**Assignment 4 – JavaScript Interactivity**

**Students:** Uzbekbayev Bekarys,Agabekuly Asylbek,Syrgabaev Aset  
**Group:** SE-2438  
**Project:** *AutoVerdict – Car Reviews and Comparisons*  
**Course:** Web Technologies 1 (Front-End)  
**Date:** October 2025

**Objective**

The goal of this assignment was to extend the existing AutoVerdict website with interactive JavaScript components that demonstrate key JS fundamentals: variables, functions, loops, arrays, validation, DOM manipulation, random number generation, and control statements.

Each page of the project now includes at least one JavaScript-based feature that improves user interactivity and usability.

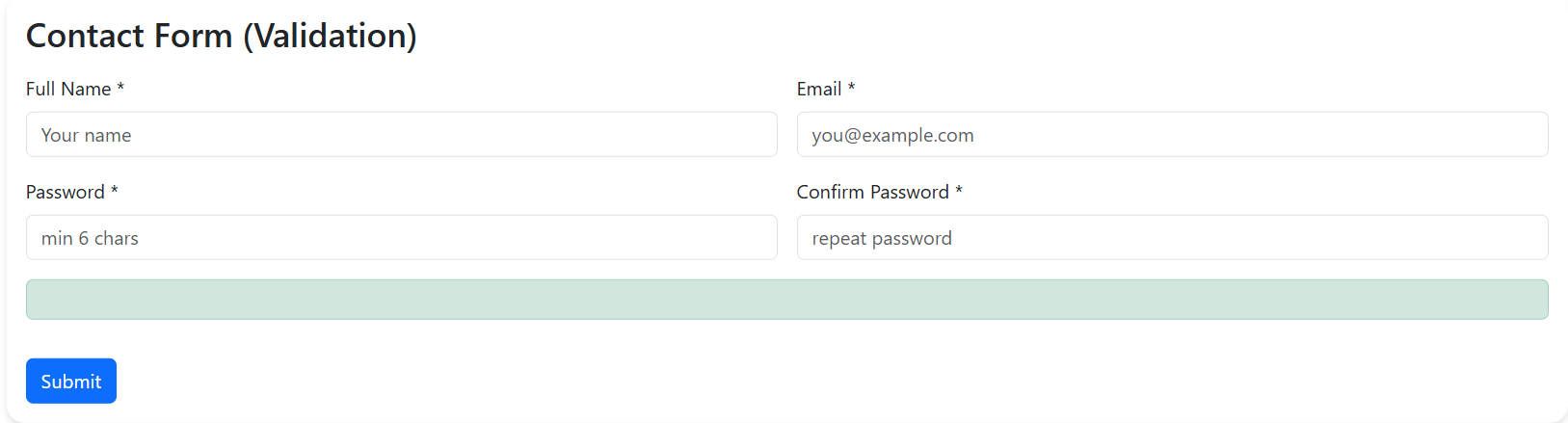
**1. Form Validation – index.html**

**Purpose:**  
To validate user input in a contact form before submission.

**Implementation:**

* Checks that *Name* and *Email* are filled in.
* Verifies email format with a regular expression.
* Ensures the password length is ≥ 6 characters.
* Confirms that password and confirmation match.
* Displays red alert with error list or green success message.

**Result:**  
When the user clicks **Submit**, JavaScript prevents empty or invalid data and shows validation messages dynamically.



