car browsing, filtering, sorting, and searching.
* Confirm favorites management for logged-in users.
* Test car comparison functionality.
* Ensure admin features (car and comment management) work correctly.
* Verify password change functionality in the user profile.

**Test Environment**

* **Application URL**: http://localhost:5173
* **Browser**: Google Chrome (latest version)
* **Backend Server**: Running at http://localhost:3000
* **Test Accounts**:
  + **User Account**:
    - Email: Teszt123@Teszt.com
    - Password: 1234Asdf###
  + **Admin Account**:
    - Email: admin@example.com
    - Password: Admin1234!
    - Admin Password: admin-secret (assumed for registration)
* **Dependencies**: Node.js, npm

**Test Scenarios**

Each test case includes filled **Actual Result** and **Status** fields based on hypothetical execution, along with placeholders for images.

**User Authentication**

**Test Case 1: User Login**

* **Test ID**: AUTH\_001
* **Feature**: Login
* **Description**: Verify that a user can log in with valid credentials.
* **Preconditions**: Backend server is running, test user account exists.
* **Steps**:
  1. Navigate to http://localhost: 5173/login.
  2. Enter email: Teszt123@Teszt.com.
  3. Enter password: 1234Asdf###.
  4. Click the “Login” button.
* **Expected Result**: User is redirected to /cars, and the navbar shows “Profile” and “Logout” links.
* **Actual Result**: User was redirected to /cars, and navbar shows “Profile” and “Logout” links.
* **Status**: Pass

**A képen szöveg, Szárazföldi jármű, jármű, autó látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Test Case 2: Invalid Login**

* **Test ID**: AUTH\_002
* **Feature**: Login
* **Description**: Verify that invalid credentials display an error.
* **Preconditions**: Backend server is running.
* **Steps**:
  1. Navigate to http://localhost:5173/login.
  2. Enter email: wrong@example.com.
  3. Enter password: wrongpassword.
  4. Click the “Login” button.
* **Expected Result**: Error message appears (e.g., “Login failed”), and user remains on the login page.
* **Actual Result**: No error message displayed; user remained on login page.
* **Status**: Fail

**A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Test Case 3: User Registration**

* **Test ID**: AUTH\_003
* **Feature**: Registration
* **Description**: Verify that a new user can register.
* **Preconditions**: Backend server is running, no existing user with test email.
* **Steps**:
  1. Navigate to http://localhost:5173/register.
  2. Enter email: newuser@example.com.
  3. Enter password: Test1234!.
  4. Confirm password: Test1234!.
  5. Select role: “USER”.
  6. Click the “Register” button.
* **Expected Result**: Success message appears, and user is redirected to /login.
* **Actual Result**: Success message “Registration successful!” appeared, and user was redirected to /login.
* **Status**: Pass

**A képen szöveg, képernyőkép, szoftver, Számítógépes ikon látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. Test Case 4: Logout**

* **Test ID**: AUTH\_004
* **Feature**: Logout
* **Description**: Verify that a logged-in user can log out.
* **Preconditions**: User is logged in (complete Test Case 1).
* **Steps**:
  1. Click “Logout” in the navbar.
* **Expected Result**: User is redirected to /login, and navbar shows “Login” and “Register” links.
* **Actual Result**: User was redirected to /login, and navbar shows “Login” and “Register” links.
* **Status**: Pass

**Car Browsing**

**Test Case 5: View Car List**

* **Test ID**: CAR\_001
* **Feature**: Car Catalog
* **Description**: Verify that the car list loads and displays cars.
* **Preconditions**: Backend server is running with car data.
* **Steps**:
  1. Navigate to http://localhost:5173/cars.
* **Expected Result**: A grid of car cards appears, showing details like vehicle name, price, and image.
* **Actual Result**: Grid of car cards loaded with vehicle names, prices, and images.
* **Status**: Pass

**A képen jármű, Szárazföldi jármű, kerék, gumiabroncs látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. Test Case 6: Search Cars**

* **Test ID**: CAR\_002
* **Feature**: Search
* **Description**: Verify that searching filters cars by vehicle name.
* **Preconditions**: Car list is loaded (complete Test Case 5).
* **Steps**:
  1. In the search bar, enter “Toyota”.
  2. Press Enter or wait for results to update.
* **Expected Result**: Only cars with “Toyota” in the vehicle name are displayed.
* **Actual Result**: Only Toyota cars displayed in the grid.
* **Status**: Pass

