BRAIN-TEC

ASSIGNMENT: *“PHP ADDRESS DATABASE”*

1)First Sights and Decisions

As talked with Mr. José Luis Benito, the first decision I took was to use Python instead of PHP because I think it would be a better indicator for the job since I apply for a Python Odoo developer job, so I've been using Python with Flask to do de assignment.

2)Aproaching the problem

The first step was to create the database so I wrote the next SQL code (written here for most readability):

```
CREATE DATABASE IF NO EXISTS exercise CHARACTER SET utf8;
USE exercise;
DROP TABL IFEXISTS contact;
CREATE TABLE contact (
         contact_id        MEDIUMINT NOT NULL AUTO_INCREMENT,
         firstnameVARCHAR(100) NOT NULL,
         lastname  VARCHAR(100) NOT NULL,
         address   VARCHAR(200),
         /* Since this is a contact-book, the email can't be repeated... */
         email             VARCHAR(100) NOT NULL UNIQUE,
         /* […] neither the phone number. */
         phone             VARCHAR(30) UNIQUE,
         PRIMARY KEY (contact_id)
);
```

Besides the primary key, I setted the mail and the phone number as unique, because we don't want two contacts with the same information in our contact book.

For this application I used MVC so I created the folder structure of the project and planned the components:

- **Model:** Just one, the Contact model.
- **Views:** Here we have at least two different views, the form one to create contacts and the list to view it.
- **Controllers:** Here I added the code that controls the routing of the application.

3)WorkFlow and thoughts

First I coded the `__init__.py` file to initialize the Flask application and so I wrote the `bd.py` middleware to ease the connection and manipulation of the database.

After this I wrote the Contact model with all the methods required and the test for these methods. I wrote the methods to pass the tests and then I created the views (called templates in Flask) and the controllers to control the routing.

After this, I realized that I'd needed some other functions outside of the Contact model like `get_database_contacts()` which gets all the database contacts or the function `load_xml_contact(xml_file)` which loads all the contacts in a XML file and returns an array of contacts so I created the tests for these functions.

After write the tests of these functions I coded the functionality to pass the tests. Writing this functionality I read in the Python ElementTree documentation that therer are some XML attacks to wich the ElementTree library was weak so I used the recommended library `defusedxml` to sanitize the XML file.

After all the functionality was working I wrote the views/templates of the contact form and the contact list, to create a more usable experience I used the Bootstrap 4 Framework, and then I wrote the controllers for the contact form and list and the uploader of XML files.

For the code that modifies the DOM, to change colors and create a dynamic experience, I used Javascript and JQuery served as static files using the Jinja2 utilities of Flask for it.

This was the main route but along all the process I rewrite some tests to adapt to the restrictions that I was finding so I refactored the code once and again and created some boilerplate code for the Flask application like the `config.py` file and so on.

4) Final thougts and commentaries

After completing this assignment I think that in this project hides more in it than I first thought, like the sanitization of the SQL statements or the XML files defuse to avoid security issues. It was a little project but I learned a lot of it.

Please, read the README.md file inside the project to know how to setup it and run it.