# AlgoBulls Backend Developer (Web)
# Coding Assignment
May 2023

**Project:** Design a simple To-Do List App using Django

**Description:** You need to design backend for a web-based To-do List application, as per the given requirements:

1. Create a Django app with appropriate models to store information for the To-do List app. The app should be able to store the following information:
   a. **Timestamp**: Timestamp at which a task was created.
      Should be auto-set when creating a new entry. A user should not be able to edit this.
   b. **Title**: Title of the task to be done.
      i. A user can set this while creating a new entry. A user can also change this while updating an existing entry.
      ii. Max length: 100 characters.
      iii. Mandatory field
   c. **Description**: Description of the task to be done.
      i. A user can add details about this task.
      ii. Max length: 1000 characters
      iii. Mandatory field
   d. **Due Date**: Expected due date to finish the task
      i. A user can set this while creating a new entry. A user can also change this while updating an existing entry.
      ii. Optional field
   e. **Tag:** One or more tags that the user can add to the entry
      i. A user can set this while creating a new entry. A user can also change this while updating an existing entry. Multiple tags can be added to the same entry
      ii. Optional field
      iii. Multiple tags with the same value should be saved only once.
   f. **Status**: Shows status of a task
      i. Should be one of these values.
         1. OPEN (Default value)
         2. WORKING
         3. DONE
         4. OVERDUE
      ii. Mandatory field
2. Django Admin interface should be enabled for the model(s). The admin site should have the following:

a. Appropriate validation checks are in place. The ones defined in Point 1 above should be enforced.
(Ex: In 1.a, *A user should not be able to edit the timestamp.*)
b. Proper *changelist* view with filters for every model
c. Proper *fieldsets* for every model
3. The following REST APIs should be created using DRF (DjangoRestFramework):
    a. CREATE a todo item
    b. READ one todo item
    c. READ all todo items
    d. UPDATE a todo item
    e. DELETE a todo item
4. Create & Share a working Postman Collection which can be used to test all the APIs mentioned in Point 3. Above.
5. Enable support for [Basic Authentication](#) for all the APIs.


**You may use the following to complete the assignment:**

*Backend:*
1. Python 3.8+
2. Django 3.1+
3. Django Rest Framework 3.1+

*Testing:*
1. Postman Collection


**Note:**
1. You can assume things that are not defined with a reasonable value.
2. Add validation checks anywhere there is a possibility of an obvious violation of general logic. Example: 'Due Date' field value cannot be before 'Timestamp created' field value
3. Use PyCharm IDE for development. It will help you find basic issues quickly.
4. Do not use any third-party python package for implementation of tags (ex: taggit should not be used). You should implement all the models using pure Django only.


**Deliverables:**
1. A working Django App that accommodates all the requirements.
Please upload your code on a **private** GitHub repo and share it with the following GitHub ids: **algobulls-dev**, **shubham-das-algobulls**
    i. The repo should contain:
        1. Working Django code
        2. *requirements.txt:* The dependent Python packages must be captured in this file such that all dependencies can be installed using *pip* inside a [*virtualenv*](#).

ii. Make sure your code is cleaned up as per PEP8 standards, before you do the final submission. It is ok to keep pushing any number of intermediate code commits.

iii. Add sufficient comments for complex logic, before you do the final submission.

iv. Include a README that includes basic documentation on how to run the code.

2. Host the Django App over any cloud service provider <u>& share the link</u> over email (details below).
   If you are not aware of any, you can use https://www.pythonanywhere.com/

3. Share Django Admin superuser credentials for the Django App link shared in *Deliverables, Point 2* above over the email (details below).

4. Share the link to a working Postman Collection over the email (details below).
   a. [Example of how a shared Postman Collection will look like.](#)
   b. We expect 5 requests in this Postman Collection, corresponding to the APIs asked for in *Description, Point 3*.
   c. We should be able to fire APIs from here which directly interact with your hosted app in *Deliverables, Point 2* above.

Once all the above points are done, please send a mail to both developers@algobulls.com, mentioning your name and GitHub ID, requesting a review. Attach this coding assignment pdf to the mail as well.

**Duration:**
The ideal time is 3 days. If you need extra time, please request the same with appropriate reasoning.

**Asking for clarification/hints:**
Please send an email to developers@algobulls.com & shubham.das@algobulls.com with your query and we will get back to you.