NESTJS

ENG. MOSTAFA MAHMOUD HELMY

📍 ITI-SMART VILLAGE

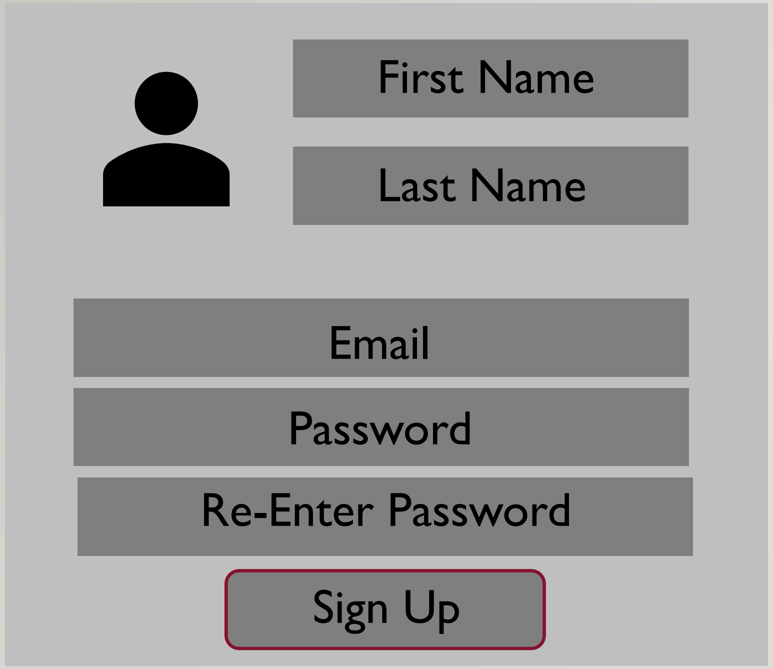
✉ P.MostafaMahmoud@gmail.com

DAY 3

- Authentication [login - registration]
  - Hash Password
  - Verify Pass
- Authorization [Permissions]
  - JWT (Json Web Token)
  - Guard

