CleverFarm Backend Homework

Create RESTful web application which would expose basic CRUD methods for management of Farm and their Fields. Each Farm can have 0..N Fields.

Every Farm should have a unique ID, a mandatory name and an optional note.

Every Field should have a unique ID, a mandatory name, and Geometry denoting borders.

A Field should be able to compute it's area from geometry.

Try implementing some type of validation on the API (it's up to you how many):

- a field must have a geometry with area > 0
- a field geometry must be a Polygon and not a LineString
- a field should not overlap another field
- it shouldn't be possible to change field id
- fields should be located in the Czech Republic, or any other country
- ...

For working with geometries, you can use java **Geotools** library.

Application should make use of **Maven, Spring Boot, Spring MVC**. You should use **Kotlin**.

Use **JOOQ** library and arbitrary **AWS Postgres RDS managed database** (Free AWS Cloud plan should be sufficient) **or your local Postgres Database, with Postgis extension installed (for working with geometries)** to make the stored Farms and Fields persistent across application restarts.

Do not be afraid to use existing libraries if useful (e.g https://github.com/dmitry-zhuravlev/jooq-postgis-spatial for postgis x jooq interoperability etc.)

SQL scripts for DB initialization should be included in application. Ideally application should be able to initialize it's database after start.

Take care to follow best practices regarding OO design, RESTful API, SQL.

Externalize application configuration (DB connection etc.) to separate YAML or properties file, so that it is possible to run it locally, or in a different environment.

Create a free account on **www.gitlab.com**, create a repository for this project and version your progress using **git**.

Optionally, think about ways, how would you improve this REST API or application code. You can implement some (or not 🙂).

Some ideas for this are:

- API authentication
- paging and sorting
- versioning the DB schema
- auto documenting your API using Swagger (or another similar tool)
- ...

Please use a initial commit and then implement all of the functionality into another branch. Create a merge request to your master branch after you are done implementing.

And don't forget to write a **README** file consisting of steps necessary to run this project locally.