# 1  Practice 7.1. Files and cookies

This practice is the next in the series of practices consisting of a Contacts application. This practice is based on the previous one. Before start it, finish the previous one, make the corrections you consider, if needed, and start coding.

The inclusion of comments in the scripts will be valued.

Before start, create a new git branch named `contacts-u7` and change to it:

```
git branch contacts-u7
git checkout contacts-u7
```

Do all the practice in the `contacts-u7` branch. Once finished, do the submission via GitHub Classroom as detailed in the end of the practice.

## 1.1  Exercise 1

Create a new page, named `import_export.php`, with a form and a button for downloading all the database contacts (all the fields) in a JSON file, with the `.json` extension. The file must be downloaded automatically once the button is pressed.

The json file must be well formed and must contain a main array named `'Contacts'` with all the contacts. For example:

```json
{
  "Contacts": [
    {
      "id": 1,
      "title": "Mr.",
      "name": "Mike",
      "surname": "Molina",
      "birthDate": "1975-10-21",
      "phone": "555445466",
      "email": "molina@mail.com",
      "favourite": true,
      "important": true,
      "archived": true
    },
    {
      "id": 2,
      ...
```

> **Recommendation**: Create a directory with write permissions in order to store your files.

Create a partial script with a navigation bar to navigate between this page and the rest of the app pages, `contact_list` and `contact_form`.

### 1.2 Exercise 2

In `import_export.php`, add an input of type file and another button to upload a JSON file and insert it in the database. The json file must be well formed and must contain a main array named `'Contacts'`, like in the previous example. The file must have the `.json` extension.

### 1.3 Exercise 3

In `import_export.php`, add a `<select>` input with a list of 3-4 languages. When the user chooses a language, the script must store a cookie with the chosen language that will expire en 10 days.

In all your pages, read the language cookie and display a message according to its content (for instance "Your language is English"). You can display the message in the header, navigation bar or footer parts.

### 1.4 How to submit to GitHub Classroom

1. This task must be submitted to the same repository of the previous one. Remember to make a new branch before starting the practice.
2. Once you finish the task, make a commit with the comment "PRACTICE 6.1 SUBMISSION COMMIT", merge the branch into the main branch, and push it to GitHub. For example:

```
git commit -m "PRACTICE 7.1 SUBMISSION COMMIT"
git checkout main
git merge contacts-u7
git push
```

3. Before that, you can do the commits and pushes you want. If you change your code after your submission commit, make another commit and push with the same text in the message adding the corrections you've done.Make a pull-request if you want so i could know your code is submitted.
4. Once pushed to GitHub, create a tag with the name "Practice 7". Create a new tag if you make corrections after the first submission (ex. "Practice 7 corrected").

If you have any doubt in your task, you can push your code and ask me by email what's your problem.