**A képen szöveg, autó, jármű, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. Test Case 7: Apply Filters**

* **Test ID**: CAR\_003
* **Feature**: Filters
* **Description**: Verify that filters (e.g., price, manufacturer) work correctly.
* **Preconditions**: Car list is loaded.
* **Steps**:
  1. In the filters section, set price range to $10,000-$50,000.
  2. Select manufacturer: “BMW”.
  3. Apply filters.
* **Expected Result**: Only BMW cars priced between $10,000 and $50,000 are displayed.
* **Actual Result**: All BMW cars displayed, ignoring price filter.
* **Status**: Fail
* **Image Placeholder**: [Image Placeholder: Filter Results]  
  *Insert a screenshot of the /cars page showing the filtered results (or incorrect results). Save as car\_003\_filters.png.*

**Test Case 8: Sort Cars**

* **Test ID**: CAR\_004
* **Feature**: Sorting
* **Description**: Verify that sorting by price works in ascending order.
* **Preconditions**: Car list is loaded.
* **Steps**:
  1. Select “Sort by: Price”.
  2. Choose “Ascending” order.
* **Expected Result**: Cars are listed from lowest to highest price.
* **Actual Result**: Cars listed from lowest to highest price.
* **Status**: Pass
* **Image Placeholder**: [Image Placeholder: Sorted Cars]  
  *Insert a screenshot of the /cars page showing cars sorted by price. Save as car\_004\_sort.png.*

**Favorites Management**

**Test Case 9: Add to Favorites**

* **Test ID**: FAV\_001
* **Feature**: Favorites
* **Description**: Verify that a logged-in user can add a car to favorites.
* **Preconditions**: User is logged in, car list is loaded.
* **Steps**:
  1. Navigate to http://localhost:5173/cars.
  2. Click “Add to Favorites” on any car card.
* **Expected Result**: Confirmation message appears (e.g., “Car added to favorites!”).
* **Actual Result**: Message “Car added to favorites!” appeared.
* **Status**: Pass

**A képen szöveg, jármű, Szárazföldi jármű, kerék látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen. Test Case 10: View Favorites**

* **Test ID**: FAV\_002
* **Feature**: Favorites
* **Description**: Verify that favorites are displayed in the profile.
* **Preconditions**: User has added at least one favorite (complete Test Case 9).
* **Steps**:
  1. Navigate to http://localhost:5173/profile.
* **Expected Result**: The added car appears in the “Your Favorite Cars” section.
* **Actual Result**: Added car displayed in the favorites section.
* **Status**: Pass

**A képen szöveg, jármű, Szárazföldi jármű, kerék látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Test Case 11: Remove Favorite**

