Students are allowed to work in group of two or individually. The purpose of this project is to provide practical work for students in JavaScript and JQuery. Late submissions are not allowed and will not be accepted. Any form of plagiarism is not allowed.

In this task create completely new 3 Web Pages. In this project, you will connect to your own custom fake API created using Git (to get Mock JSON data):

- Create your own fake online REST server to connect and get data from. Follow instruction here:
  - https://mockend.com/
  - https://www.youtube.com/watch?v=NEQa9U6zONA&ab_channel=CForCoding
- In total create 5 different resources to connect to and get data from. The resources can be any but make sure each resource contains at least 10 JSON objects (JSON array with 10 JSON objects).
- Make sure each JSON object of the same category contains the same property. For example:
  - Let's assume, you created resource "/cars"
  - Each JSON object in "url/cars" must contain the same properties (for example):


```
{
    "modelId": 10325,
    "id": 1,
    "modelName": "Mercedes-Benz",
    "image": "https://upload.wikimedia.org/wikipedia/commons/f/f6/2018_Mercedes-Benz_A200_AMG_Line_Premium%2B_1.3_Front.jpg",
    "price":  24000
}
```

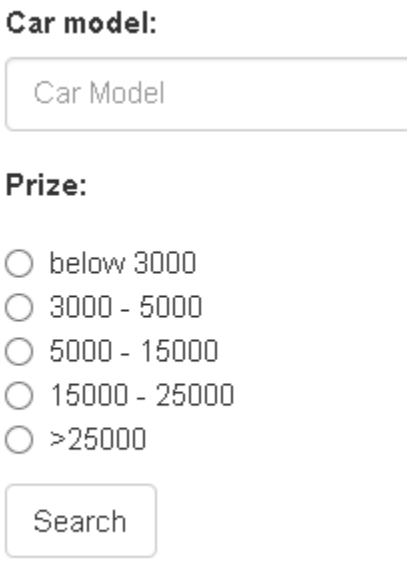

Figure 1.

  - Next JSON object in the array in "url/cars" must have different "id", "image", "modelId", etc
- The topic of your web page can be any, but make sure it is consistent throughout all 3 Web pages (For example, if you create a web page about cars, make sure your Mock API contains data about cars)
- Also create a resource for POST method. When user registers to your web site the data is sent to "url/posts". It must contain email and passwords of each user. You can add initial JSON array containing at least 5 already registered users (JSON objects with properties: "email", "password", "id").

In your HTML pages make sure user can access this data asynchronously using either JS or JQuery:

- Create a registration form:
  - If user does not have login and password, he/she can register to you web site by providing email and password. Make sure you validate email and a password using REGEX.
  - When user registers if email and password is already in your fake server, it should notify user that email is taken
  - When user registers data is sent to POST resource of your fake API and if the process is successful **modal** pops up indicating the user is logged in
  - If user is already registered to the web page, he/she have to **login** in this case do the following:

        - Send a GET request to POST resource, to extract all JSON objects (each contains email and password)
        - Compare login and password from the **login** form input fields to the JSON objects values. If it matches user can login, otherwise indicate that email or password does not match
- Create **search** to extract data from fake API you created**. For example**, assuming we are connecting to resource "url/cars", that contains JSON objects shown on Figure 1. You can create a from for a search with following fields:
    - Car model. User can type in what car model he/she is searching for
    - Prize range



    - If user enters car model without prize range all cars of this model will be returned
    - If user enters prize without model it must returns all models with selected prize are returned
    - The response (JSON objects) must be presented to the user in the most accessible way. The cars, in this case, are presented as pictures with description.
    - **Please, remember for your work you can choose any theme and JSON response as well as forms and presentation can be different, just make sure it is done in the similar manner, with similar functionality**
    - To process data write multiple JS functions and use Promises

Please submit all the codes in .zip format with name of both group members.