The Boulevard Consulting Group, LLC
2345 Crystal Drive, Suite 205
Arlington, VA 22202

Problem Statement:

You are required to build a RESTful API that allows users to create, read, update, and delete articles. An article should have a title, content, and a timestamp of when it was created. The API should also allow users to retrieve a list of all articles, as well as a single article by its ID.

Requirements:

- Use Django to build the API
- Use Django REST framework to design and implement the API endpoints
- Database backend choice is up to you (e.g., PostgreSQL, SQLite)
- Write unit tests to ensure the correctness of the API to cover 80% of code

API End Points

1. GET /api/articles/: Returns a list of all articles in the database. Each article should contain the title, content, and created timestamp. Implement pagination so that only 10 articles are returned per page.
2. GET /api/articles/:id/: Returns a single article by its ID. The response should contain the title, content, and created timestamp of the article.
3. POST /api/articles/: Creates a new article. The request body should contain the title and content of the article. Return the ID, title, content, and created timestamp of the newly created article.
4. PUT /api/articles/:id/: Updates an existing article by its ID. The request body should contain the title and content of the updated article. Return the ID, title, content, and created timestamp of the updated article.
5. DELETE /api/articles/:id/: Deletes an existing article by its ID.

Requirements Details:

- Use Django to build the API. You can use any version of Django.
- Use Django REST framework to design and implement the API endpoints. You can use any version of Django REST framework.
- Use Django's built-in ORM to define the Article model with the title, content, and created timestamp fields.
- Write unit tests to ensure the correctness of the API. Use Django's built-in test framework to write unit tests for each API endpoint. Test that each endpoint returns the correct response and handles errors gracefully.

Additional Requirements:

- Implement pagination for the list of articles endpoint. Use Django REST framework's pagination features to paginate the list of articles.
- Add authentication and authorization to the API using token-based authentication. Implement token-based authentication using Django REST framework's authentication and permission classes.
- Use Docker to containerize the application. Use Docker to create a container for the application and provide instructions for running the application in the container.

The Boulevard Consulting Group, LLC
2345 Crystal Drive, Suite 205
Arlington, VA 22202

Evaluation Criteria:

- Code quality and organization
- Correctness and completeness of the API implementation
- Clarity and readability of the code
- Use of best practices for Django and RESTful API design
- Attention to detail in implementing the additional requirements

Duration and Submission:

Please submit your coding challenge solution prior to your scheduled technical interview. When finished, please your code via email (.zip file) to aga.brown@boulevardcg.com.