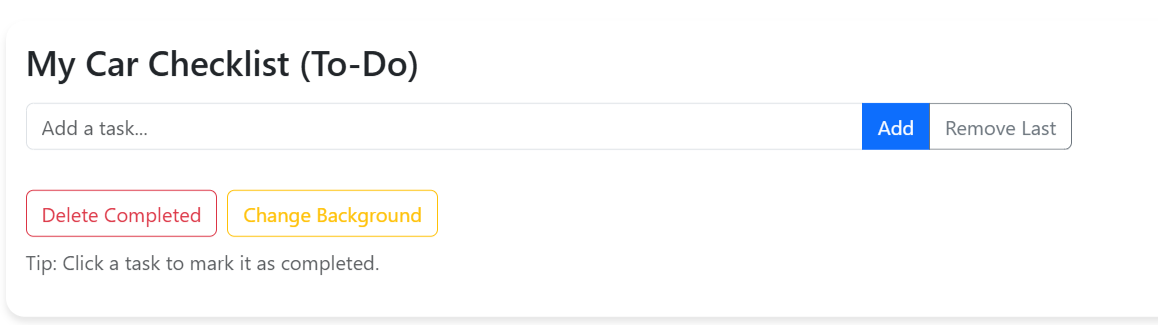
**2. Interactive To-Do List + Background Color – guides.html**

**Purpose:**  
To allow the user to manage personal tasks and dynamically change the background color of the page.

**Implementation:**

* Adds tasks to a list using an input field and “Add” button.
* Tasks can be clicked to mark them as completed (strike-through).
* Buttons: **Delete Completed**, **Remove Last**, and **Change Background**.
* “Change Background” picks a random color each click.

**Result:**  
The user can interactively manage a checklist and visually personalize the page.



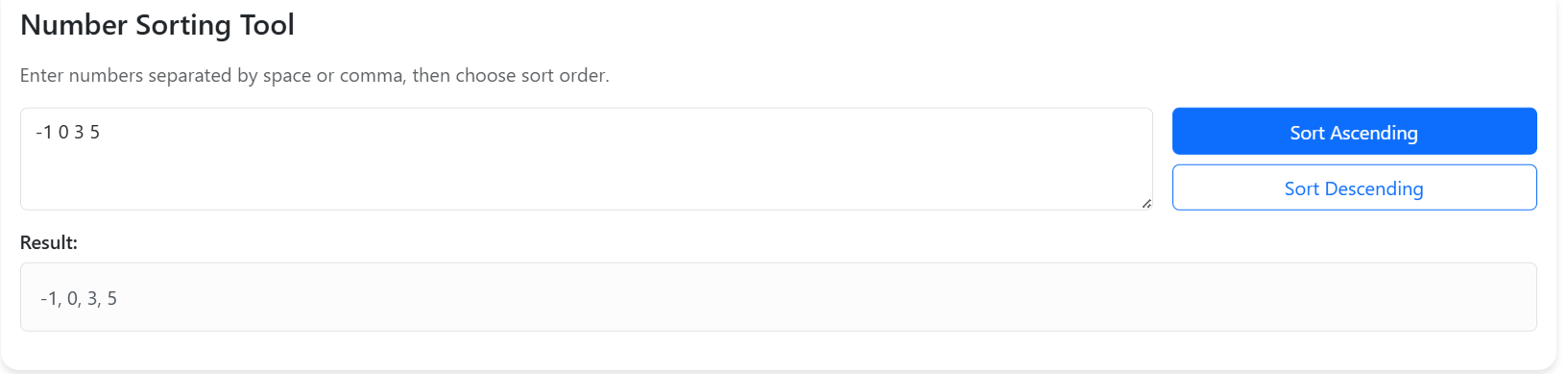
**3. Number Sorting Tool – comparisons.html**

**Purpose:**  
To sort user-entered numbers in ascending or descending order.

**Implementation:**

* User inputs numbers separated by commas or spaces.
* JavaScript parses input into an array and validates that all values are numeric.
* Uses Array.sort() for ascending/descending order.
* Displays the sorted numbers dynamically on the page.

