Day 2

1. Create Nestjs Project
2. Create a module (Users module)
3. Use mongoose to create the schema
   ```
   email:string (is unique)
   Password:string (min 8, alphanumeric)
   fullName:string
   age:number (min 16, max 60)
   mobileNumber (11 characters starts with 01)
   role admin or normal
   ```
4. Create the following endpoints
   a. /users/sign-up (public)
   b. /users/sign-in (public)
   c. /users/my-profile (protected) (returns the profile of the authenticated user)
   d. /users/all
      i. (return list of the other users, excluding the authenticated user)
      ii. (only for admin user)
5. Add proper validation for all the dtos
6. Add Swagger documentation for the endpoints
7. Use custom decorator to validate role `@Role(‘admin’)
8. **(BONUS)** Use exception filters to handle any exception
9. **(BONUS)** Create exception filter that handles all exceptions and return error in format of
   ```
   {
           status:number;
           reason:string,
           date:Date
   }
   ```

—---

Day 1

Create a Nestjs project

Create a controller to serve employees data (validate data using class-validator)

1. List employees
2. Get employee by id
3. Add employee
4. Update employee
5. Delete employee
6. Get highest paid employee

Use swagger for documenting the endpoints

Serve static data

```
{
        Id:number;
        Name: string;
        Age: string;
        Salary: number;
}
```