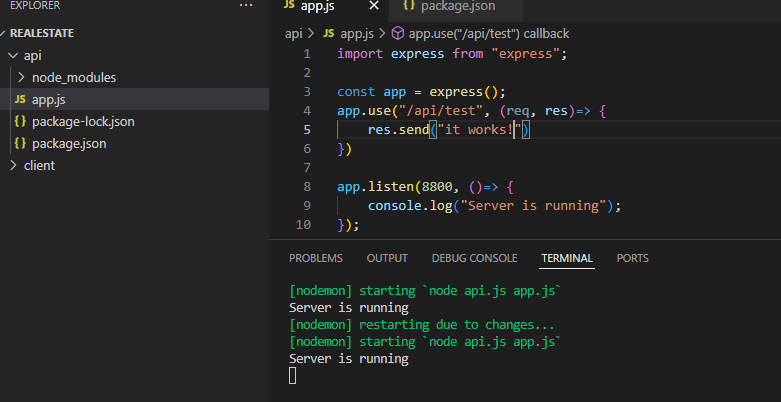
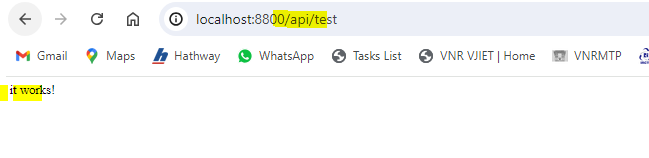
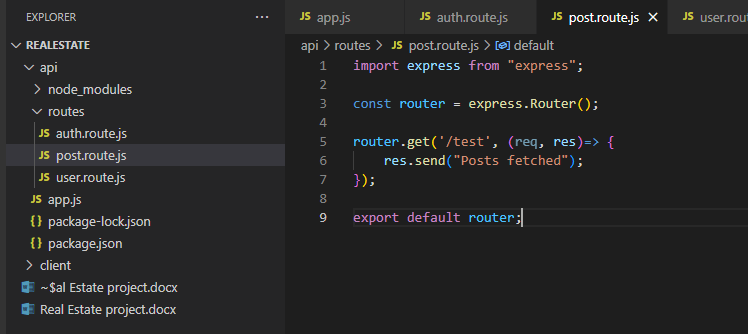
**Real Estate project**

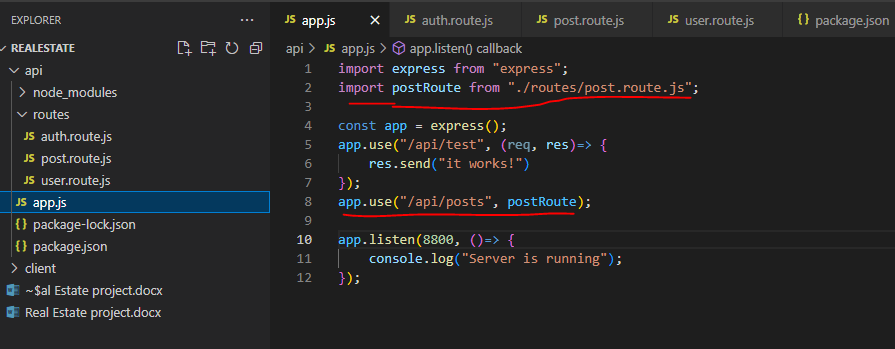
Create project folder and initiate app via express

