

# Assignment 2 (6 points total)

In this assignment we will practice making asynchronous function calls in JavaScript and learn how to make requests with JavaScript after we have loaded the page. We will also work with cookies and see where and why they are used in web development.

Prerequisites:

- Have Node and Express installed
- Install body-parser and cookie-parser NPM packages

## Task 1 (2 points)

In this task you will have to create a simple web application using Node and Express for working with a car dealer stock. Your application should at least have the following routes implemented:

- GET / - Returns a list of all cars available
- POST / - Adds a car to the list
- GET /:carID - Returns a car with the specified ID
- DELETE /:carID - Deletes a car with the specified ID

The list of the cars should persist when the server is restarted. You must generate an ID on the server when adding a new car to the list. If you wish, you can have additional routes, for example modifying a car or retrieving all cars of a given make, filtering them by production year, etc.

Hints:

- Use fs module in Node to work with file system
- You can use callbacks, Promises or async/await for asynchronous calls
- util.promisify can turn callback-based Node functions into Promise-based versions
- Use body-parser to retrieve data from the request body
- You can use Array.filter() or Array.splice() to remove elements from an array
- Use JSON.stringify() and JSON.parse() to easily convert your JS objects to and from strings, respectively

## Task 2 (2 points)

In this task you need to create a web application to monitor your PC's RAM usage. Your app should refresh the usage every second by making a request to a server **without** reloading the page. Your app should display absolute and relative RAM usage. For absolute value, you should have 4 buttons to select the measurement units from bytes, KB, MB and GB. Show the value up to 2 decimal digits at most. For relative values show a percentage rounded to 2 decimal digits at most. The text should be also colored based on the amount of free RAM as follows:

- Green if the usage is below 60%

- Default (black) if the usage is between 60 and 80%
- Orange if the usage is above 80%
- Red if the usage is above 90%

If your webpage needs the total amount of memory, it must get this value **from the server** as well.

Hints:

- You can use `os.freemem()` function to get the amount of free RAM
- You can use `fetch()` to make requests to your server once the page is loaded
- You can use `style` property of HTML elements in JS to change the CSS of the element directly

### Task 3 (2 points)

In the last task, you have to create a simple event countdown app. Your webpage should have a form for adding a new event, a list of all currently added events with their names and countdown text showing the number of days, hours, minutes and seconds until the event and a button to remove all events. Use an **HTTP-only cookie** to store events for this task. Your server must use EJS to render the list of events upon request. The events should persist when restarting the browser and be cleared after 24 hours.

Cookies have a (very large but not infinite) size limit. Avoid storing any extra data in the cookies. In this task you should only store the name and time of the event in the cookie. If you want, you can make the countdown text to be updated every second, but **do not** reload the page.

Hints:

- HTML forms have a date input with a convenient date selector
- JS includes a Date object useful when working with dates
- Use `res.cookie()` and `res.clearCookie()` to work with cookies in Express

## IMPORTANT INFORMATION

Submit all of your sources to the assignment in OMA before the deadline for full points.

Assignments submitted even a moment after the deadline will only be able to get half points.

**!!!There is no exception to this rule!!!**