Технически университет София – Филиал Пловдив

Курсов проект по Java

**Изготвил:** Даниел Гочев Михов, 41а

**Фак. номер:** 374960

**Специалност:** КСТ

# Car Repair Service

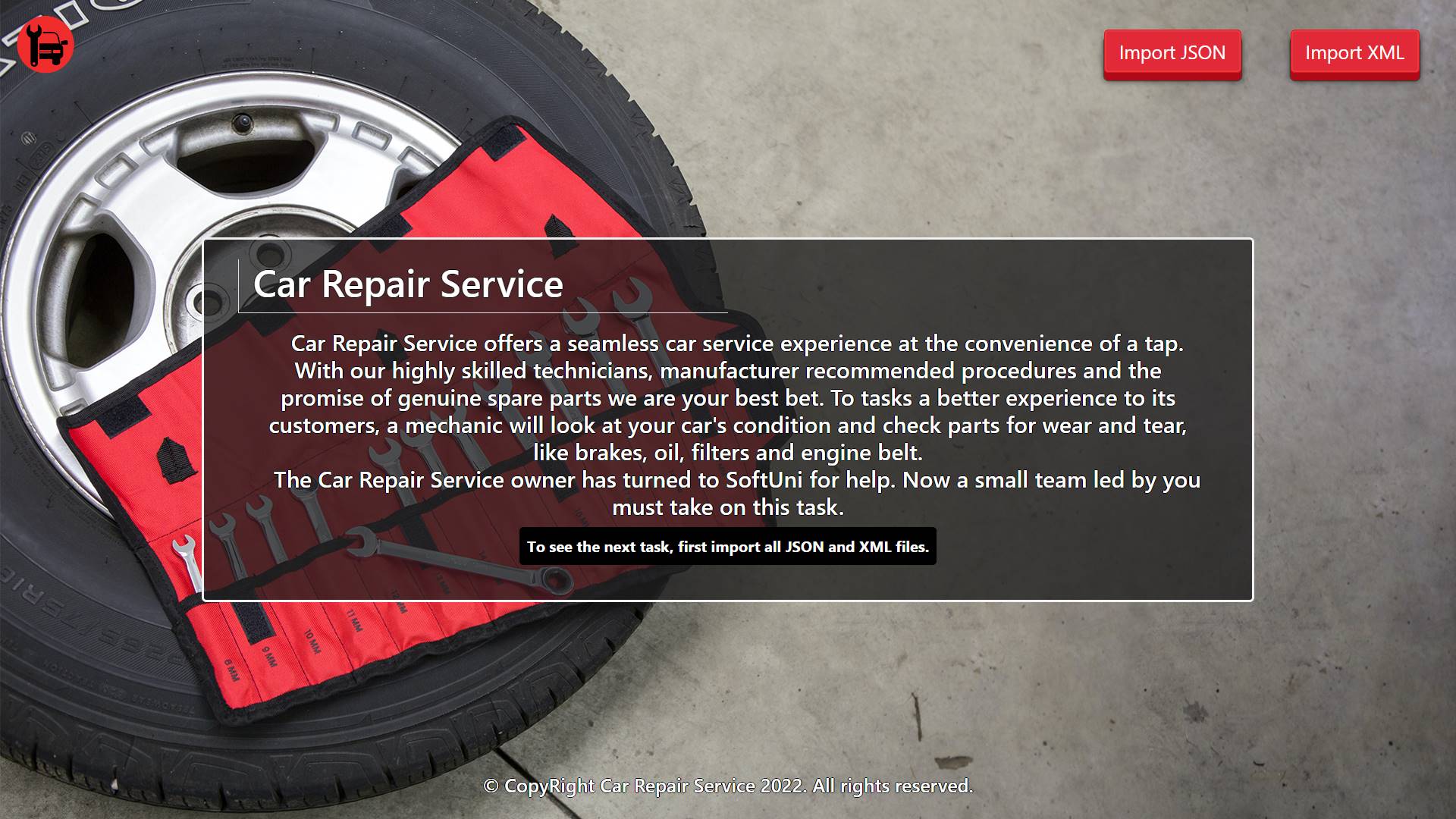
Car Repair Service offers a seamless car service experience at the convenience of a tap. With our highly skilled technicians, manufacturer recommended procedures and the promise of genuine spare parts we are your best bet. To tasks a better experience to its customers, a mechanic will look at your car's condition and check parts for wear and tear, like brakes, oil, filters and engine belt.