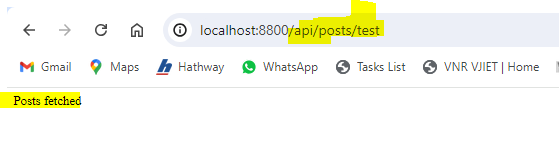


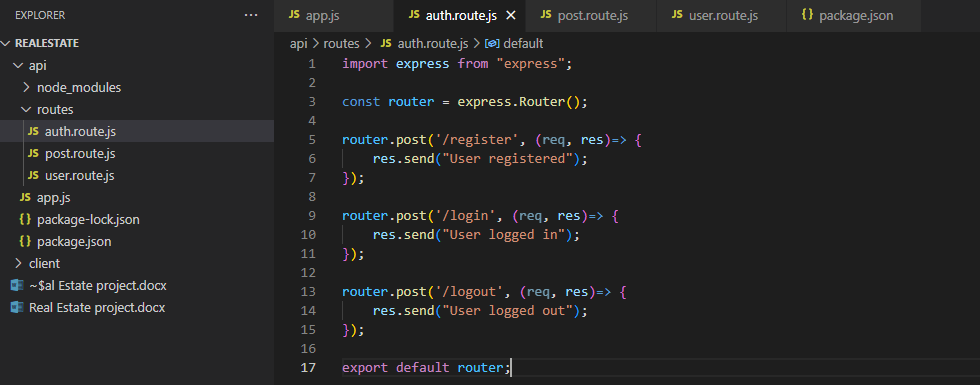


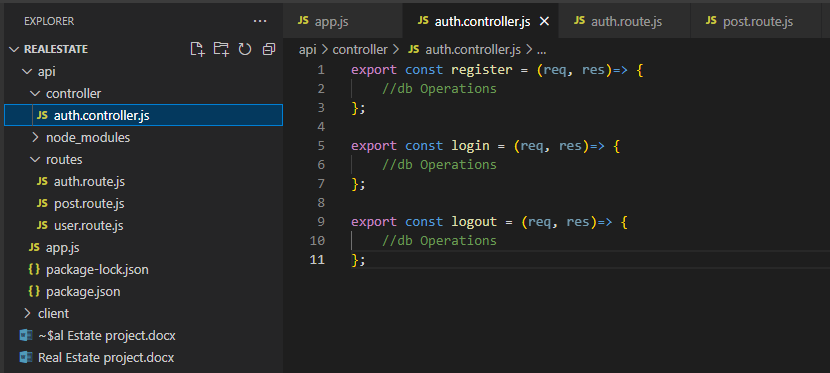
## **Test Routes**

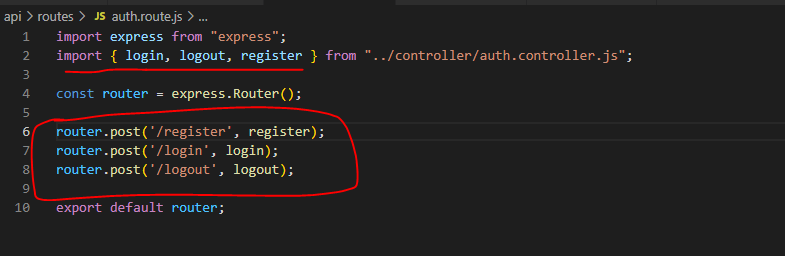




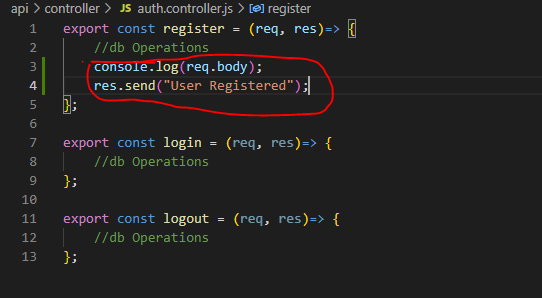


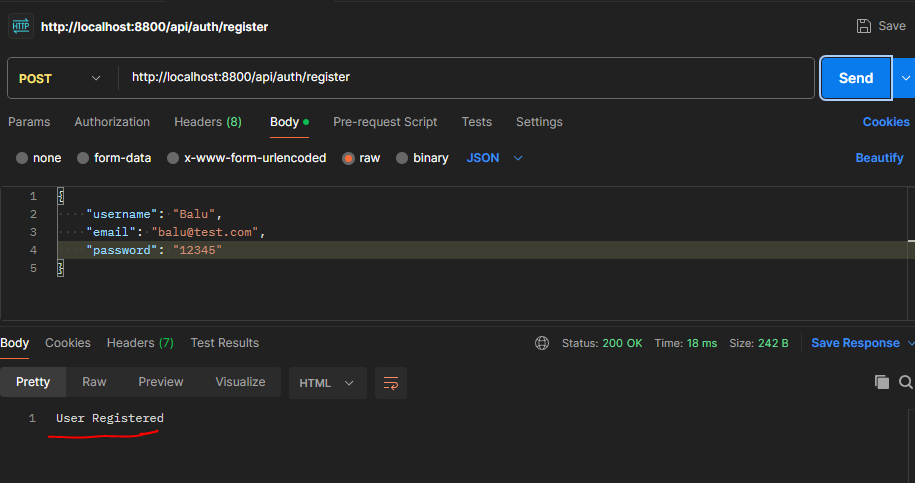


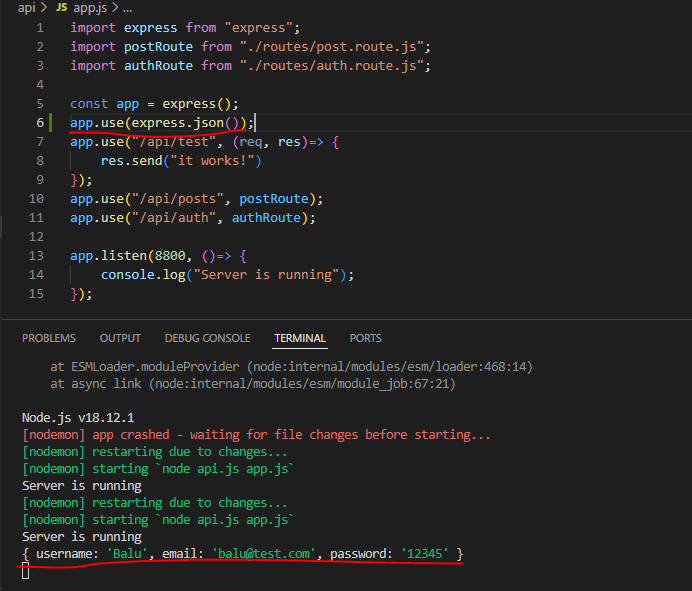




### **Register API test**

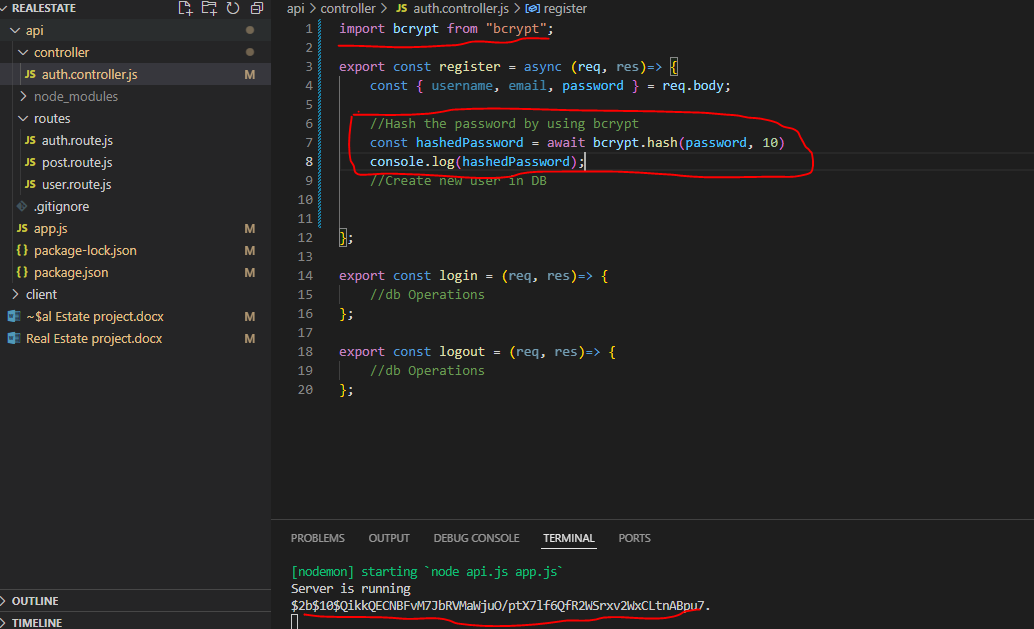






