Documentation for HomeJam-assignment and functionality explanation.

API:

The folder contains the backend-API based on REST and follows MVC architecture.

Inside server folder enter command npm i to install all the dependencies module.

Add .env file inside the server folder.

In the .env file enter the following environment variables

MONGO_URI = mongodb+srv://Dairing:Dairing@cluster0.lkhn8.mongodb.net/homejamDB?retryWrites=true&w=majority
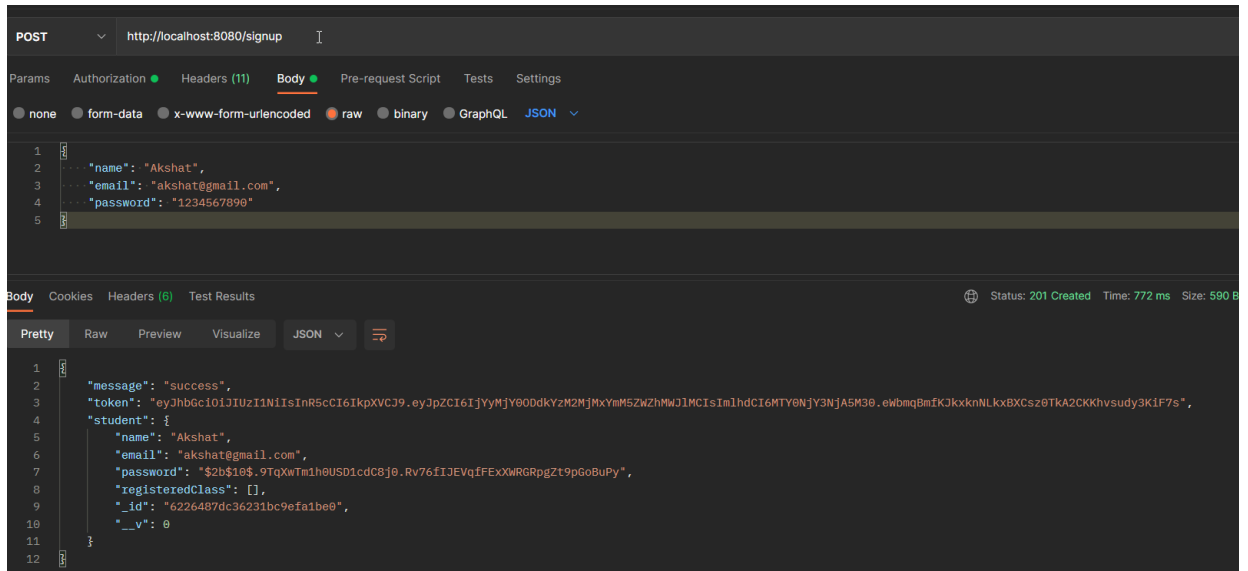
JWT_SECRET = homejam

I have used MongoDB Cloud to make free tier cluster and Database named HomeJam.

I have also used bcrypt for password encryption and JWT for token

Enter npm start to start backend server.

The server will be running on http://localhost:8080

(Use this for API Testing)

POST   http://localhost:8080/signup

Params   Authorization   Headers (11)   Body   Pre-request Script   Tests   Settings

none   form-data   x-www-form-urlencoded   raw   binary   GraphQL   JSON

```json
{
    "name": "Akshat",
    "email": "akshat@gmail.com",
    "password": "1234567890"
}
```

Body   Cookies   Headers (6)   Test Results   Status: 201 Created   Time: 772 ms   Size: 590 B

Pretty   Raw   Preview   Visualize   JSON

```json
{
    "message": "success",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyMjY0ODdkYzM2MjMxYmM5ZWZhMWJlMCIsImlhdCI6MTY0NjY3NjA5M30.eWbmqBmfKJkxknNLkxBXCsz0TkA2CKKhvsudy3KiF7s",
    "student": {
        "name": "Akshat",
        "email": "akshat@gmail.com",
        "password": "$2b$10$.9TqXwTm1h0USD1cdC8j0.Rv76fIJEVqfFExXWRGRpgZt9pGoBuPy",
        "registeredClass": [],
        "_id": "6226487dc36231bc9efa1be0",
        "__v": 0
    }
}
```

You can signup and login like this,
Similarly all routes are defined inside routes folder, you can use that as refrence.

Now for authentication middleware you have to copy the token and paste it in authorization header (Token with bearer) and id with logged in user Object ID in headers as key and value.

Headers   9 hidden

	KEY	VALUE
✓	id	62260cb7681f02ac29bb4186
	Key	Value

Type

Bearer Token ⌄

The authorization header will be automatically generated when you send the request. Learn more about authorization ↗

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJp
ZCI6IjYyMjYwY2I3NjgxZjAyYWMyOWJiNDE4
NiIsImIhdCI6MTY0NjY2MTYzNH0.aFxELRpZ
Zt20OjOu9Tj-ry-iUwsJ0SkxmkHHNgLkEQg

Comments are there to for easily running the api and all the required functionality are explained using comments.