Take Home Assignment Fullstack Web Developer

Swisscom

Your task is to build a Web application that allows users to manage feature toggles for their applications. Other applications can then query this service to get active features for customers.

The main idea is that someone with the role product manager can control the rollout of features to customers while developers continually work on new features that might not be in a finished state.

A user should be able to perform the following tasks using a browser:

- Create a feature toggle
- Update a feature toggle
- Archive a feature toggle

Entity "FeatureToggle":

- `displayName?: string`
- `technicalName: string`
- `expiresOn?: datetime`
- `description?: string`
- `inverted: bool`
- `customerIds: string[]`

The service should additionally provide a REST API which allows to query what features are ON or OFF given a customer. This API will be invoked by other applications that enable or disable features based on the response.

Instructions

- Please do not take more than a day to work on the project. It is fine if it is not complete or does not meet your standards. We will discuss the content of what has been submitted and not what has been accomplished given the time frame.
- You can use any technology to implement the web application's backend, frontend and persistence layer. However, please refrain from using exotic technologies, as we would like to be able to run the project on our own machines.
- Please submit your results at least 3 days before the evaluation interview. You can send us a link to the repository that we can clone or a link to download the files as a compressed file archive. It is not necessary to deploy the app somewhere.
- If you are not familiar with the concept of feature toggles, you are welcome to read this excellent article: https://martinfowler.com/articles/feature-toggles.html. However, we don't require you to read it.
- If something is not clear because the requirements are short, you can make your own assumptions. Please write down your assumptions.

Example API request:

```
POST /api/v1/features
{
  "featureRequest": {
    "customerId": "1234",
    "features": [
        {"name": "my-feature-a"},
        {"name": "my-feature-b"},
        {"name": "my-feature-c"},
        {"name": "my-feature-d"}
    }
  }
}
```

Example API response:

```
{
  "features": [
      // customer is in the list of the feature toggle:
      {"name": "my-feature-a", "active": true, "inverted": false, "expired": false},

      // customer is in the list of the feature toggle, but toggle is inverted:
      {"name": "my-feature-b", "active": false, "inverted": true, "expired": false},

      // customer is NOT in the list of the feature:
      {"name": "my-feature-c", "active": false, "inverted": false, "expired": false},

      // customer is NOT in the list of the feature, but feature toggle expired:
      {"name": "my-feature-d", "active": true, "inverted": false, "expired": true}
    ]
}
```