

Pecan - Python Coding Task

Maximum duration: 2 hours

You have been provided with a zip file of a toy Python 3 Flask application, called Pecan. Extract it to a directory, and run it with the following:

```
pip install flask
FLASK_APP=main.py flask run
```

You should see output in your console including:

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Pecan is an API which shows some information about customers, including about their personality. All the available customer data (for 250 customers) is included in the file `customers.json`.

Tasks

Initially, Pecan only supports filtering by gender (e.g. <http://localhost:5000/?gender=Male>).

Modify the `get_customers` function in `main.py` to support different types of filtering and/or sorting. Use `README.md` to document any new features and tests that you add.

Tips

- **What types of filters could you add?** Some of the fields are categories (like gender), but others are numeric (like age) or dates. What is the most useful way for users to filter on these fields?
- **What is the most useful way to add sorting?** Some fields may be more obvious than others - think about what users of your service might find most useful.
- **Are there any edge cases?** What is the best way to handle these and how can you test that your handling works?
- **What is the best way to structure your code?** Feel free to add new functions, classes or files where you think it is appropriate. Imagine that another developer will have to extend this in future - how can you make their job easy?

You can add anything to Pecan that you wish to complete this task. You can change anything except the `customers.json` file.

Submission

Zip your full application up and send it back to us, along with any instructions for running it if it needs more than `FLASK_APP=main.py flask run`. Also include any instructions for how to run tests that you have written.

You will be assessed on:

- The usability of any new features you implement
- The quality and cleanliness of your code and tests

If you want, you can also include with your submission a list of suggested features that you would add if you had additional time.

Special Note!

Don't try to complete everything!

Just be sure to show us enough that we can get an impression of your coding skills - spending hours and hours implementing dozens of features is unnecessary. Remember that we are looking for readable, maintainable and well-tested code.

Have fun and if you have any questions then get in touch!