**Result:**  
Provides instant visual sorting results with error handling for invalid input.



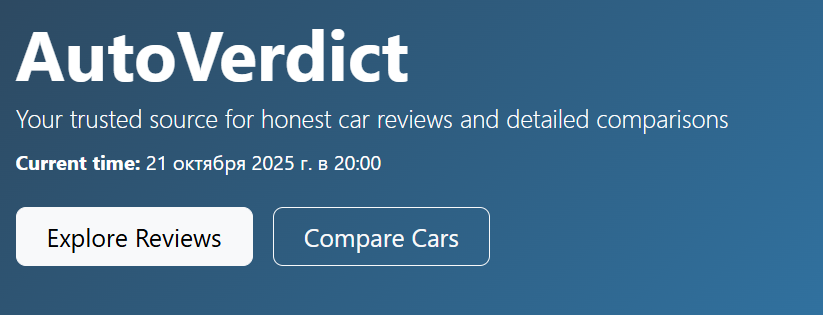
**4. Display Current Date and Time – index.html**

**Purpose:**  
To display the current date and time using JavaScript’s Date object.

**Implementation:**

* A <span id="dtNow"> is updated every second using setInterval().
* Uses toLocaleString() for formatted date/time (e.g., “October 21, 2025, 19:20 PM”).

**Result:**  
The time on the homepage updates in real-time, showing dynamic content on page load.



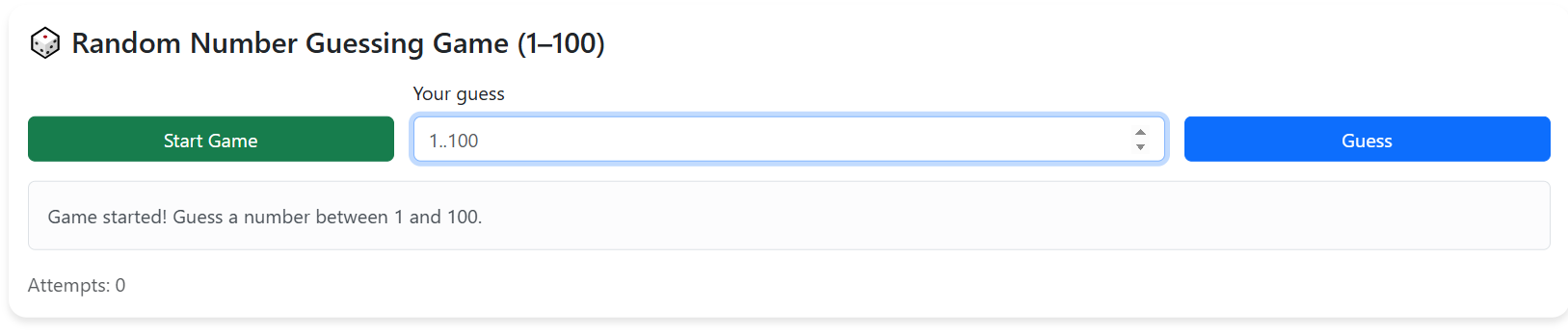
**5. Random Number Guessing Game – reviews.html**

**Purpose:**  
To create a simple game demonstrating random number generation and control flow.

**Implementation:**

* On “Start Game”, a random number (1 – 100) is generated using Math.random() and Math.floor().
* User enters guesses; program responds “Too high” / “Too low” / “Correct!”.
* Tracks number of attempts.
* Disables input when the correct number is guessed.

**Result:**  
An engaging interactive game that demonstrates loops, conditionals, and user feedback logic.



**6. Team Members**

Uzbekbayev Bekarys • Syrgabayev Aset • Agabekuly Asylbek

**7. Technical Summary**

| **Component** | **JS Concepts Used** | **Files** |
| --- | --- | --- |
| Form Validation | Variables, regex, DOM, conditionals | index.html |
| To-Do List | Arrays, loops, event listeners, DOM | guides.html |
| Sorting Tool | Arrays, parsing, comparison functions | comparisons.html |
| Background Change | Random selection, DOM style manipulation | guides.html |
| Date & Time | Date object, timers | index.html |
| Guessing Game | Random numbers, if/else, loops, state | reviews.html |
| Shared JS File | Modular structure, event initialization | script.js |

**8. Conclusion**

All required features from Assignment 4 have been successfully implemented and tested on multiple pages of the AutoVerdict project.  
The use of JavaScript enhances interactivity, provides input validation, and adds engaging user experiences through real-time updates, randomization, and DOM-based control.