## **Bcrypt – Used for hashing passwords**

Hash the passwords using bcrypt => npm I bcrypt



## **Prisma**

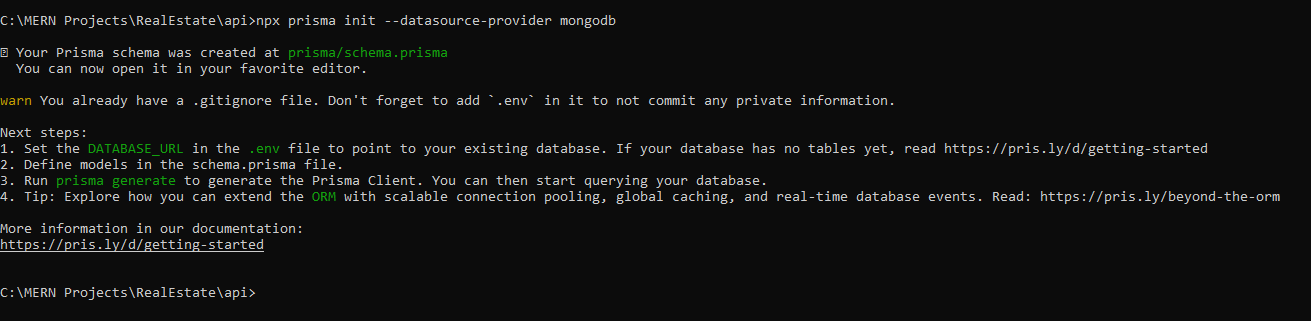
To store the user details to db we use prisma

Install => npm I prisma

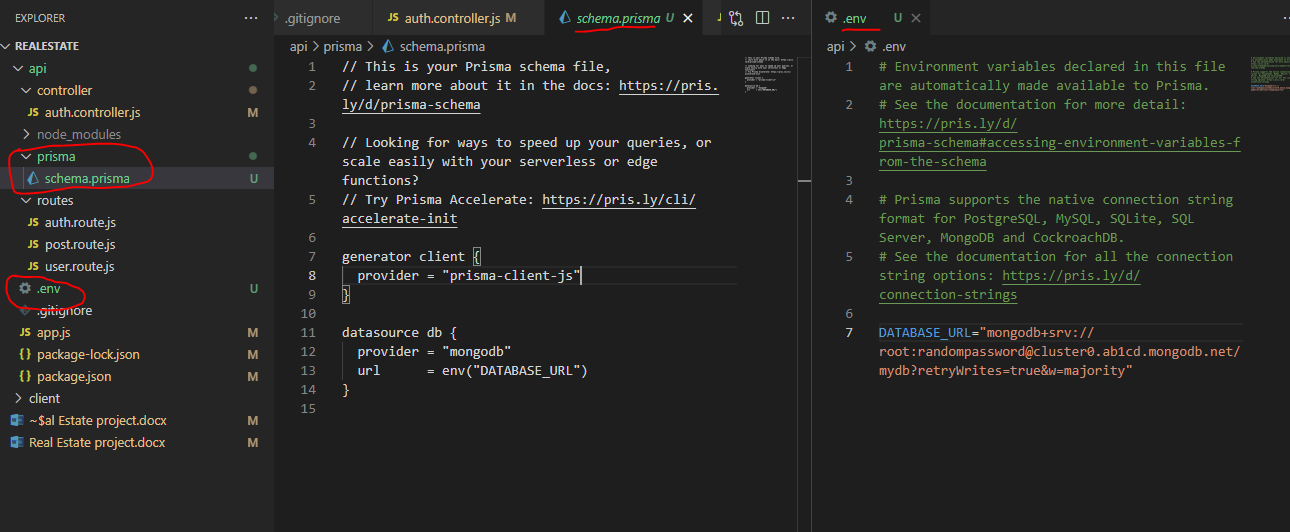
Follow <https://www.prisma.io/docs/getting-started/quickstart>

Now initialize prisma

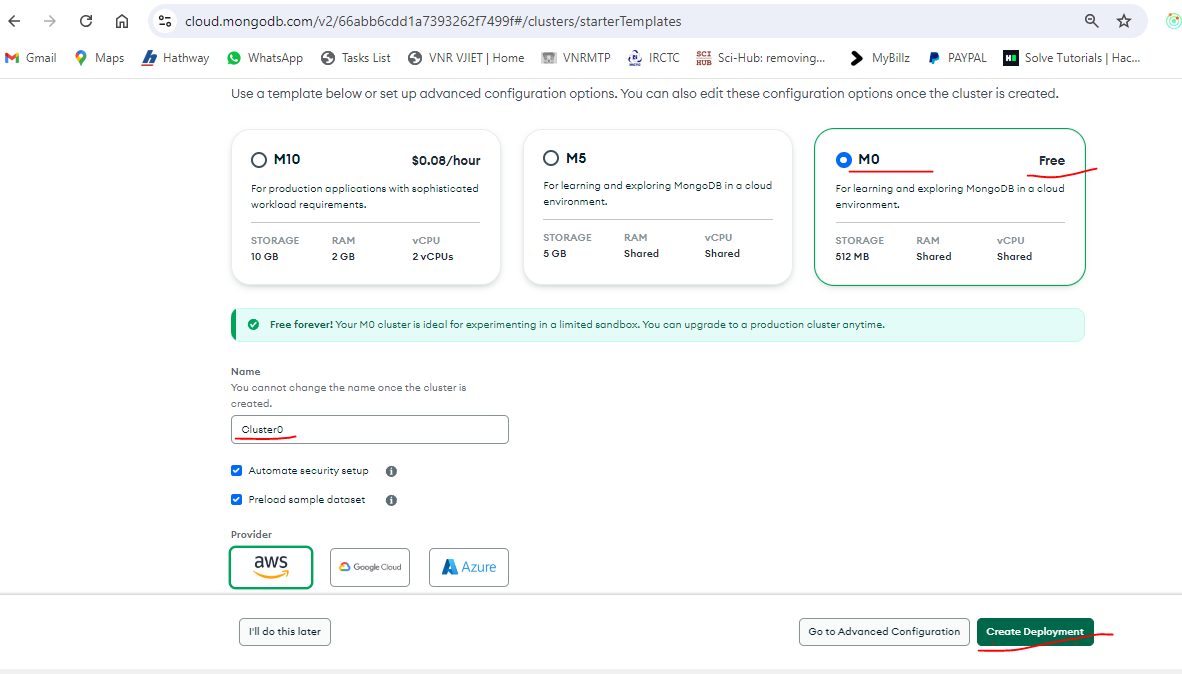
Run => npx prisma init --datasource-provider mongodb

