dibimbing.id

Backend Server Development

Mini Project

1. Subject: Mini Project of Backend Server Development
2. Types:
    - Hands-on
3. Topics covered: Node.js, API, DB

Scenario

- You are working in an e-commerce company and single handedly responsible with a **merchant service** that handles the catalog of products owned by merchants

Objective

- You need to build a **merchant service** from scratch according to the requirements

Context

- For the MVP, the requirements of **merchant service** are defined below
- The user is a merchant

Requirements

Milestone 1

- A merchant service should be written with Node.js and using DB as a persistent storage
- A merchant service will expose several APIs with these functions
    - A merchant could register itself/create an account in the merchant service
    - A merchant could remove its data/delete its account in the merchant service

  - A merchant could add products in the merchant service
  - A merchant could delete a product in the merchant service
  - A merchant could update a product in the merchant service
  - A merchant could get the list of its products from the merchant service
  - Merchant information that a merchant service needed is
    - id
    - password
    - name
    - address
    - join_date
    - phone_number
  - Product information that is needed is
    - id
    - name
    - quantity
    - Price
- Commit the working code to git

Milestone 2

- Add a /login api to authenticate user
- Authentication using Basic Auth
- Commit the working code to git

Milestone 3

- Add JWT authorization as access_token to access resources after user logged in
- JWT is passed in Auth header as Bearer token
- Commit the working code to git

Milestone 4

- Move JWT from Auth header as Bearer token to cookies based token
- Commit the working code to git

What and How To Submit

- Create a doc written with markdown (md) format that explains
  - A simple architecture diagram of the merchant service
  - An Entity Relational Diagram of the merchant service data model

    - List of API contract like this
        - Bonus: Document the API contract in Swagger/OpenAPI spec format
    - Name the doc as ARCHITECTURE.md and checked that in the repo
- A working code checked in in your github repo
- Send the github repo link to <TODO>