Authentication vs Authorization

Registration



First Name

Last Name

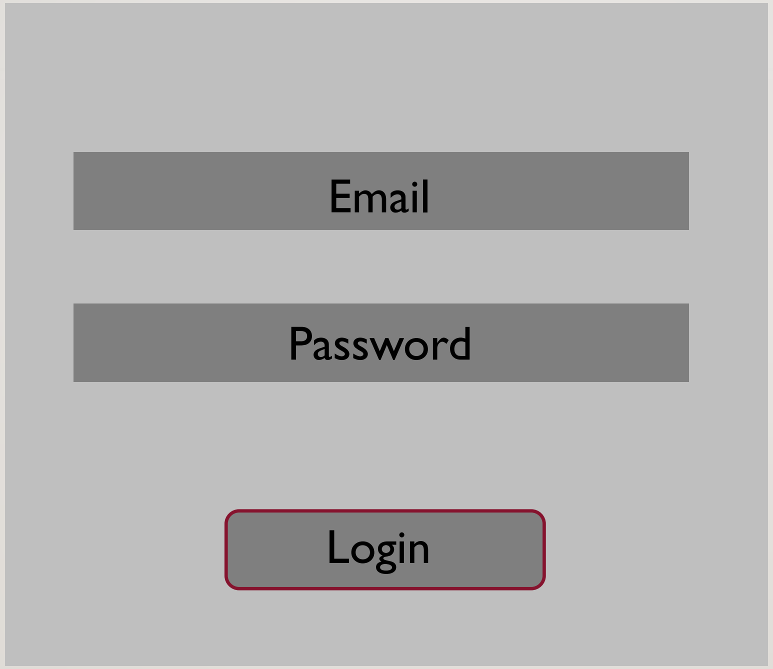
Email

Password

Re-Enter Password

Sign Up

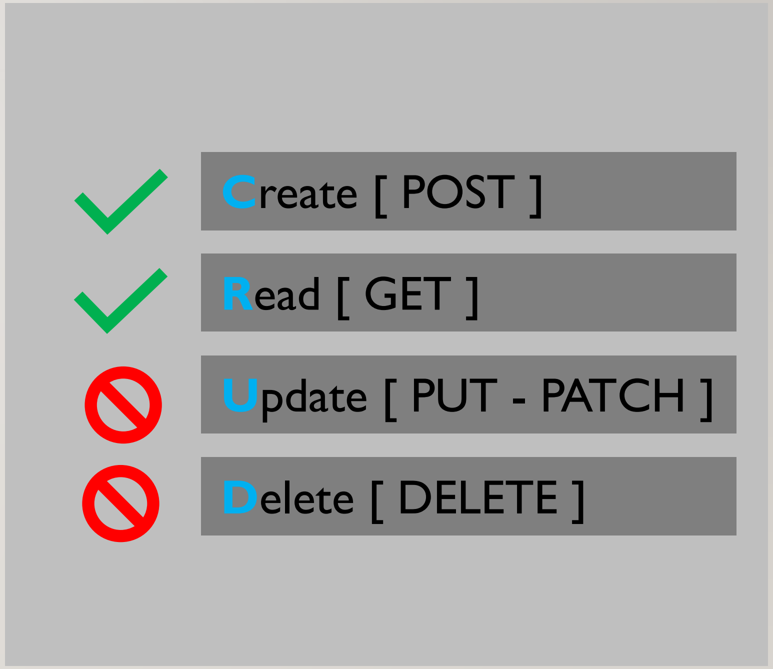
Login



Email

Password

Login



✓ Create [POST]

✓ Read [GET]

⊘ Update [PUT - PATCH]

⊘ Delete [DELETE]