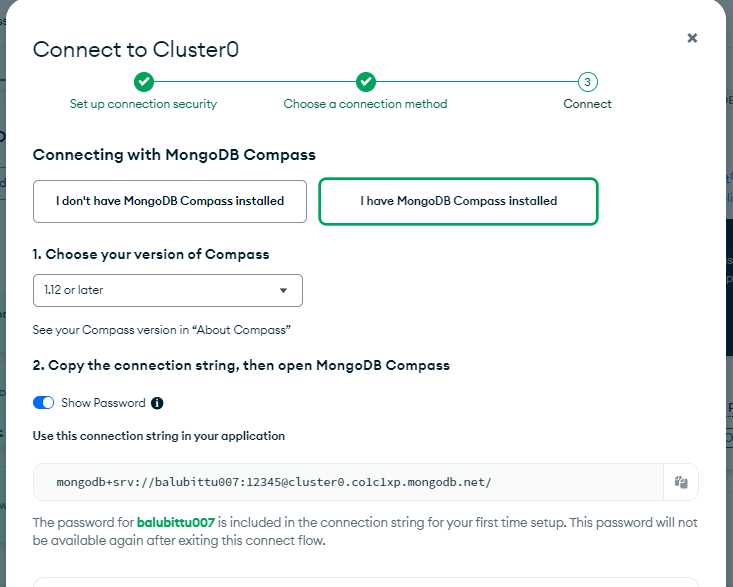


A new folder prisma is created also the env file is generated



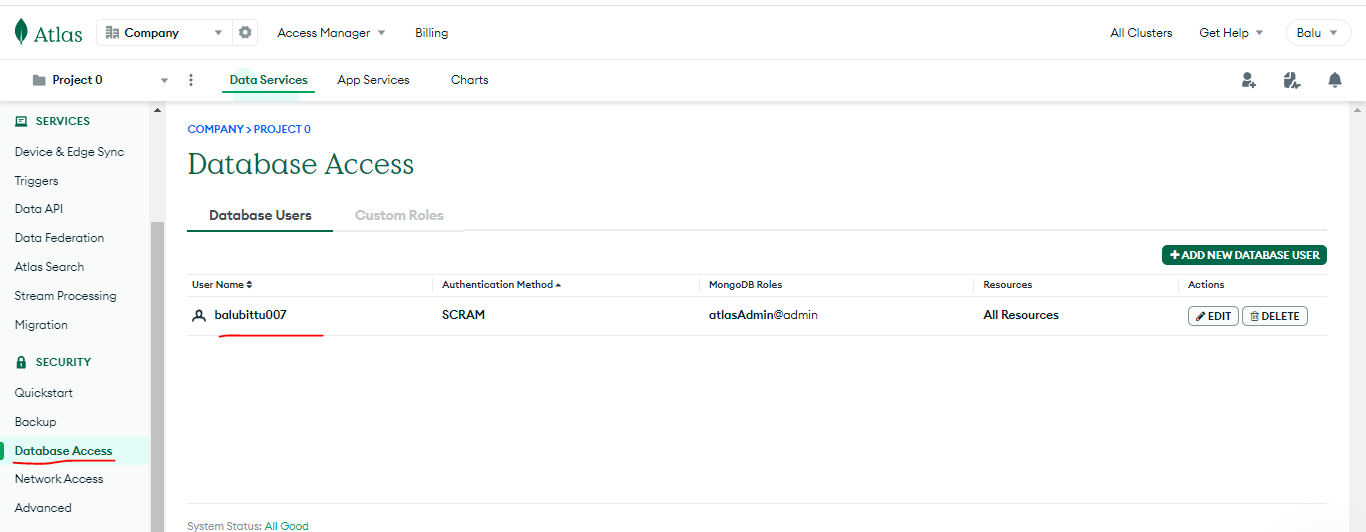
Create deployment in atlas



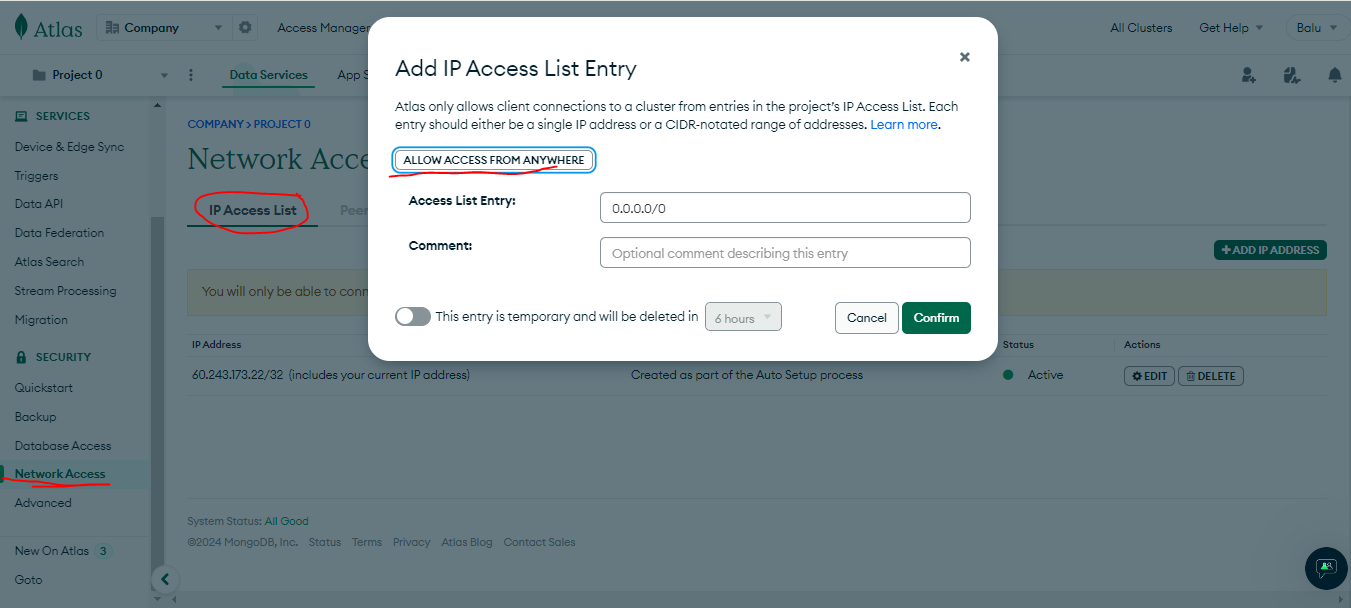


mongodb+srv://balubittu007:12345@cluster0.co1c1xp.mongodb.net/