## Functionality Overview

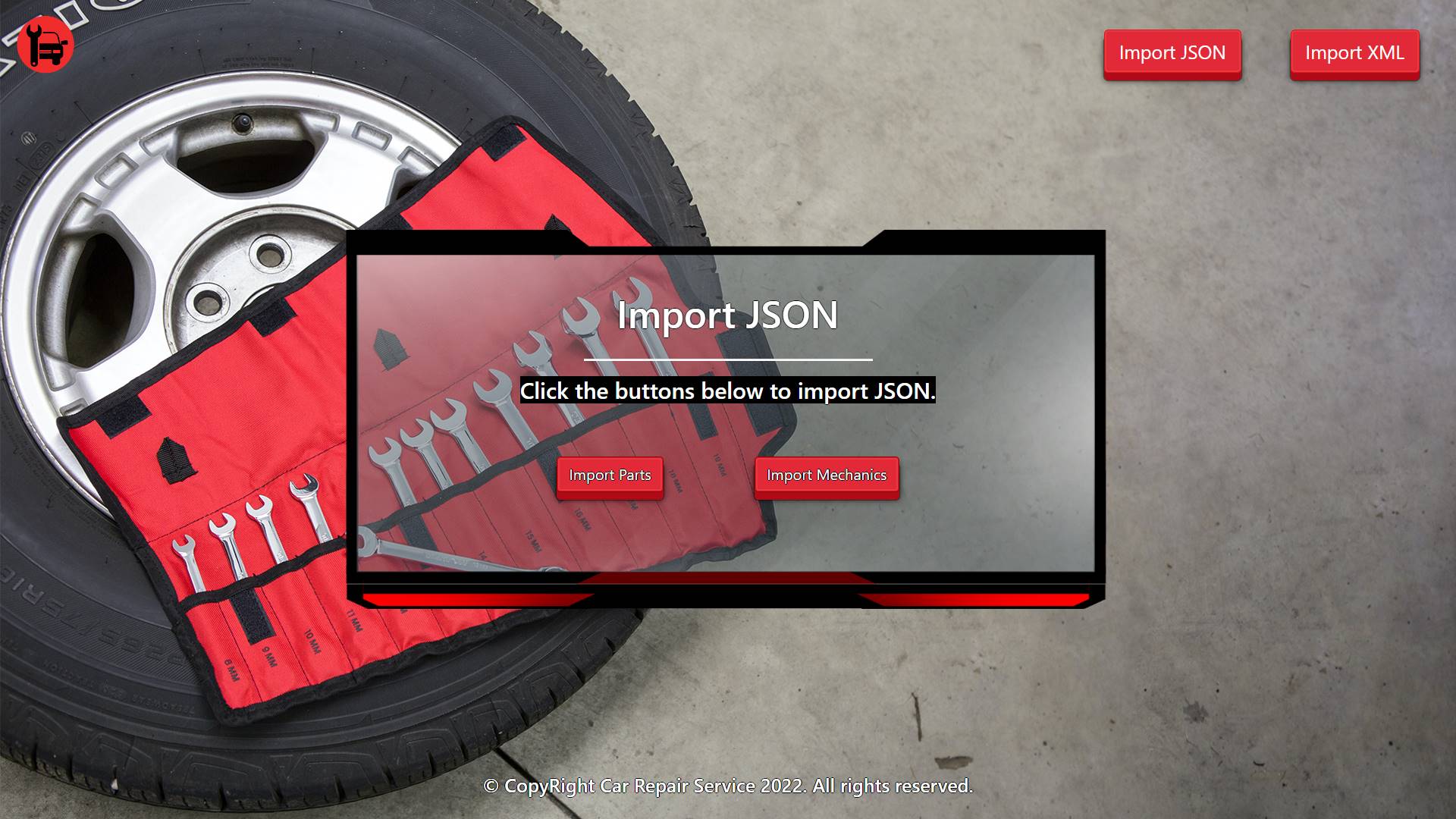
The application should be able to easily **import** hard-formatted data and **support functionalities** for also **exporting** the imported data. The application is called – **Car Repair Service**.

Look at the pictures below to see what must happen:

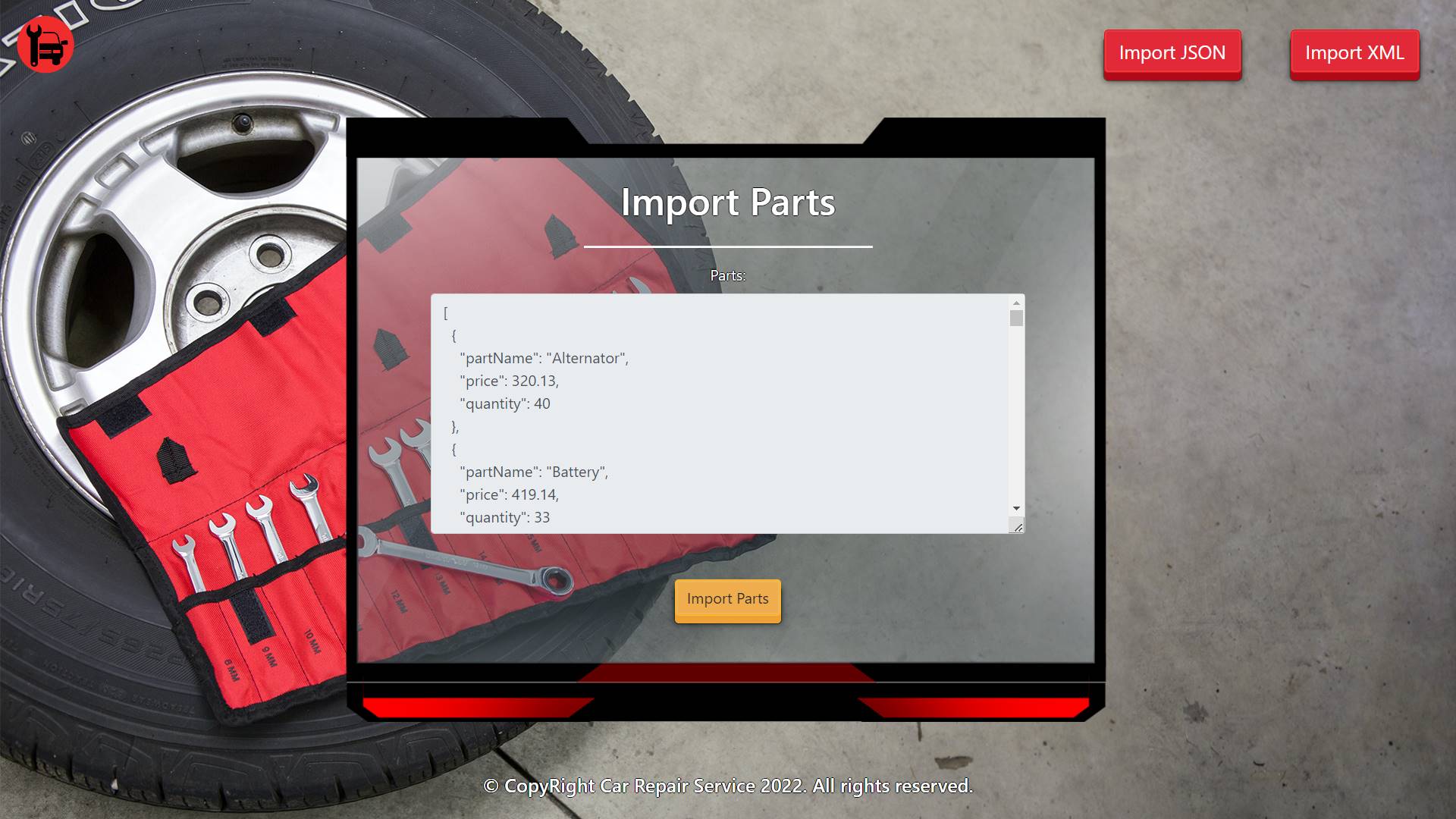
* The home page before importing anything:



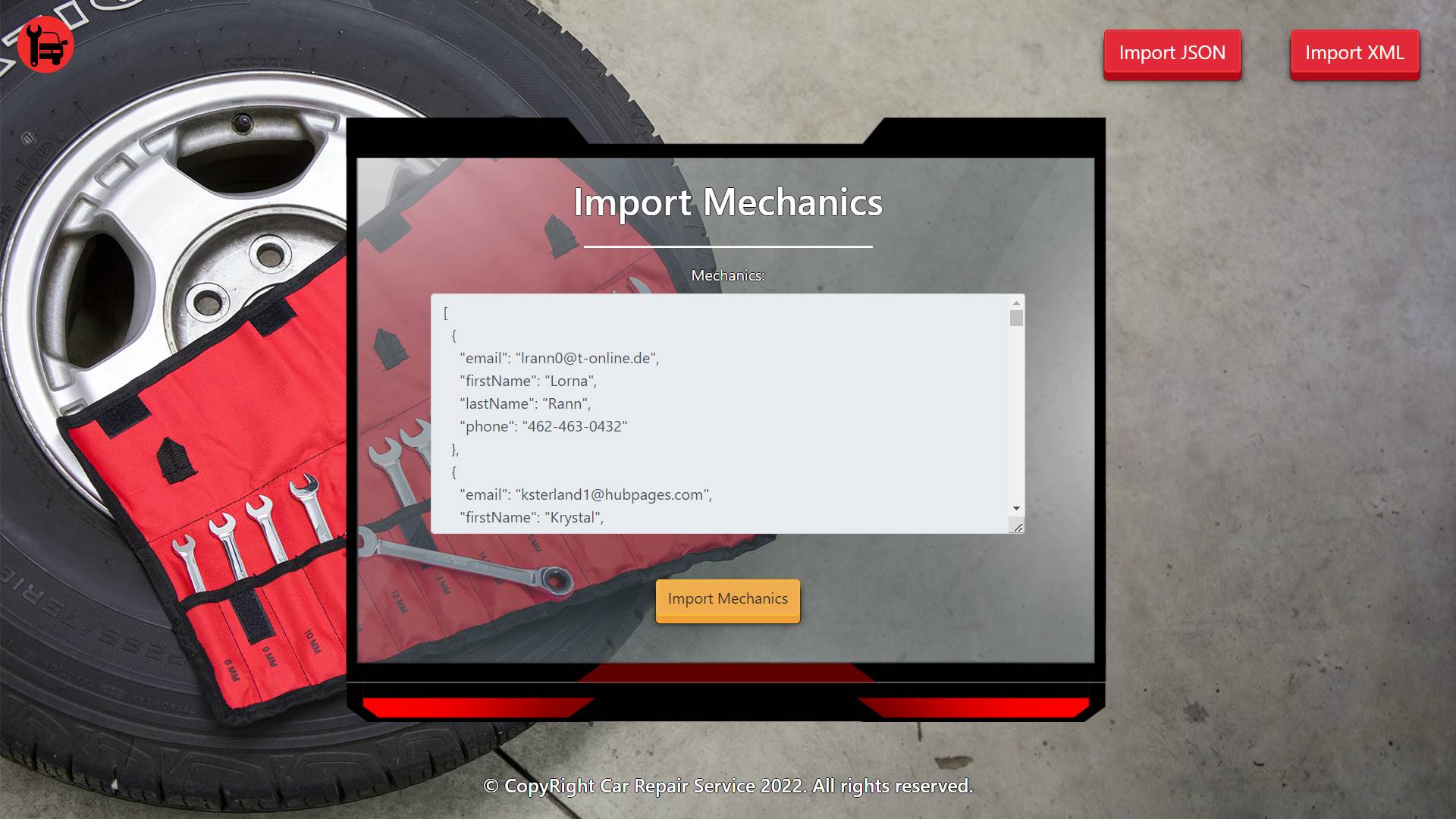
* The import JSON page before importing anything:



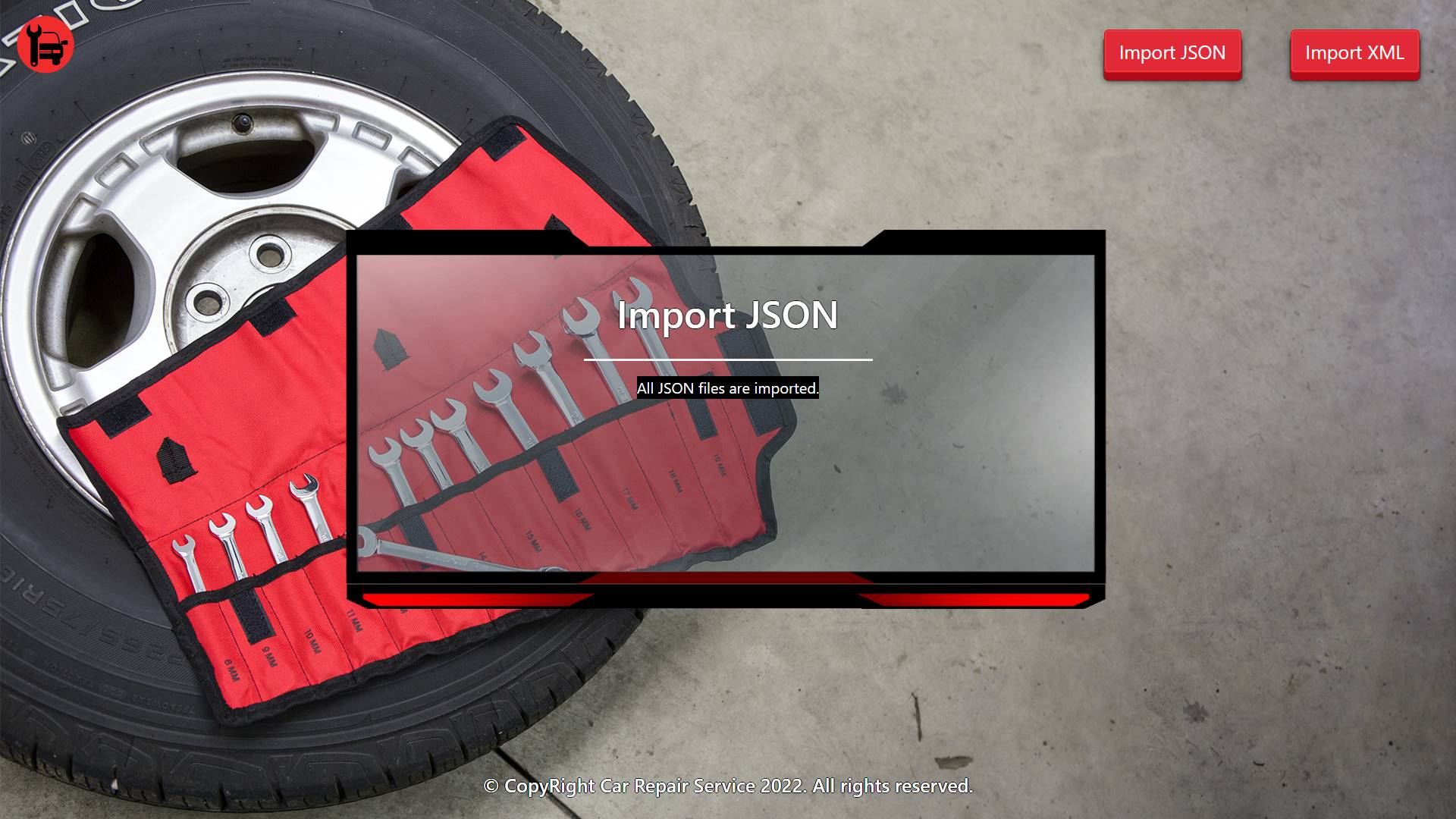
* Import the parts first:



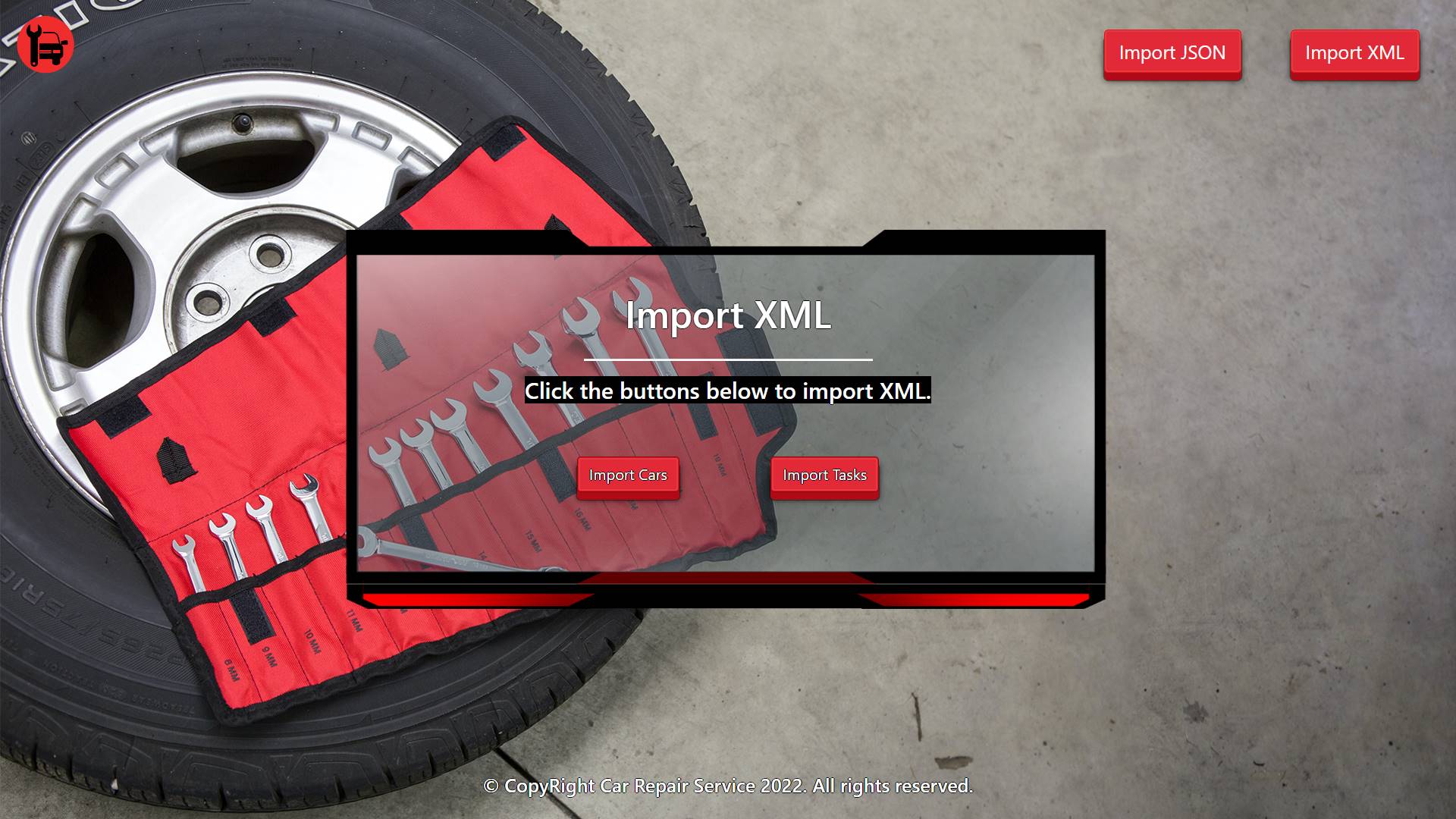
* Import the mechanics second:



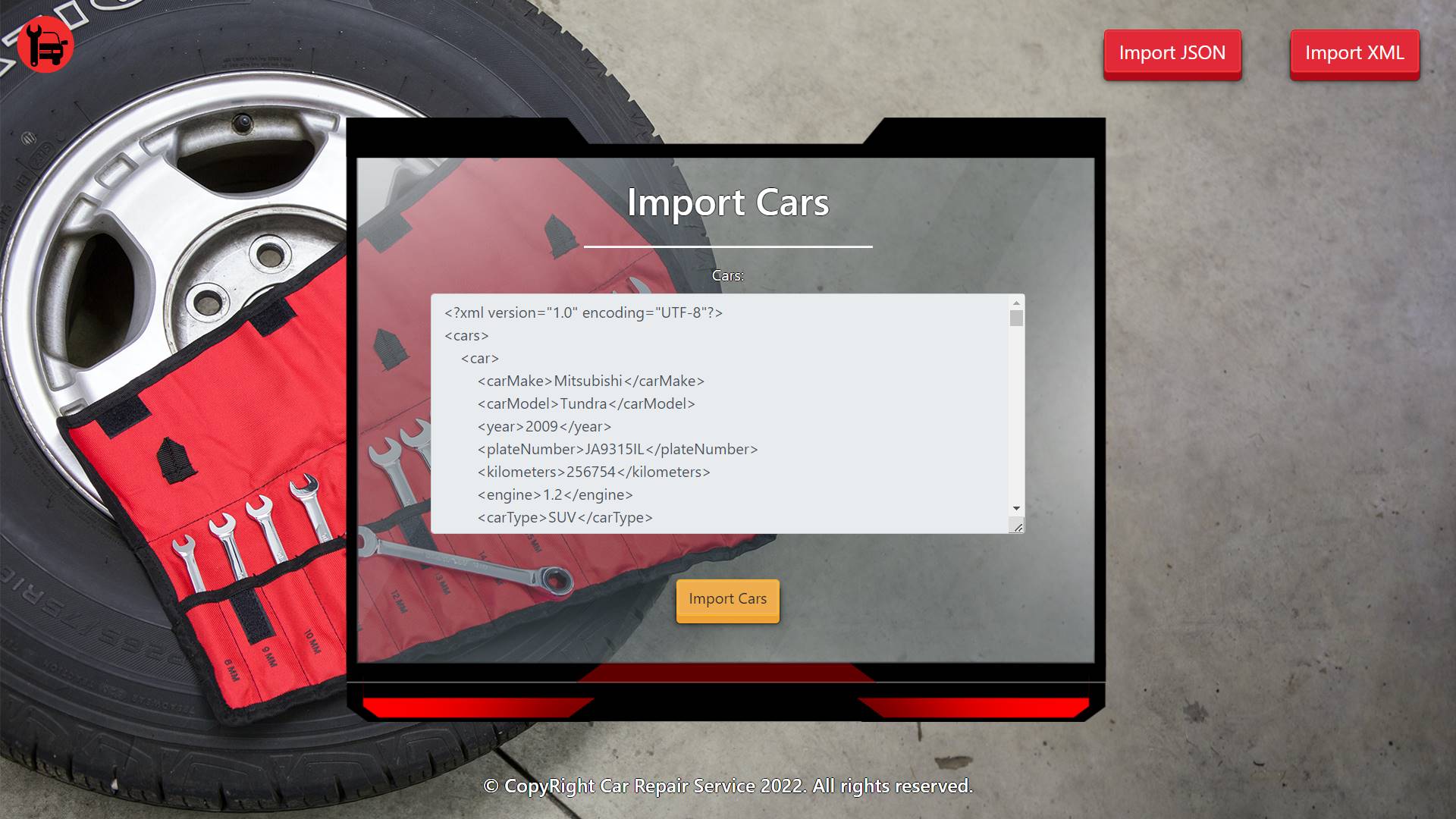
* The import JSON page after importing both files:



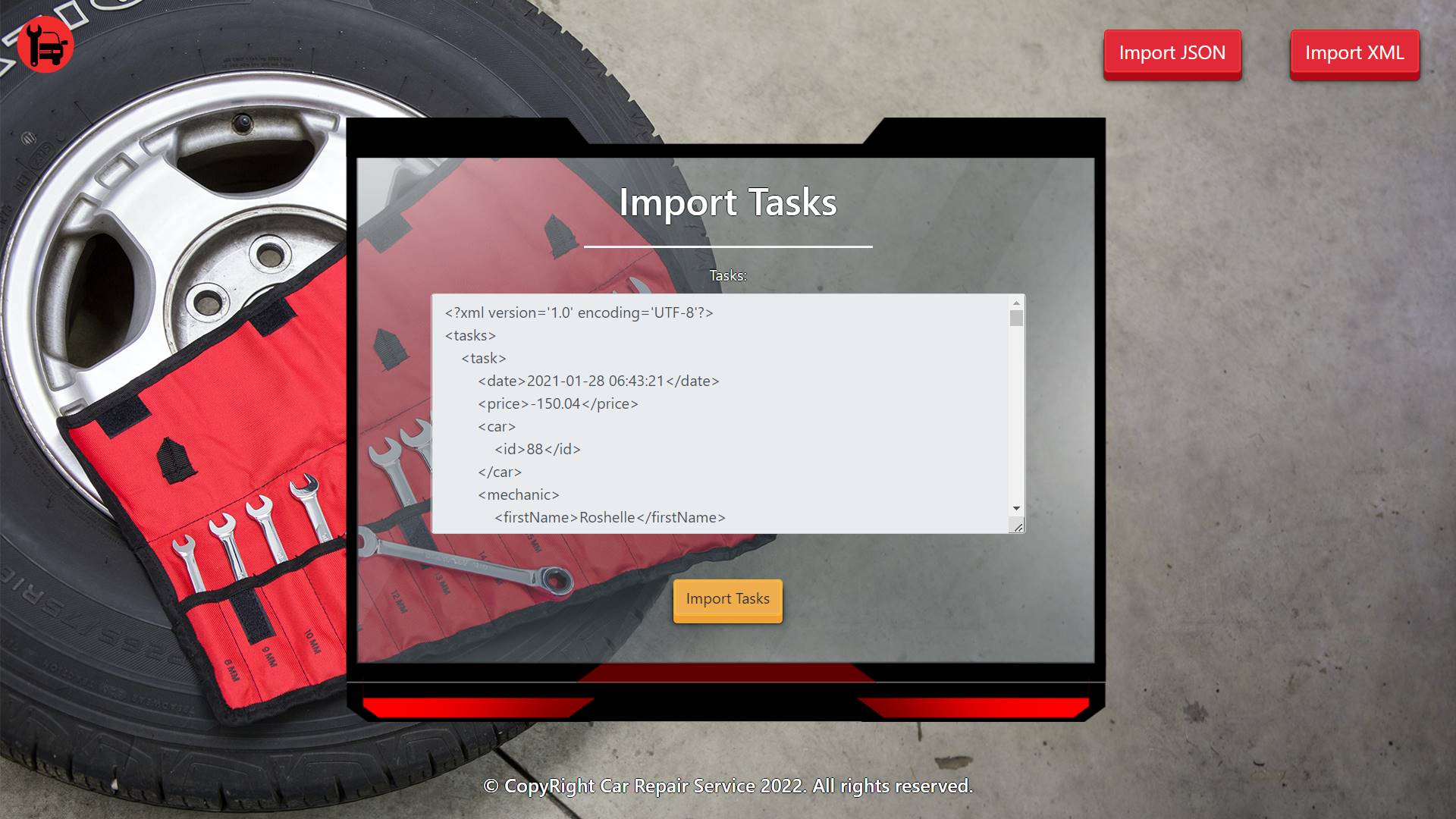
* The import XML page before importing the given data:



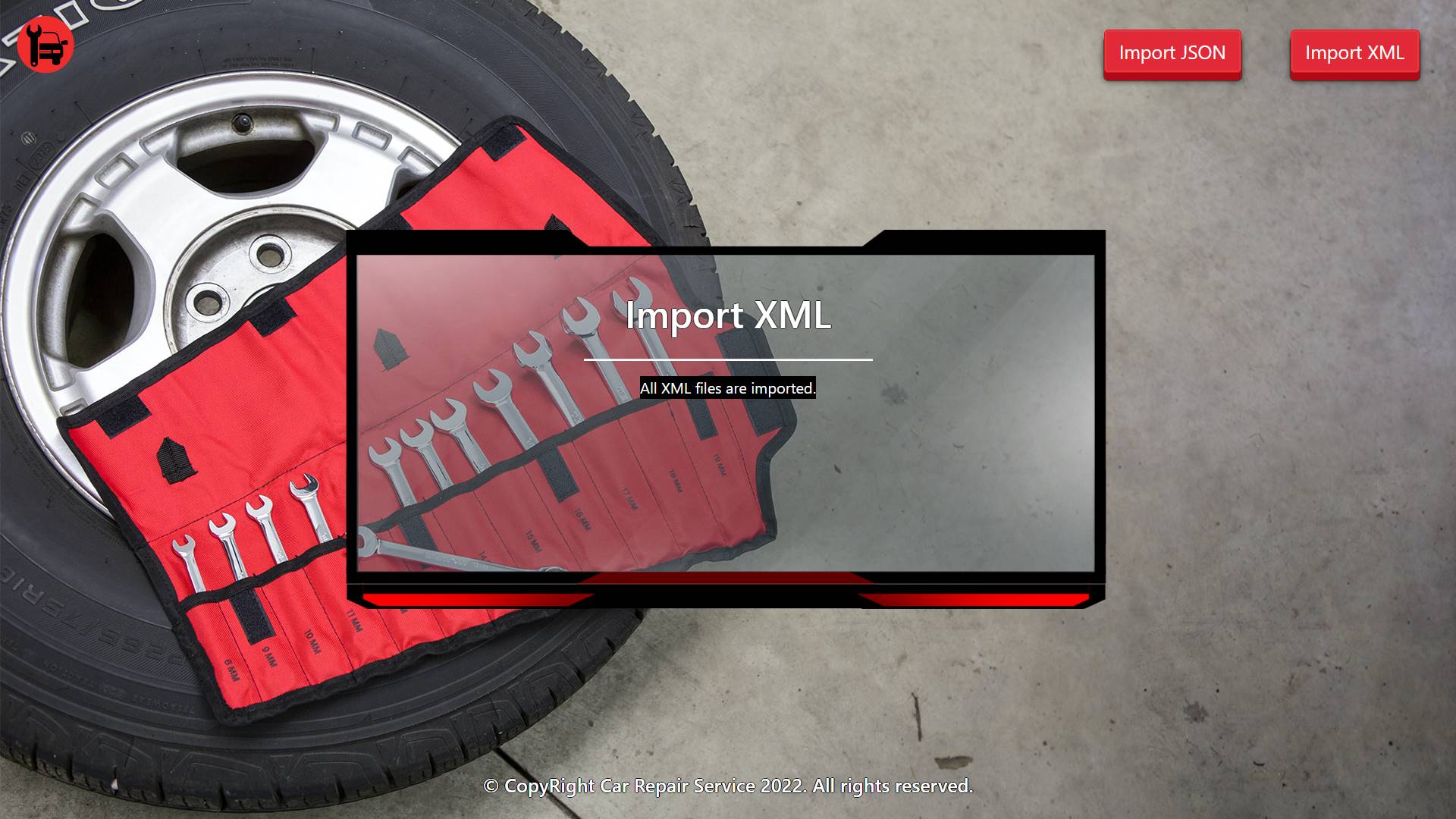
* Import the cars data:



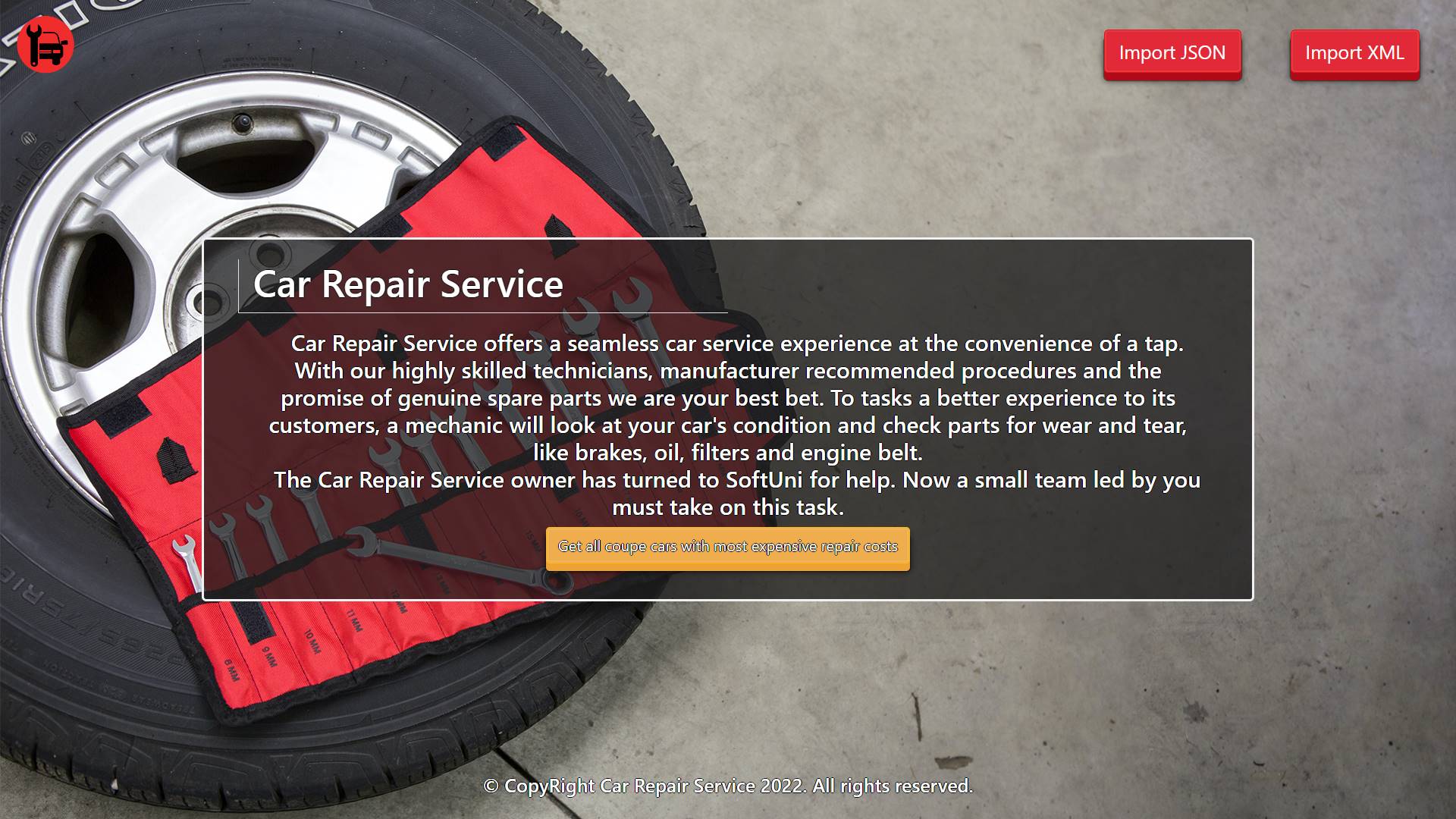
* Import the tasks data:



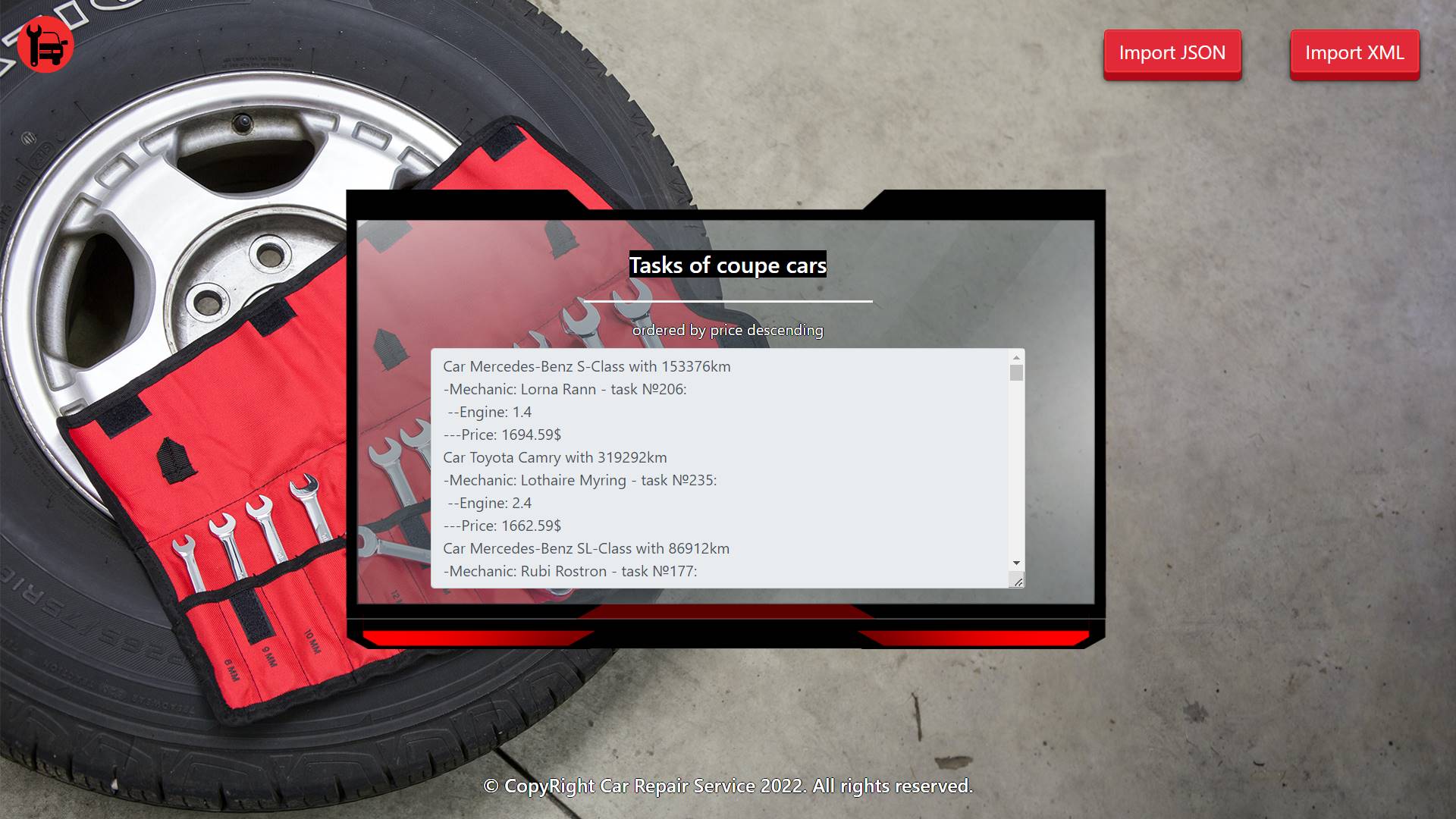
* The import XML page after importing the data:



* The home page after the data is imported:



* Export tasks for coupe cars:



## Model Definition

There are 4 main models that the **Car Repair Service database** application should contain in its functionality.

Design them in the **most appropriate** way, considering the following **data constraints**:

### Part

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* part name – accepts **char sequence** (between **2** to **19** inclusive). The values are **unique in the database**.
* price – a **floating point number**. Must be between **10** and **2000** (both numbers are **INCLUSIVE**)**.**
* quantity – accepts a **positive number**.

### Mechanic

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* first name – accepts **char sequences** as valueswhere their character length value **higher than or equal to 2.** The values are **unique in the database**.
* last name – accepts **char sequences** as valueswhere their character length value **higher than or equal to 2.**
* email – an **email –** (must contains ‘@’ and ‘.’ – dot). The email of the mechanic is **unique**.
* phone – accepts **char sequences** as valueswhere their character length value **higher than or equal to 2.** Can be **nullable.** The values are **unique in the database**.

### Car

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* car type – the enumeration, one of the following – **SUV, coupe, sport**
* car make – accepts **char** **sequence** (between **2** to **30** inclusive).
* car model – accepts **char** **sequence** (between **2** to **30** inclusive).
* year – accepts a **positive number**.
* plate number – accepts **char** **sequence** (between **2** to **30** inclusive). The values are **unique in the database**.
* kilometers – accepts a **positive number**.
* engine – accepts **number** values that are **more** than or **equal** to **1.00.**

### Task

* id – accepts **integer** values, a **primary identification field, an auto incremented field**.
* price – accepts a very big **positive number**.
* **date** – a **date** and **time** of registering the task in the "**yyyy-MM-dd HH:mm:ss**" format.
* **Constraint**: The task table has a relation with the parts table.
* **Constraint**: The task table has a relation with the mechanics table.
* **Constraint**: The task table has a relation with the cars table.

#### Relationships

One **Task** may have only one **Mechanic**, but one **Mechanic** may have many **Tasks**.

One **Task** may need only one **Part**, but one **Part** may be used in many **Tasks**.

One **Task** may have only one **Car**, but one **Car** can be in many **Tasks**.

**Constraint**:

* Name the entities and their class members **exactly** in the **format stated** above.
* All fields are **NOT NULL** unless explicitly stated to be nullable.