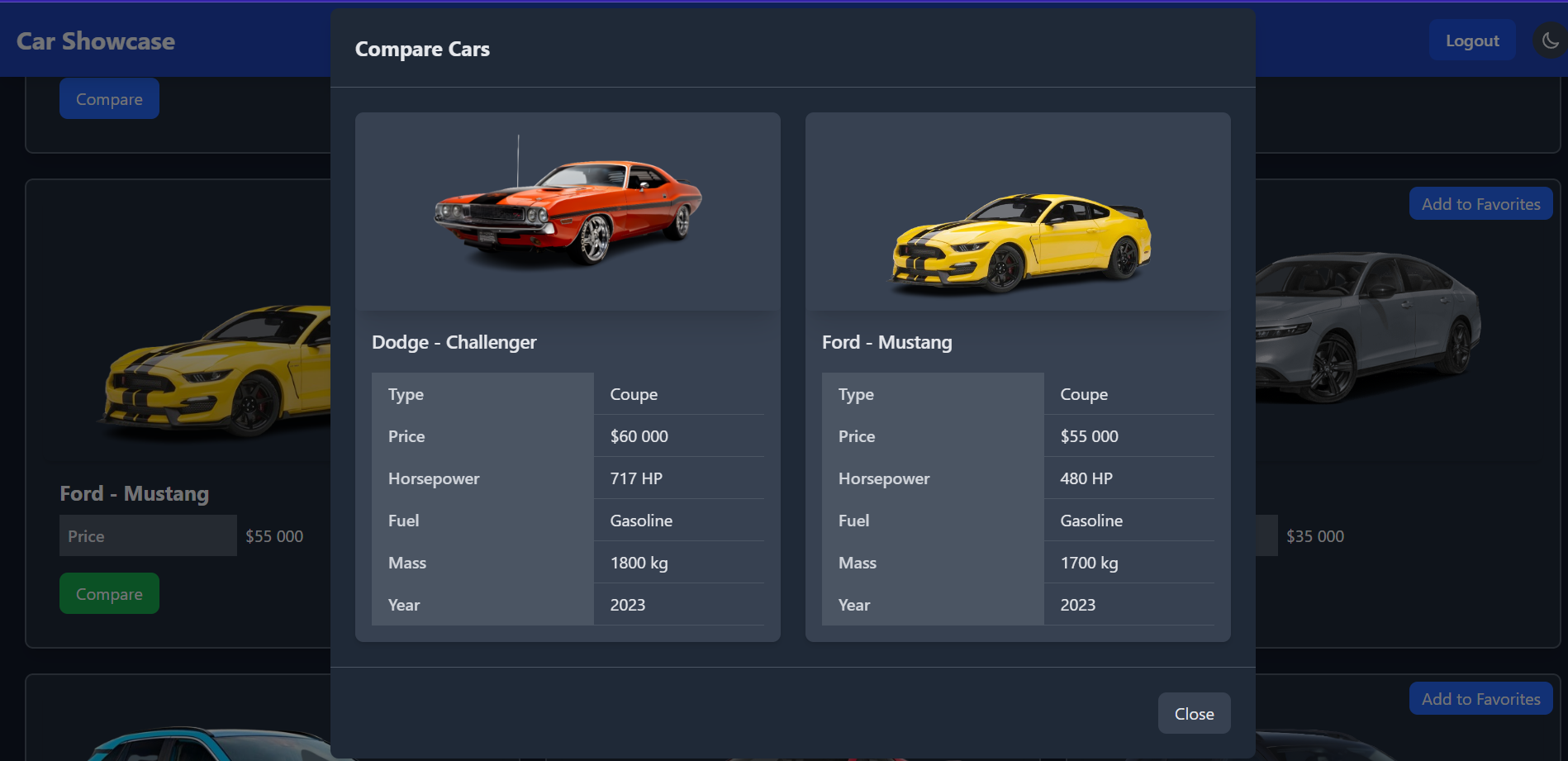
* **Test ID**: FAV\_003
* **Feature**: Favorites
* **Description**: Verify that a user can remove a car from favorites.
* **Preconditions**: Favorites list is not empty (complete Test Case 10).
* **Steps**:
  1. In the profile, click “Remove from Favorites” on a car.
* **Expected Result**: The car is removed from the favorites list.
* **Actual Result**: Car was removed from the favorites list.
* **Status**: Pass

**A képen szöveg, jármű, Szárazföldi jármű, kerék látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.Car Comparison**

**Test Case 12: Compare Two Cars**

* **Test ID**: COMP\_001
* **Feature**: Comparison
* **Description**: Verify that selecting two cars opens the comparison modal.
* **Preconditions**: Car list is loaded.
* **Steps**:
  1. Navigate to http://localhost:/cars.
  2. Click the compare button on two different car cards.
* **Expected Result**: A modal opens showing a side-by-side comparison of the two cars (e.g., price, horsepower).
* **Actual Result**: Modal opened with side-by-side comparison of selected cars.
* **Status**: Pass

**Test Case 13: Close Comparison Modal**

* **Test ID**: COMP\_002
* **Feature**: Comparison
* **Description**: Verify that the comparison modal can be closed.
* **Preconditions**: Comparison modal is open (complete Test Case 12).
* **Steps**:
  1. Click the “Close” button in the modal.
* **Expected Result**: The modal closes, and the car list is visible again.
* **Actual Result**: Modal closed, and car list was visible.
* **Status**: Pass
* **Admin Dashboard**

