PYTHON TECHNICAL TASK

Develop a simple RESTful API application for managing a task list (ToDo list) based on the chosen web framework: Django (DRF), Flask, FastAPI, or aiohttp.

Requirements:

1. Create a User model with the following fields:
   - First name (first_name) - string, required field;
   - Last name (last_name) - text, optional field;
   - Username (username) - string, required field, unique.
   - Password (password) - string, required field, min 6 symbols.

2. Create a Task model with the following fields:
   - Title (title) - string, required field;
   - Description (description) - text, optional field;
   - Status (status) - selection from predefined values, for example: "New", "In Progress", "Completed".
   - User id (user_id) - int, Foreign key on user table (cascade delete).

3. Implement CRUD (Create, Read, Update, Delete) operations for tasks:
   - Get a list of all tasks;
   - Get a list of all user's tasks;
   - Get information about a specific task;
   - Create a new task;
   - Update task information (can be updated only by owner);
   - Delete a task (can be deleted only by the owner).

4. Implement API endpoints for the following actions:
   - Marking a task as completed;
   - Filtering tasks by status.

5. Add user authentication and authorization using JWT. (Only authorized users have access to the service)

6. Implement pagination for the task list.

7. Write unit tests for the API endpoints.

8. Use PostgreSQL as the database to store data. Create the corresponding database configuration in the chosen web framework.

9. Create a README.md file in the project repository with project documentation, including installation instructions, API documentation, and any other relevant information.

10. Create a Git repository for the project on GitHub and regularly commit and push changes.

Expected Outcome:

The completed task should include a fully functional API application based on the chosen web framework (Django REST Framework, Flask, FastAPI, or aiohttp). All endpoints should be tested and functioning correctly.

The project repository on GitHub should contain the source code, including the README.md file with project documentation. Additionally, the application could be Dockerized, allowing it to be easily run in a Docker container.

Good luck!