🚗Car Manager “AutoElite”

Imagine you are working as a junior web developer in a small company called AutoElite, dedicated to buying and selling used cars.

The manager has asked you to develop a small internal web application to help the team manage the cars arriving at the dealership.

🧩 Objective

The manager wants a simple, modern, and visual tool where employees can register new cars, check those already added, and have a small control over prices and years.

To do this, you must create a web page that meets the following conditions:

🏁 Part 1: Car Registration

AutoElite needs a form where employees can enter the data of each car that arrives at the workshop.

For each vehicle, the following must be indicated:

- Brand
- Model
- Year
- Price

When the “**Add Car**” button is clicked, the system must check the following before saving:

- All form fields must be filled in.
- The car's year must not be greater than the current year, as that would be incorrect.

When saving the information, we must **internally record the exact date and time when the car was registered.** This date will not be displayed on the screen but will be stored in the system for future reference.

🚘 Part 2: Dealership List

Each registered car must appear in a visual list on the same page.

For each car, the following will be displayed:

- Its brand, model, year, and price
- Alongside them, the brand's image

The images will be loaded automatically from the provided images folder.

For example:

- If the car's brand is Toyota, the image `images/toyota.jpg` will be displayed.
- If the image for that brand is not found, a generic image called `none.jpg` will be shown.

In addition, at the top of the list, the total number of registered cars must be displayed.

📊 Part 3: Manager Tools

The manager wants to consult general stock information.

For that, the page must include two additional buttons:

- **"Show summary"** → When clicked, the application must open a new window showing:
  - The total number of registered cars
  - The average price of all cars (rounded to the nearest whole number)
  - A button to close the window

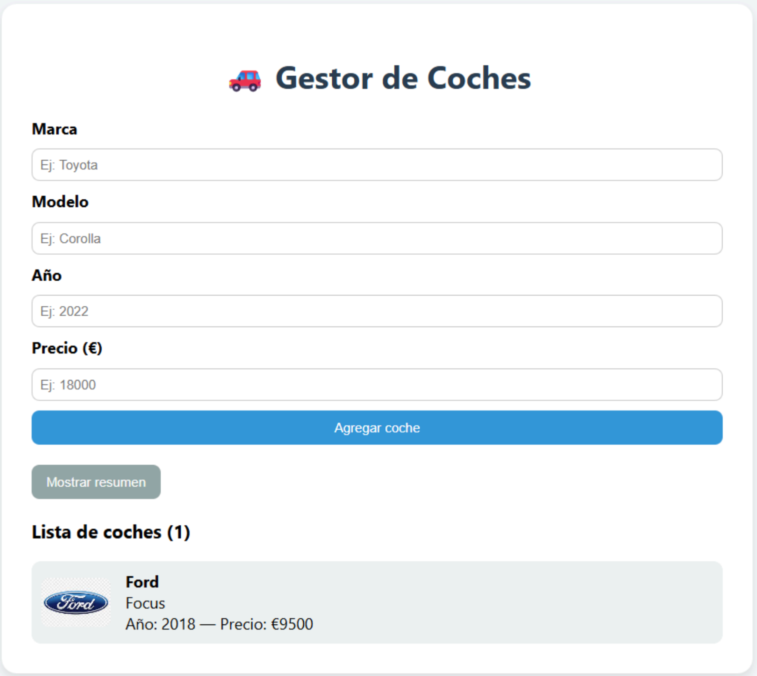
Cars must be stored in an **array of objects**, and each object must represent a car with all its data and registration date.

🧰 What you must deliver

The `script.js` file, which must work without modifying any other file.

🪄 Example of Use

Imagine a **Ford Focus** arrives from 2018, with a price of **€9,500**. The employee enters that data into the form and clicks **“Add Car.”** The car automatically appears in the list, along with the **Ford logo**.