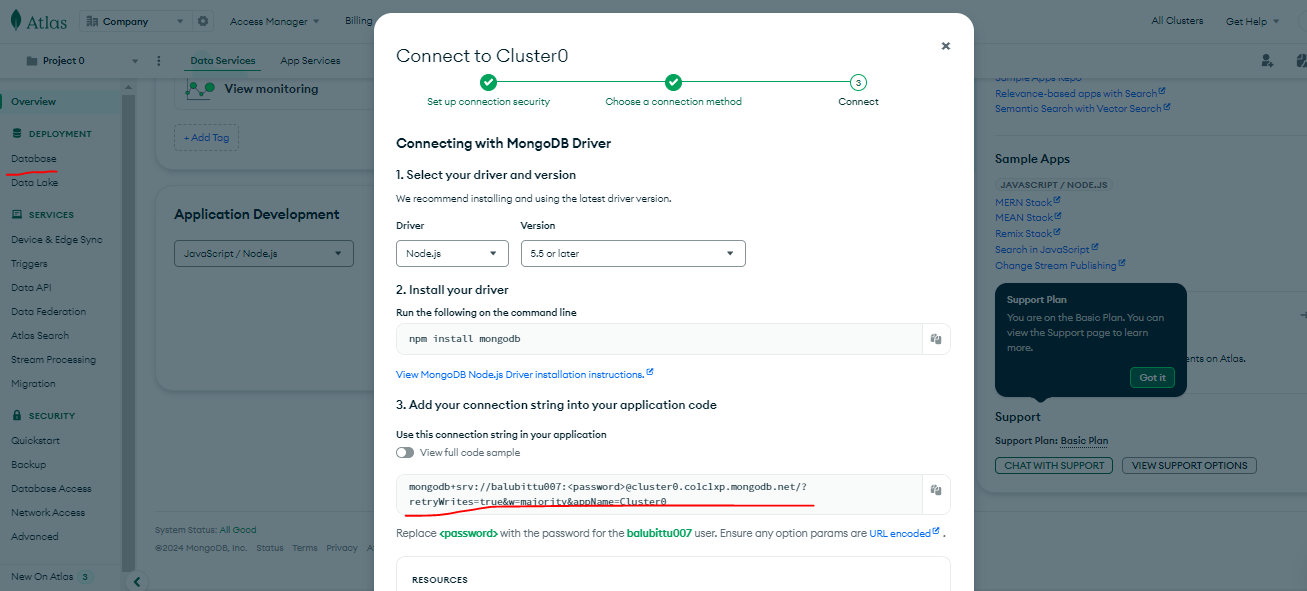
We can see the users in Database access

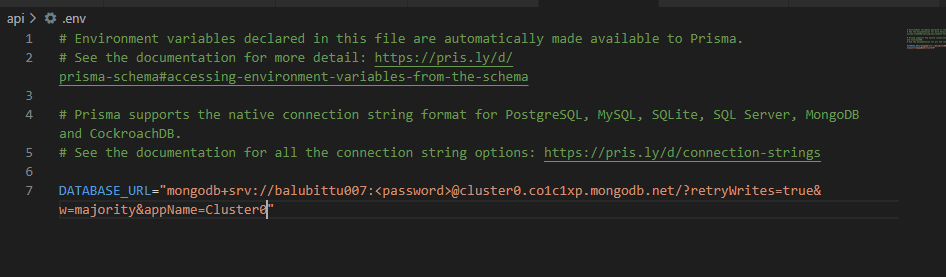


After deploying the app we need to enter server ip

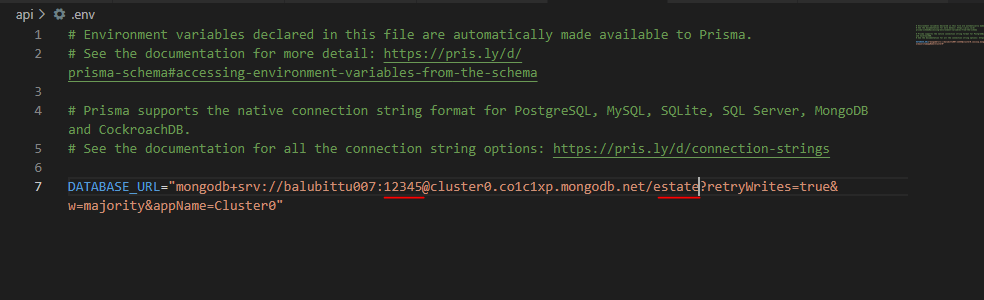


Get connection string from here



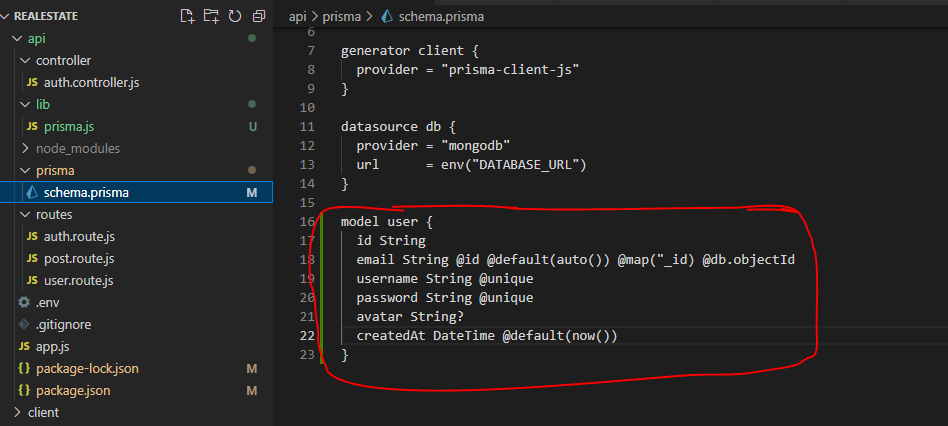


