

1 Practice 7.2. Voluntary

This practice is based on the previous one (7.1). Before start it, finish the previous one, make the corrections you consider, if needed, and start coding.

The mark of this practice **can increase the mark of the first evaluation by 0.5 points**, provided that the marks of the practice part and the exam are at least 5. The note that increases each exercise is indicated on it.

Before start, create a new git branch named `contacts-u7.2` and change to it:

```
git branch contacts-u7.2
git checkout contacts-u7.2
```

Do all the practice in the `contacts-u7.2` branch. Once finished, do the submission via GitHub Classroom as detailed in the end of the practice.

1.1 Exercise 1

(+0,2 points)

In `import_export.php`, a button for downloading all the database contacts (all the fields) in a **CSV** file. The file must be downloaded automatically once the button is pressed.

Also, add an input of type file and another button to upload a CSV file and insert it in the database.

The file must have all the fields of a Contact object in the same order, without header.

1.2 Exercise 2

(+0,3 points)

Make a new functionality to add images to each contact, storing them in the database. Add a field in the contact table of the database with the next properties:

```
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
...
| image | longblob | YES |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
```

The image must be shown properly in the `contact_form` view, only if the contact has an image stored. The image must be uploaded in the form.

Establish the next limitations: * file types: jpg, jpeg, png, gif * size limit 5MB.

1.3 How to submit to GitHub Classroom

1. This task must be submitted to the same repository of the Practice 4.1. Remember to make a new branch before starting the practice.
2. Once you finish the task, make a commit with the comment "PRACTICE 7.2 SUBMISSION COMMIT", merge the branch into the main branch, and push it to GitHub. For example:

```
git commit -m "PRACTICE 7.2 SUBMISSION COMMIT"
git checkout main
git merge contacts-u7.2
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 7.2". Create a new tag if you make corrections after the first submission (ex. "Practice 7.2 corrected").

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