Milo

Software • Safe • Reliable

Django task

1. Estimated development:

2-3 Hrs. If the task takes longer than expected, stop at any point when you feel comfortable with the code.

2. Important Notes:

Please note that you will be judged on the overall quality of the code (readability, best practice compliance, etc.) and approach to the solution. It IS OKAY to use third party packages as it is a common practice in Django.

3. Specification:

The task is made up of 5 parts plus an optional task.

1. Set up a basic (latest or LTS) Django installation

2. Extend the User model to have (yes, custom user model is the way to go):

a. a birthday field of type date

b. a random number field of type integer that is assigned a value from 1-100 on creation

Do the changes in user model in a few steps - we want you to use schema migrations here.

3. Create views for: list of all users, viewing, adding, editing and deleting a single user

4. Create two template tags:

a. a tag that will display "allowed" if the user is > 13 years old otherwise display "blocked"

b. A tag that will display the BizzFuzz result of the random number that was generated for the user. The BizzFuzz specification is that for multiples of three print "Bizz" instead of the number and for the multiples of five print "Fuzz". For numbers which are multiples of both three and five print "BizzFuzz"

c. Add a column to the list view after the birthday column that uses the allowed/blocked tag

d. Add a column to the list view after the random number column that uses the BizzFuzz tag

5. Unit test what you feel is appropriate to test.

4. **Optional task:**

Create a download link on the list view. The link would return the list of results in Excel's format (csv format is ok).

Sample Output:

Username	*Birthday*	*Eligible*	*Random Number*	*BizzFuzz*
user1	1/1/2013	blocked	40	Fuzz
user2	11/4/1975	allowed	66	Bizz
user3	7/30/2010	blocked	51	51
user4	6/16/1968	allowed	30	BizzFuzz

Expected Deliverables:

- Code
- Installation instructions - it should be possible to install the project flawlessly. For example the installation steps can be:
    a. git clone from github (or other free git repository) (you should provide url address)
    b. create a virtualenv
    c. pip install requirements.txt (you should have requirements.txt file in your project)
    d. python manage.py migrate
    e. python manage.py runserver
    f. after those steps we should see working Django project in the browser

- List of what was done and wasn’t, e.g. a release note.
- Screenshot of screens.
- Excel file if optional task was completed.