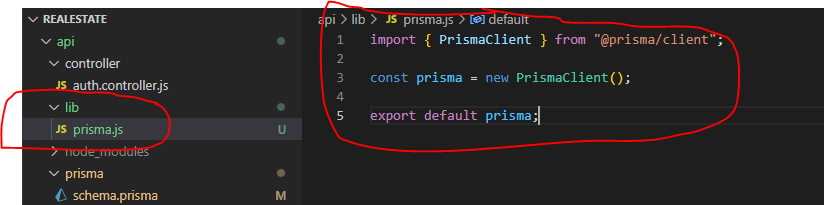
Update password and collection name in the url



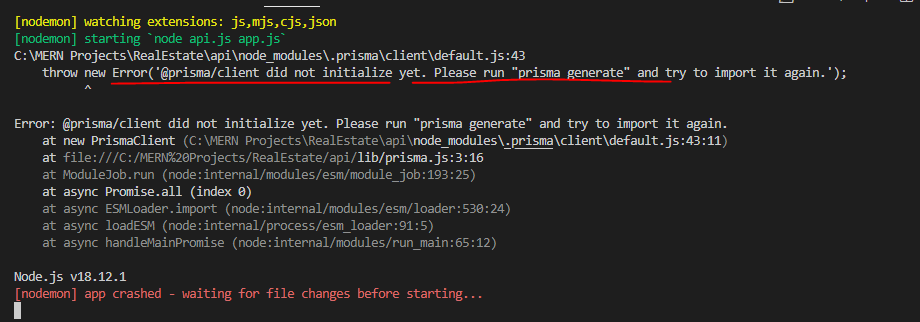
### **Creating user model:**

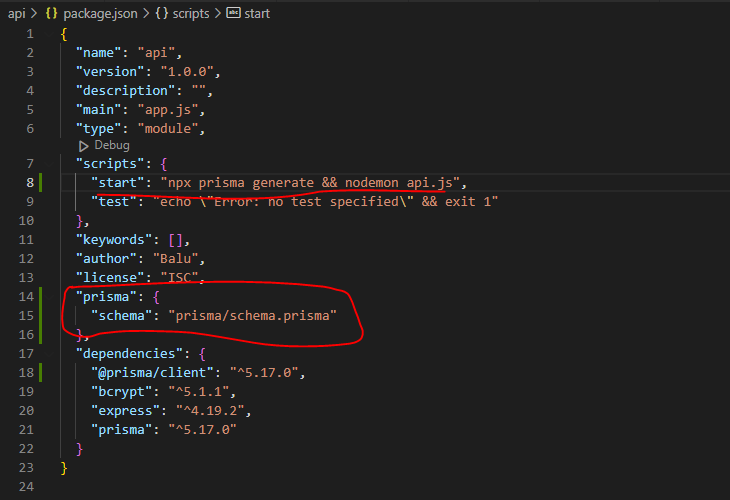
Run npm I @prisma/client





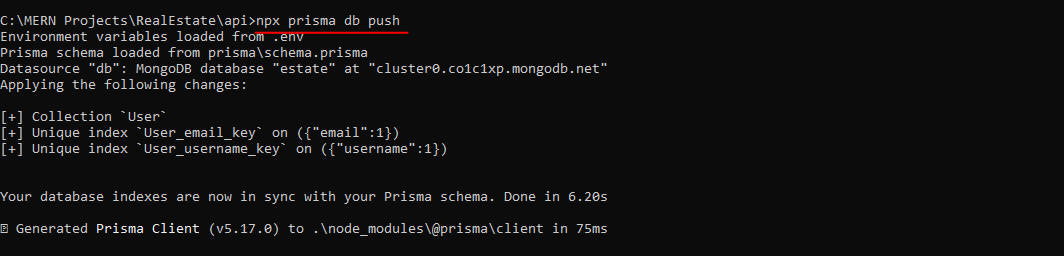
On running the server got the error below

