

BARCODE SCANNER SYSTEM DOCUMENTATION

DOCUMENTS INCLUDES:

1. The Project Main Structure
2. Project Operation Methods
3. Dependencies and Tools
4. How to Run Project
5. Products
6. API

1. THE PROJECT MAIN STRUCTURE

```
├── config/
│   └── db.js          # Database connection
├── controllers/
│   ├── userController.js # User CRUD logic
│   └── productController.js # Product CRUD logic
├── middlewares/
│   └── auth.js         # Authentication & authorization middleware
├── models/
│   ├── User.js        # User model
│   └── Product.js     # Product model
├── routes/
│   ├── userRoutes.js  # User routes
│   └── productRoutes.js # Product routes
├── utils/
│   └── jwt.js         # JWT generation logic
├── app.js             # Entry point
└── package.json
```

2. PROJECT OPERATION METHODS

config/

In the folder config there is one file with name db.js -> in this file we set up connection with database and if the connections failed showing relevant error

controllers/

In this folder there are two files:

- userController.js -> in this file we do CRUD OPERATION with users
- productController.js -> in this file we do CRUD OPERATION with products

middlewares/

In this folder there are one file with name auth.js that verify the token in the header of request that has been generate in file with name utils

models/

There are two files in this folder with names: Product.js and User.js. These files define Schemas and save them in the database

routes/

There are two files with names: userRoutes.js and productRoutes.js that direct to the right operations by the routes names

utils/

In this folder there are only one file with name jwt.js that generates the token base on our JWT_SECRET that has been configure in the .env file

.env

In this file we define such things like ip, port, database url and passwords

app.js

This is our main file that start the project

3. DEPENDENCIES AND TOOLS

"bcryptjs": "^2.4.3"	"body-parser": "^1.20.3"	"cookieparser": "^0.1.0"
"cors": "^2.8.5"	"dotenv": "^16.4.5"	"express": "^4.21.1"
"jsonwebtoken": "^9.0.2"	"mongoose": "^8.7.2"	"morgan": "^1.10.0"
"nodemon": "^3.1.7"		

DATABASE:

MongoDB (NoSQL/Unrelational Database)

4. HOW TO RUN PROJECT

```
node app
```

OR

```
node app.js
```

5. PRODUCTS

Products functionality is same as Users!

6. API REQUESTS FOR PRODUCTS AND USERS

1. PRODUCTS:

POST	/api/products/	- Create product (Admin)
GET	/api/products/	- List all products (Public)
GET	/api/products/:id	- Get product details (Public)
PUT	/api/products/:id	- Update product (Admin)
DELETE	/api/products/:id	- Delete product (Admin)

2. USERS:

POST	/api/users/	- Create user (Protected)
POST	/api/users/login	- Login user (Public)
GET	/api/users/	- List all users (Admin)
GET	/api/users/:id	- Get user details (Admin)
PUT	/api/users/:id	- Update user (Admin)
DELETE	/api/users/:id	- Delete user (Admin)