TECHNICAL TASK

Create a simple API to list, add, update, remove car number plates with their owner names.

USER STORY

User can register/store car number plates, their owner names and car model to a web endpoint through a Django REST API. The REST application should asynchronously retrieve car model image using Celery Framework, store it locally and display later on.

REQUIREMENTS

- The application does not have to require any authentication or authorization.
- The user should be able to do all CRUD operations on car numbers stored in database.
- Car plate format should be based on your home country's regulations.
- The API must be implemented using Django REST framework.
- Asynchronous car model image retrieval needs to be implemented using Celery Framework.
- Django and Celery should communicate through a RabbitMQ broker.
- Use version control (git) for your project and publish it to any public version control vendors – GitHub, BitBucket, GitLab or any other you prefer.
- Put a small README.md file which shows how to launch the project in local environment.
- You can pick any database that Django supports.

NICE TO HAVE

- Provide any reasonable amount of unit / smoke tests for your code
- Create a Django admin view to access the data
- Implement sufficient input data validations
- Database and RabbitMQ containers provided through docker compose file to easily test portability of the solution.