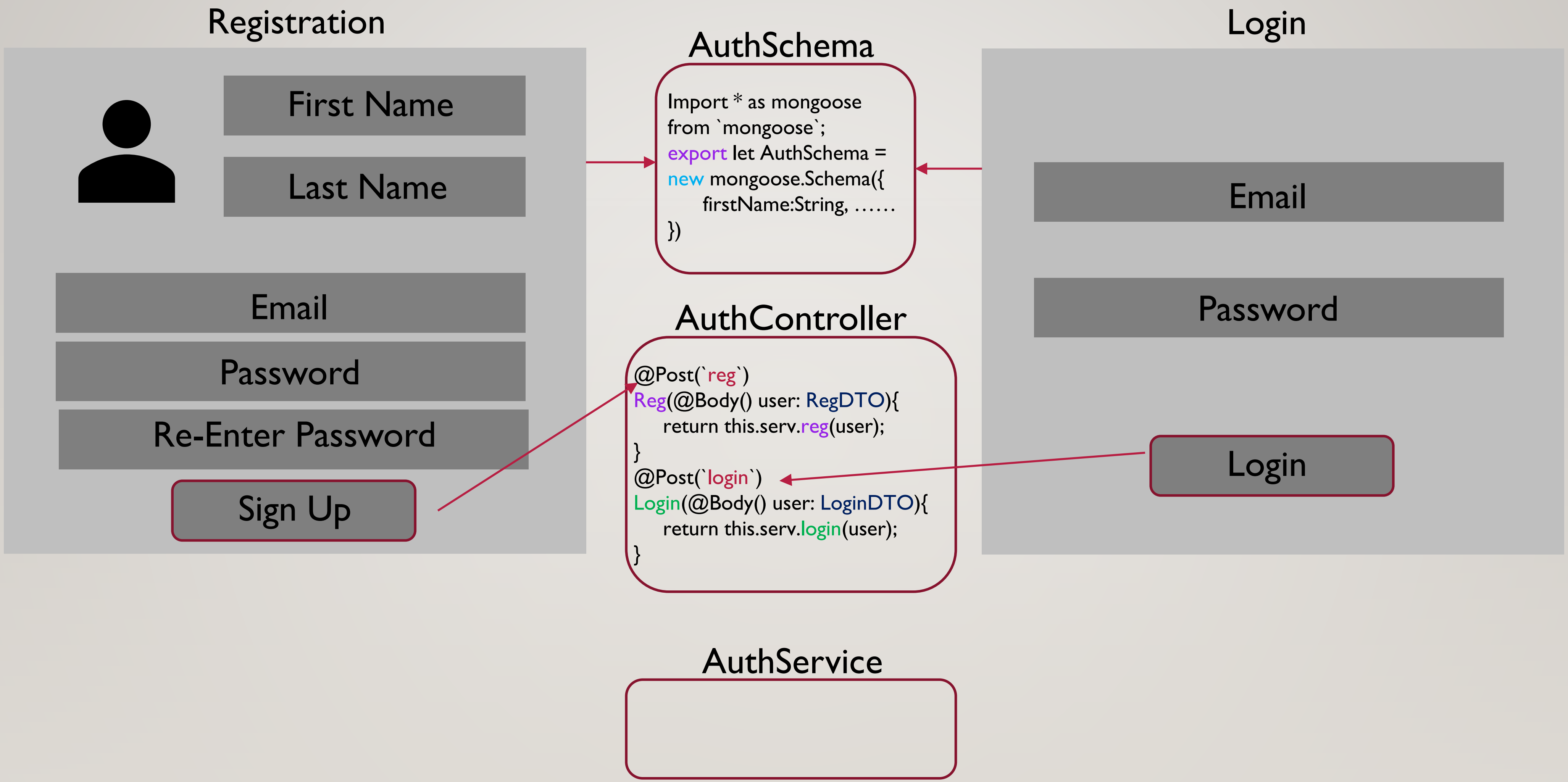
Authentication

Who you are?

Authorization

What can you do?

Authentication



AuthSchema

