**Dear team leader;**
**This document has a description of the back-end assignment,**

**#Techniques Used:**

**- Engine:**

| | |
|---|---|
| 1- Node JS v16.19.1 ( Nest JS / TypeScript ) | |

**- DB Engine:**

| | |
|---|---|
| 1- Postgresql database | https://www.postgresql.org/docs/ |

**- ORM:**

| | |
|---|---|
| 1- Prisma ORM | https://www.prisma.io/docs |

**- API Documentation:**

| | |
|---|---|
| 1- Swagger API Documentation | https://swagger.io/docs/ |

**- Authentication Strategy:**

| | |
|---|---|
| 1- Passport-JWT | http://www.passportjs.org/packages/passport-jwt/ |
| 2- Passport-Local | https://www.passportjs.org/packages/passport-local/ |
| 3- Bcrypt ( Password encrypter ) | NPM |

**- Mail Sender:**

| | |
|---|---|
| 1- mailer | NPM |

**- Validation Strategy:**

| | |
|---|---|
| 1- class-transformer | NPM |
| 2- class-validator | NPM |

**- Environment File:**

| | |
|---|---|
| 1- dotenv | NPM |
| 2- dotenv-cli | NPM |

**- File Management:**

| | |
|---|---|
| 1- fs | NPM |

---------------------------------------------------------------------------------------------------------------------

**#First start instruction:**
**- npm I**
**- read “.env.example” file and create .env file and fill required information.**
**- npm run generate // to generate schema tables in ./nodemodule**
**- npm run migrate // to create db named in .env in database URL and create tables.**
**- npm start or npm run start:dev // for watch.**
**- after the run and connection to the database succeed, you have to log this information:**

```
Server Running on http://localhost:3000

 API Documentation For Dashboard Running on http://localhost:3000/api/v1/dashboard/api-docs

API Documentation For Client Running on http://localhost:3000/api/v1/client/api-docs
```

**this URLs generated from .env file information and refer of server URL and API documentation.**
**- http://localhost:3000/api/v1/dashboard/api-docs // for Admin Dashboard API documentation.**
**- http://localhost:3000/api/v1/client/api-docs // for User Client API documentation**

**#Header Keys:**
**- refresh-token // we need to send it in refresh token endpoint.**
**- Authorization // we need to send Bearer authentication token**

**#Client API Documentation:**

**Sign up User:**

**- /api/v1/client/auth/signup-email**

-- to signup user by email and send OTP code by verification mail and return user data.

**- /api/v1/client/auth/signup-phone-number**

-- to signup user by phone number and return user data and we can send OTP code by SMS Message but not included now.

**- /api/v1/client/auth/verify-user**

-- to verify user after send mail by signup endpoint and return access token.

**- /api/v1/client/auth/resend-confirm-otp**

-- to resend OTP code to user email if old OTP code expired and return message to done.

**Login User:**

**- /api/v1/client/auth/login**

-- to login user by email or phone number and return access token or user isn`t verified.

**- /api/v1/client/auth/logout**

-- to logout user and return only success 200

**- /api/v1/client/auth/refresh**

-- to create refresh token and new expired time and return access token.

**- /api/v1/client/auth/is_authenticated**

-- to make sure is user authenticated and return userId and email.

**User Password:**

**- /api/v1/client/auth/forget-password**

-- to send OTP code by to email and return message to done.

**- /api/v1/client/auth/reset-password**

-- to reset password after OTP code sent and return message to done.

**File Uploader:**

**- /api/v1/client/upload/upload-file**

-- to upload file by multipart/formdata and send uploadFileFor in query for store this uploaded file in specific folder and our folder key now is : PRODUCT_IMAGE
and return url and type and we need file URL to store it in create or update product in imageUrl field.

**- /api/v1/client/upload/delete-file**

-- to delete file and return success 200.

**User Management:**

**- /api/v1/client/user/ change-my-password**

-- to change my password if I authenticated and return message for done.

**- /api/v1/client/user/get-my-profile**

-- to get my profile information and return user information.

**- /api/v1/client/user/update-my-profile**

-- to update my profile information and return user information after update.

**Product:**

- **/api/v1/client/product/my-products**

-- to get list of products assigned to me by Admin with pagination.

**------------------------------------------------------------------------------------------------------------**

**#Admin Dashboard API Documentation:**

**Login Admin User:**
**- /api/v1/dashboard/auth/login**
-- to login user by email or phone number and return access token or user isn`t verified.
**- /api/v1/client/auth/logout**
-- to logout user and return only success 200
**- /api/v1/dashboard/auth/refresh**
-- to create refresh token and new expired time and return access token.
**- /api/v1/dashboard/auth/is_authenticated**
-- to make sure is user authenticated and return userId and email.

**Admin User Password:**
**- /api/v1/dashboard/auth/forget-password**
-- to send OTP code by to email and return message to done.
**- /api/v1/dashboard/auth/reset-password**
-- to reset password after OTP code sent and return message to done.

**File Uploader:**
**- /api/v1/dashboard/upload/upload-file**
-- to upload file by multipart/formdata and send uploadFileFor in query for store this uploaded file in specific folder and our folder key now is : PRODUCT_IMAGE and return url and type and we need file URL to store it in create or update product in imageUrl field.
**- /api/v1/dashboard/upload/delete-file**
-- to delete file and return success 200.

**User Management:**
**- /api/v1/dashboard/user/ change-my-password**
-- to change my password if I authenticated and return message for done.
**- /api/v1/dashboard/user/ change-user-password**
-- to change user password if I authenticated and return message for done.

**- /api/v1/dashboard/user/update-user-profile**
-- to update user profile information and return user information after update.

- **/api/v1/dashboard/user/users-list**
-- to get list of users with pagination.

**Product:**
- **/api/v1/dashboard/product** **Method type: POST**
-- to create new product and get imageUrl from file uploader endpoint after upload file.

- **/api/v1/dashboard/product/:id** **Method type: GET**
-- to get product information by id and return product data.

- **/api/v1/dashboard/product/:id** **Method type: PUT**
-- to update product information by id and return updated product data.

- **/api/v1/dashboard/product/:id** **Method type: DELETE**
    -- to delete product information by id and return message for done.

- **/api/v1/dashboard/product/list**
    -- to get list of products with pagination.

- **/api/v1/dashboard/product/user-products/:userId**
    -- to get list of products assigned to user with pagination.

- **/api/v1/dashboard/product/assign-products**
    -- to assign products to specific user and return user data with his assigned products.

--------------------------------------------------------------------------------------------------------------

**#File Storage:**
**-We store files on local folder “public” in project root and we can use AWS S3 Bucket.**