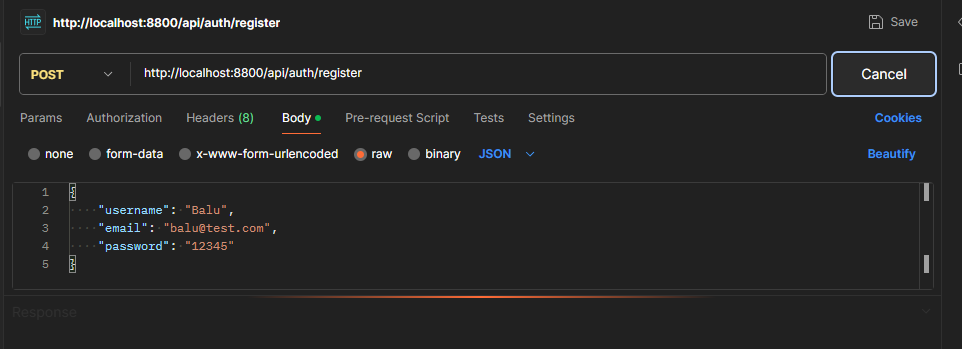


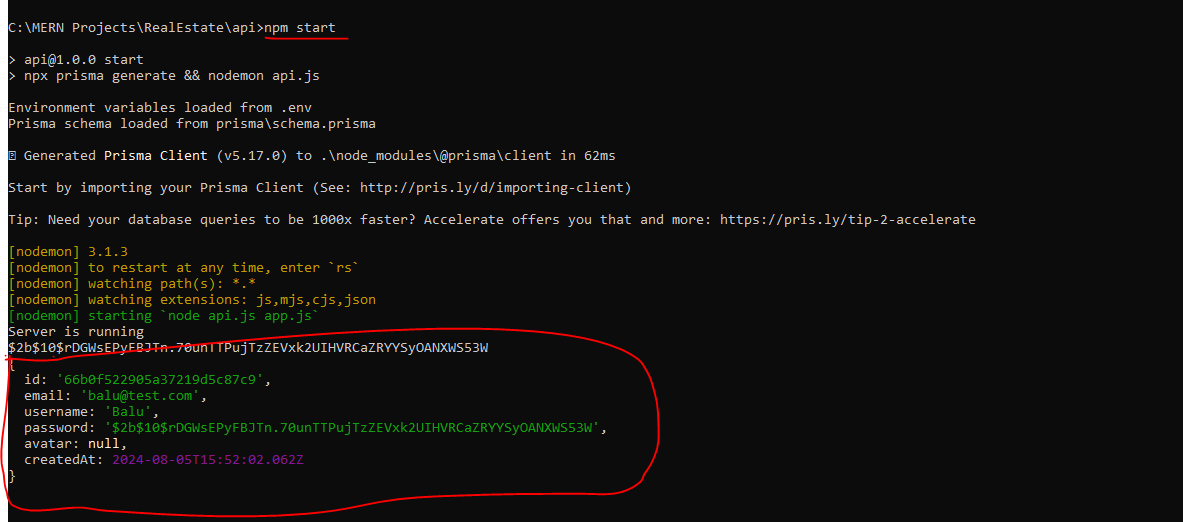




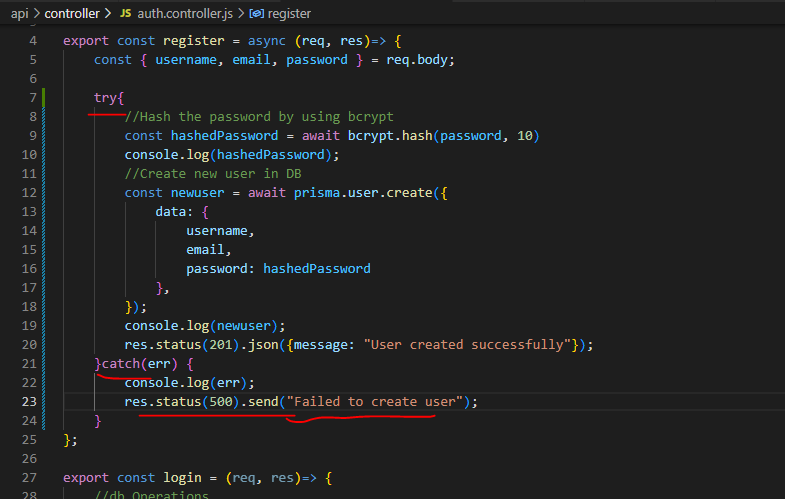
The error is caused due to not pushing changes on prisma schema to db.=> npx prisma db push



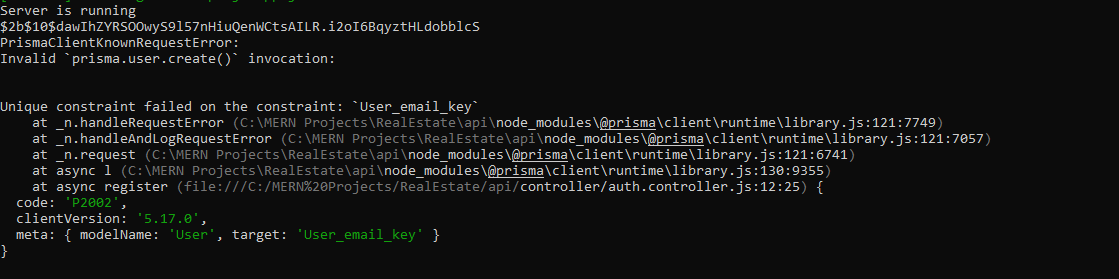
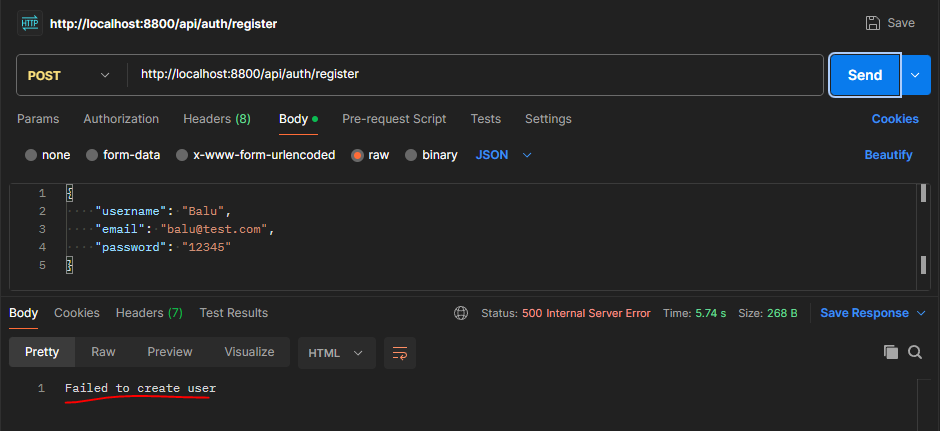