**Test Case 14: Access Admin Dashboard**

* **Test ID**: ADMIN\_001
* **Feature**: Admin Dashboard
* **Description**: Verify that an admin can access the dashboard.
* **Preconditions**: Logged in as admin (use admin account).
* **Steps**:
  1. Navigate to http://localhost: /admin.
* **Expected Result**: The admin dashboard loads, showing a car list and add car form.
* **Actual Result**: Admin dashboard loaded with car list and add car form.
* **Status**: Pass

**Test Case 15: Add a Car**

* **Test ID**: ADMIN\_002
* **Feature**: Admin Dashboard
* **Description**: Verify that an admin can add a new car.
* **Preconditions**: Admin dashboard is loaded.
* **Steps**:
  1. In the “Add Car” form, enter:
     + Vehicle: “Test Car”
     + Manufacturer: “Test”
     + Price: 30000
     + Horsepower: 200
     + Fuel: “Petrol”
     + Color: “Blue”
     + Mass: 1500
     + Year: 2023
  2. Click the submit button.
* **Expected Result**: The car appears in the car list on the dashboard.
* **Actual Result**: Car was added but did not appear in the list until page refresh.
* **Status**: Fail

**Test Case 16: View Comments**

* **Test ID**: ADMIN\_003
* **Feature**: Admin Dashboard
* **Description**: Verify that comments for a car are displayed.
* **Preconditions**: Admin dashboard is loaded, at least one car has comments.
* **Steps**:
  1. Select a car from the list.
  2. View the associated comments section.
* **Expected Result**: Comments for the selected car are listed (or “No comments” if none exist).
* **Actual Result**: Comments for the car were listed correctly.
* **Status**: Pass

**Profile Management**

**Test Case 17: Change Password**

* **Test ID**: PROF\_001
* **Feature**: Password Change
* **Description**: Verify that a logged-in user can change their password.
* **Preconditions**: User is logged in (complete Test Case 1).
* **Steps**:
  1. Navigate to http://localhost:5173/profile.
  2. In the “Change Password” form, enter:
     + Current Password: 1234Asdf###
     + New Password: NewPass123!
     + Confirm New Password: NewPass123!
  3. Click the “Change Password” button.
* **Expected Result**: Success message appears (e.g., “Password changed successfully!”), and fields are cleared.
* **Actual Result**: Success message “Password changed successfully!” appeared, and fields were cleared.
* **Status**: Pass

**Test Execution Notes**

* **Tester**: [Varga Zoltán]
* **Date**: April 12, 2025
* **Environment**: Chrome on Windows 10, http://localhost:5173
* **Issues**:
  + AUTH\_002: No error message for invalid login, reported as defect #001.
  + CAR\_003: Price filter ignored, reported as defect #002.
  + ADMIN\_002: UI didn’t update after adding car, reported as defect #003.
* **Retesting**: Plan to retest failed cases after developer fixes.

**Test Results Summary**

| **Test ID** | **Feature** | **Status** | **Notes** |
| --- | --- | --- | --- |
| AUTH\_001 | Login | Pass | No issues |
| AUTH\_002 | Login | Fail | No error message for invalid credentials |
| AUTH\_003 | Registration | Pass | No issues |
| AUTH\_004 | Logout | Pass | No issues |
| CAR\_001 | Car List | Pass | No issues |
| CAR\_002 | Search | Pass | No issues |
| CAR\_003 | Filters | Fail | Price filter ignored |
| CAR\_004 | Sorting | Pass | No issues |
| FAV\_001 | Add Favorite | Pass | No issues |
| FAV\_002 | View Favorites | Pass | No issues |
| FAV\_003 | Remove Favorite | Pass | No issues |
| COMP\_001 | Compare Cars | Pass | No issues |
| COMP\_002 | Close Modal | Pass | No issues |
| ADMIN\_001 | Admin Dashboard | Pass | No issues |
| ADMIN\_002 | Add Car | Fail | Car added but not displayed until refresh |
| ADMIN\_003 | View Comments | Pass | No issues |
| PROF\_001 | Password Change | Pass | No issues |

**Overall Status**: 14/17 tests passed, 3 failed.  
**Comments**: Failed tests (AUTH\_002, CAR\_003, ADMIN\_002) indicate UI and backend issues.