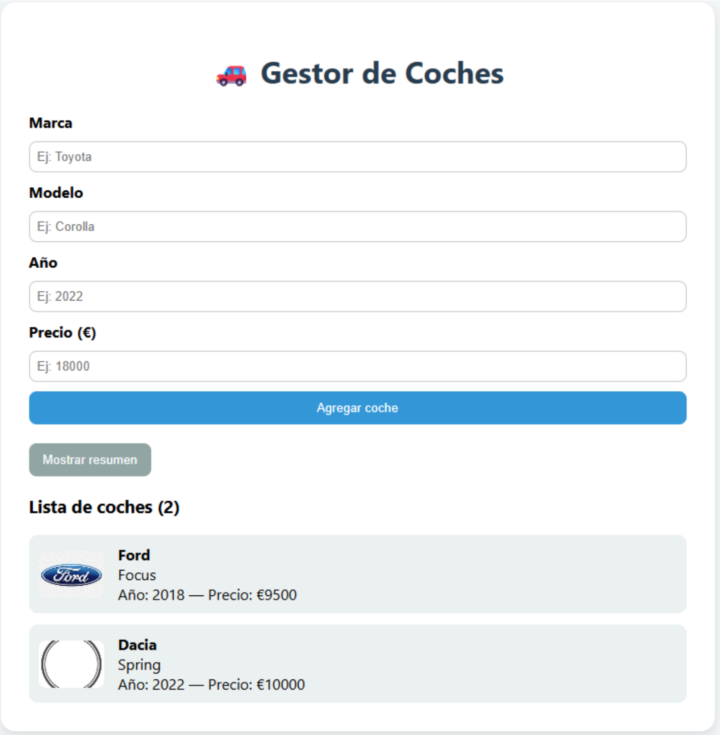


🚗 Gestor de Coches

Lista de coches (1)

Ford
Focus
Año: 2018 — Precio: €9500

If later a car is added from a brand without an image (for example, “Dacia”), the application will show the generic `none.jpg` image.



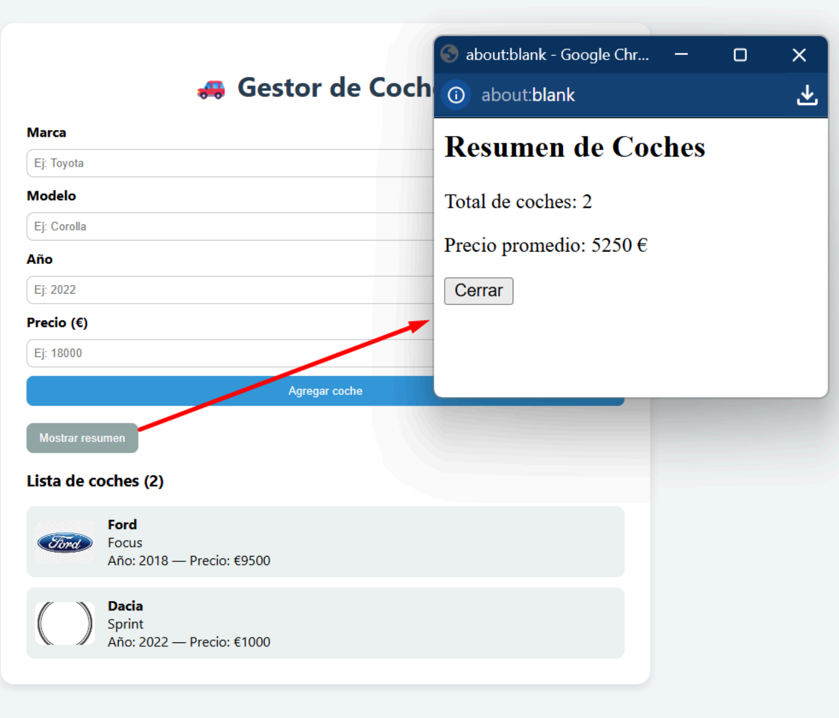
🚗 Gestor de Coches

Lista de coches (2)

Ford
Focus
Año: 2018 — Precio: €9500

Dacia
Spring
Año: 2022 — Precio: €10000

Later, the manager clicks **“Show summary”**, and the app displays how many cars there are and what the average price is.



Lista de coches (2)

Ford
Focus
Año: 2018 — Precio: €9500

Dacia
Sprint
Año: 2022 — Precio: €1000

Resumen de Coches

Total de coches: 2

Precio promedio: 5250 €