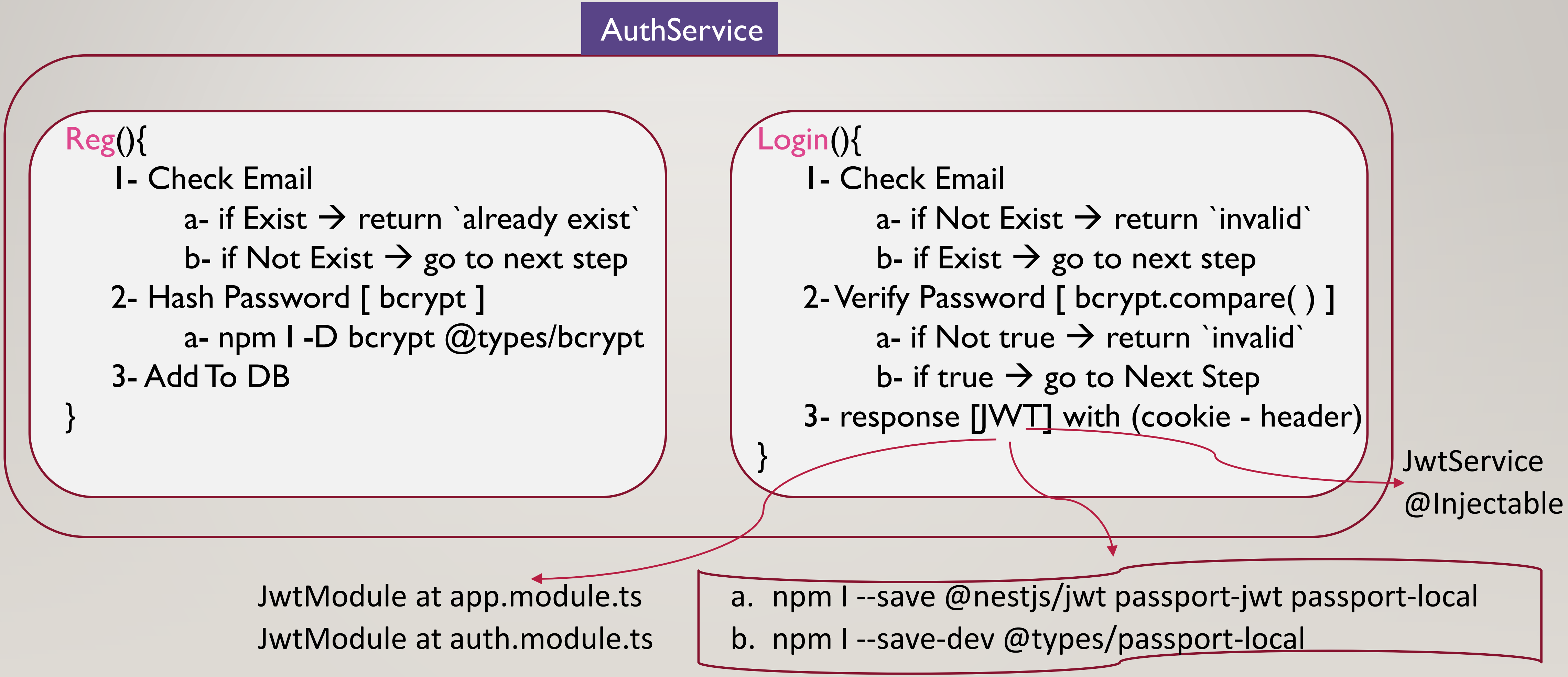
```
Import * as mongoose
from `mongoose`;
export let AuthSchema =
new mongoose.Schema({
      firstName:String, ……
})
```

