

# Assignment 2:

## An Online Car Rental System using AJAX and JSON

Due: Sunday 26 May 2024, at 11:59 pm

### Introduction

In this assignment, you are required to develop a dynamic website with interactive features for a car rental company. The website should provide a streamlined online car renting experience for potential users. It should be able to handle various exceptions, such as input validation and change in car availability.

The website should enable users (i.e., potential customers) to

- Browse cars by type or brand,
- Search for cars using keywords,
- Check basic info and availability of a car,
- Make a rental request (reservation) for an available car,
- Cancel or confirm a rental order.

This assignment accounts for 35% of the total mark in the subject. You are required to complete this assignment individually.

### Objectives

- Learn about design of practical websites with rich interactive features for a specific application.
- Learn how to design customised data structures using JSON.
- Learn how to load, parse, and store JSON files using JavaScript.
- Learn how to set, update, and delete cookie and/or session.
- Learn how to use AJAX technique to send and retrieve data from the web server asynchronously without interfering with the display and behaviour of existing web pages.

### Visual Components

#### 1) Website logo

There should be a logo for the website. The logo should appear on all pages of the website.

#### 2) Search box

There should be a search box allowing users to look up cars by using keywords.

#### 3) Car categories

Cars should be categorised by type (e.g., Sedan, Wagon, SUV) and brand (e.g., Ford, Mazda, BMW).

Users can click a category (if non-empty) to view all the cars in the category.

Search box and car categories should appear on all except the reservation and order confirmation pages.

#### 4) A grid view of cars

When multiple cars are shown on a page (e.g., the homepage, the page showing cars by category, or search results), the cars should be aligned to grid (i.e., a nice tabular format).

Each car should show the key information, e.g., [car model](#), [image](#), [price per day](#), and [availability](#).

Beside each car, there is a button named “rent” through which users can enter the reservation.

#### 5) Reservation

User can view the car that she/he intends to rent ([car model](#), [price per day](#)) and edit the [quantity to rent](#), [start date](#) and [end date](#) of the renting, and the [total price](#) for the rental duration.

User can 1) cancel this reservation, 2) proceed to place an order for the car, or 3) simply leave this page.

In the third case, the reservation information will be kept in Cookie/Session; so, the user can later resume the previous reservation. When the user presses the “rent” button next to another car, the reservation will change for making the rental booking for the new car.

User must provide [name](#), [mobile number](#), [email address](#) and [valid driver’s license available](#) to place an order. All are compulsory for the order to be placed.

**User can only rent one car model at a time.**

## 6) Order confirmation

Once an order is placed, the page shows a web link which the user can click to confirm the order. The rental order is not placed until the user clicks the Web link.

This confirmation will trigger an update of the JSON file and MySQL database.

## Data Structures

- Design a JSON file “[cars.json](#)” to store information about cars. There should be at least 30 cars in the JSON file. Each car should have at least 10 attributes (as shown in the table below). For each car, the availability status should show to the user as either ‘Yes’ or ‘No’.
  - The website should show car information by loading data from the JSON file.
  - The total number of cars available for each car model should be 1, 2 or 3 by default.

Type	Brand	Car Model	Image	Mileage	Fuel type	Seats	Quantity	Price/day	Description
sedan	Toyota	Camry 2013	xxx	10000	petrol	5	2	\$120	xxxxxxx

- Design a MySQL table named “[orders](#)” to keep track of all rental orders. The table should consist of at least such basic information as “[user\\_email](#)”, “[rent\\_start\\_date](#)”, “[rent\\_end\\_date \(inclusive\)](#)”, “[price](#)”, and “[status](#)” to indicate whether the order has been confirmed by user.

## Requirements

- Use HTML, CSS, JavaScript, AJAX and PHP to make an interactive website.
- Use Cookie, Session, JSON and MySQL to store information either temporarily or permanently.
- Use PHP to interactive with MySQL – You decide where to store images (database or the file system).
- We recommend that you use [Cloud9](#) or a [local IDE](#) (e.g., Visual Studio Code) for development.
- You are required to deploy your website to [Elastic Beanstalk](#) for marking purposes.

## Submission Process

- 1) Upload a zip file of your source code to [Canvas](#), then click “Submit Assignment”.

The screenshot shows the Canvas submission interface. At the top, there are three tabs: 'File Upload' (selected), 'Text Entry', and 'Website URL'. Below the tabs, there is a text prompt: 'Upload a file, or choose a file you've already uploaded.' Underneath this, there is a 'Choose File' button and the text 'No file chosen'. To the right of this, there is a link: '+ Add Another File'. Further down, there is a link: 'Click here to find a file you've already uploaded'. Below that, there is a 'Comments...' text box. Under the comments box, there is a checkbox labeled 'I agree to the tool's End-User Licence Agreement.' with a small icon to the right. Below the checkbox, there is a line of text: 'This assignment submission is my own, original work'. At the bottom, there are two buttons: 'Cancel' and 'Submit Assignment'.

- 2) Provide the URL of your website as a comment to your submission.