Development of a web map application with Flask

This is a first test project before hiring you on a long term basis for the real project.

For this one week project I have a budget of $60 since it is not a long project.

Methodology

Using the Flask framework, you will have to develop a web map application that allows the online and persistent edition of the urban trees of a city.

The application data will be stored in three different tables:

Species table: in this table you will be able to add data related to any tree species. It will contain the following information:

- `id`
- `scientific name`
- `common name`
- `family`
- `maximum height`
- `beginning of flowering`
- `end of flowering`

Gardeners table: in this table data related to the gardeners can be added. It will contain the following information:

- `id`
- `name`
- `surname`
- `email`

Trees table: in this table you can add data related to each of the city's trees. It will contain the following information:

- `id`
- `localitation`
- `planting date`
- `state of conservation`
- `height`

- `diameter`
- `last pruning`
- `gardener (in charge of maintenance)`
- `species (to which the tree belongs)`

The location will be a field of type geometry.

The gardener and species field will be a foreign key of the Gardeners and Species tables, respectively.

Private part of the application

The application will have a private part to which only registered users will have access. The users' password will be stored in the database in encrypted form. Through this private area, it will be possible to enter, edit and delete new species, new gardeners and new trees.

As for the trees, their location can be entered or edited interactively through a map.

Public part of the application

The public part of the application displays an interactive web map with the location of all the trees in the Trees database. Clicking on each tree will display a pop-up with the following information: species name, tree height, tree diameter, conservation status and date of last pruning.

In this same public interface there will also be a link for registered users to access the private management area by entering their credentials: email and password.

Aspects to consider

In the models, date data will be stored in a field of type Date.

When creating the geosjon, you must take into account that the fields related to decimal values, dates and foreign keys cannot be serialized in JSON format. To avoid problems, you can use the simplejson and json libraries to encode them. For example:

- To encode a field with decimal values: simplejson.dumps(tree.height).
- To encode a date field: json.dumps(tree.planting, default=str). In this case, we convert the values to string.

Software

The software necessary for the realization of the practice is the following:

- PostgreSQL/PostGIS to store the data.
- A tool to create virtual environments
- The Python framework Flask.
- Flask extensions: Flask-SQLAlchemy, GeoAlchemy, Flask-Bootstrap, shapely-geojson, Flask-Admin, Flask-Migrate and Flask-Login.
- OpenLayers, for the creation of the map in the application.
- Plain text editor (Atom, Sublime Text, Notepad++, etc.), for the development of the application.

What you should deliver

To successfully complete the first project you must deliver a compressed file in .zip format containing all the files of the Flask project.

You do not have to deliver the virtual environment. You just have to include the requirements.txt file with the information of the packages to install in the virtual environment. Remember that with the pip3 freeze command you can get all the packages and versions installed in your virtual environment.

What will be evaluated

The following aspects will be taken into account:

- Incorporation of all the functionalities required in the activity statement.
- Correct implementation of the models, according to the indicated requirements.
- Correct configuration of the ORM
- Creation of a template to visualize the application data in an interactive web map.
- Implement user login