AuthController

```
@Post(`reg`)
Reg(@Body() user: RegDTO){
    return this.serv.reg(user);
}
@Post(`login`)
Login(@Body() user: LoginDTO){
    return this.serv.login(user);
}
```

AuthService

Authentication



AuthService

```
Reg(){
    1- Check Email
        a- if Exist → return `already exist`
        b- if Not Exist → go to next step
    2- Hash Password [ bcrypt ]
        a- npm I -D bcrypt @types/bcrypt
    3- Add To DB
}
```

```
Login(){
    1- Check Email
        a- if Not Exist → return `invalid`
        b- if Exist → go to next step
    2- Verify Password [ bcrypt.compare( ) ]
        a- if Not true → return `invalid`
        b- if true → go to Next Step
    3- response [JWT] with (cookie - header)
}
```

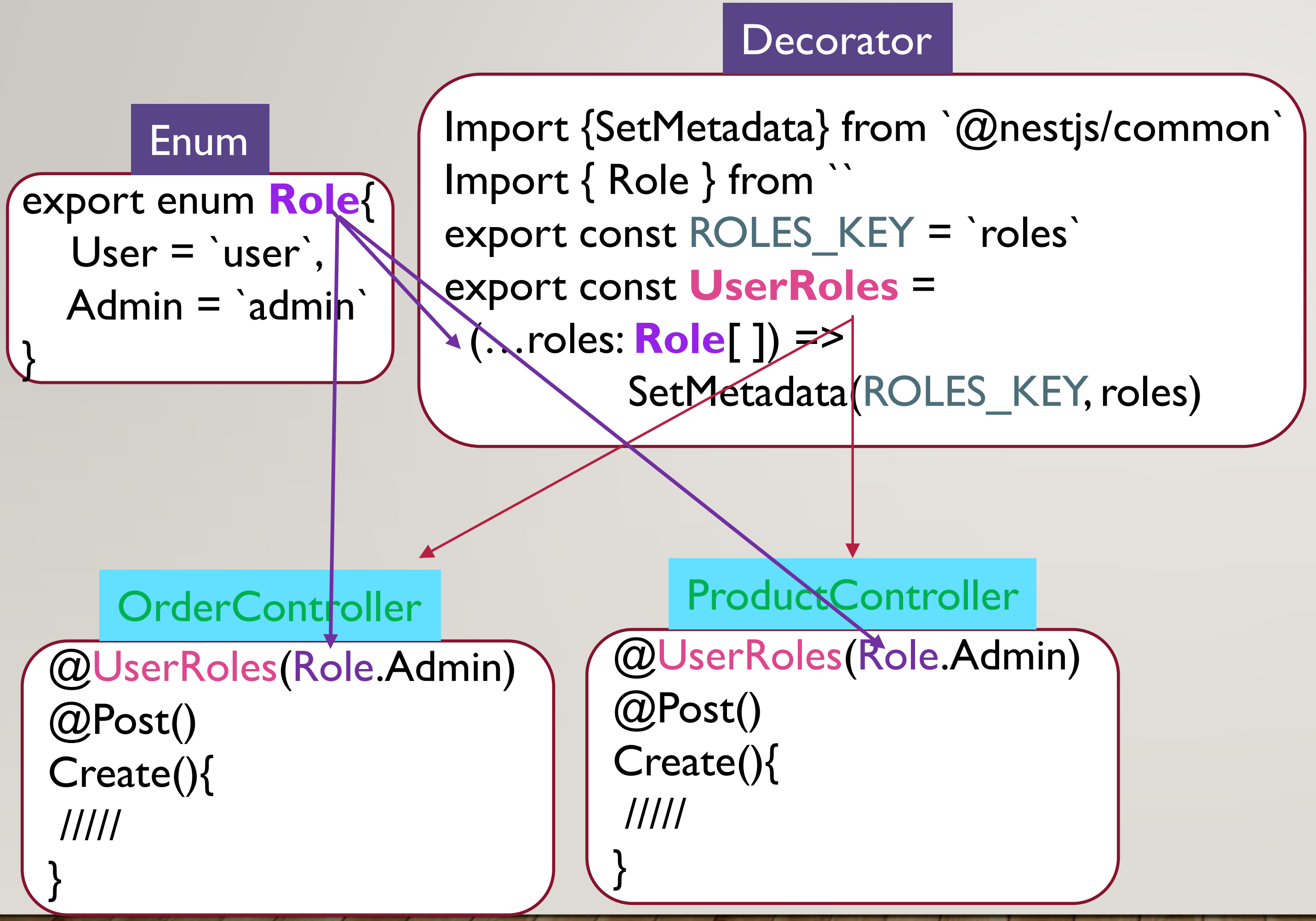
JwtService
@Injectable

JwtModule at app.module.ts
JwtModule at auth.module.ts

a. npm I --save @nestjs/jwt passport-jwt passport-local
b. npm I --save-dev @types/passport-local

Authorization

What can you do?



Enum

```
export enum Role{
    User = `user`,
    Admin = `admin`
}
```

Decorator

```
Import {SetMetadata} from `@nestjs/common`
Import { Role } from ``
export const ROLES_KEY = `roles`
export const UserRoles =
(...roles: Role[ ]) =>
            SetMetadata(ROLES_KEY, roles)
```

OrderController

```
@UserRoles(Role.Admin)
@Post()
Create(){
 /////
}
```

ProductController

```
@UserRoles(Role.Admin)
@Post()
Create(){
 /////
}
```

Guard

```
Ur Logic : /**/
class RolesGuard implements CanActivate{
      canActivate(){
         1- Reflector ➔ Roles ??
         2- context ➔ header – body – cookie …
         3- return true / false
      }
}
```

app.module.ts

```
Import {APPGUARD} from `@nestjs/core`
@Module({
   Providers: [
      AppService,
      {
         provide:APP_GURAD,
         useClass: RolesGuard
      }
   ]
})
```