AlgoBulls Backend Developer (Web) Coding Assignment

Nov 2023

As a part of this assignment, the candidate is expected to implement a small project with the details below.

Project Requirements:

1. Design a simple To-Do List App using Django & deploy it on the cloud
2. Write Unit & Integration tests with 100% coverage
3. Use GitHub Actions for basic CI/CD operations
4. Generate documentation for the entire code and host it as a static site

Detailed requirements are given below

Description: You need to design the 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 the status of a task
            i. Should be one of these values.
                1. OPEN (Default value)
                2. WORKING
                3. PENDING REVIEW
                4. COMPLETED
                5. OVERDUE
                6. CANCELLED
            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). Use class-based views:
    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 and 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.
6. You need to write the following tests for your codebase. (Follow the Django standard docs for writing your tests. Do not install any custom package for unit & integration testing. For E2E testing, you can use selenium.)
    a. Unit Tests - 100% code coverage required
    b. Integration Tests - 100% code coverage required
    c. E2E Tests - for the following scenarios:
        i. Create a todo item
        ii. View all todo items
        iii. Update a todo item
        iv. Delete a todo item

CI/CD:

Implement the below-mentioned GitHub actions for your GitHub repo:

1. Run tests automatically:
    a. Unit tests
    b. Integration tests
    c. E2E tests
2. Run linting tools:
    a. Flake8
    b. Black

Note:
1. Ensure that all the actions are triggered on every commit push.
2. Ensure that all the GitHub actions are passed before submitting your assignment for review.

Technology Stack you may use to complete the assignment:

Backend:
1. Python 3.11+
2. Django 4.2.7+
3. Django Rest Framework 3.14.0+

Test your codebase:
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 to implement **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*.
     3. Coverage report (HTML folder as a zip, screenshot of the main report). Ensure unit test coverage report is 100% and integration test report is 100%. Both reports should be separate
     4. A folder named “videos” with video recordings of all 4 E2E tests mentioned above.
   - 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.
    v. Include complete coverage reports in the repo.
    vi. Add a screenshot of the coverage summary in the README section.
2. Host the Django App over any cloud service provider & share the link over email (details below) & in the chatbox.
   If you are not aware of any, you can use https://www.pythonanywhere.com/
3. Host the documentation site over any cloud service provider for static content and share the link over email & in the chat box. The documentation should use proper English with no spelling or grammatical errors.
4. Share Django Admin superuser credentials for the Django App link shared in *Deliverables, Point 2* above, over the email (details below) & in the chat box.
5. Share the link to a working Postman Collection over the email (details below) & in the chat box.
    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.
    d. Make sure your Postman Collection is private and only shared with the following email IDs: shubham.das@algobulls.com, developers@algobulls.com

Once all the above points are done, please ping the Internshala chat box with all the links, credentials and other details. Also, please send an email to both developers@algobulls.com and shubham.das@algobulls.com mentioning your name and GitHub ID requesting a review. Attach this coding assignment PDF to the email 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:
You can ask on the Internshala chat box for any queries. The response may be delayed as we may have a lot of applications, and it may take time to respond to all of them.