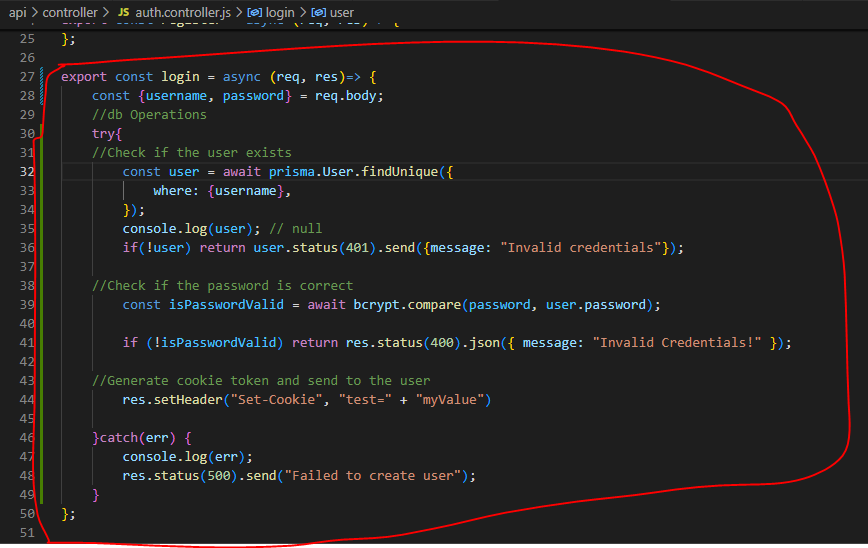
### **Register route:**

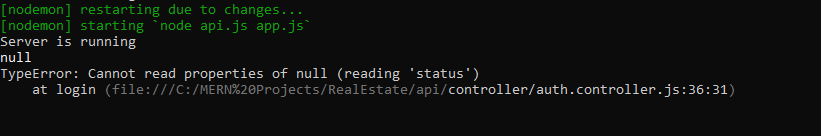


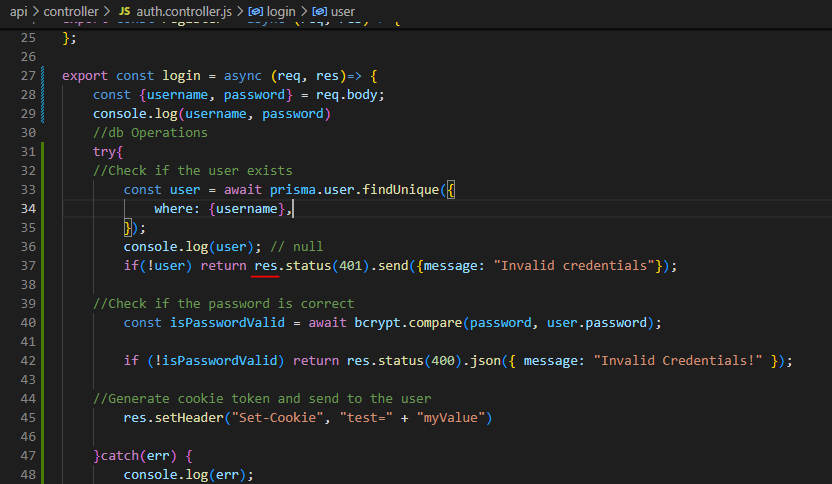
User creation failed due to non unique data sent via post request

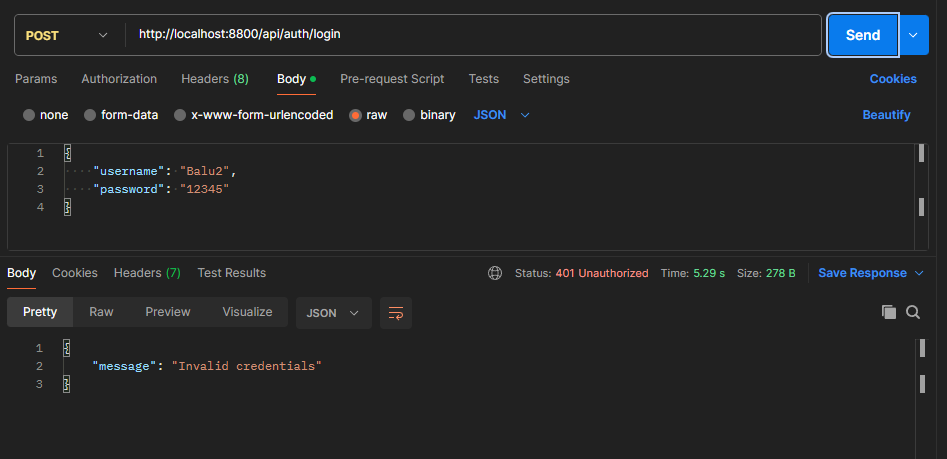
 

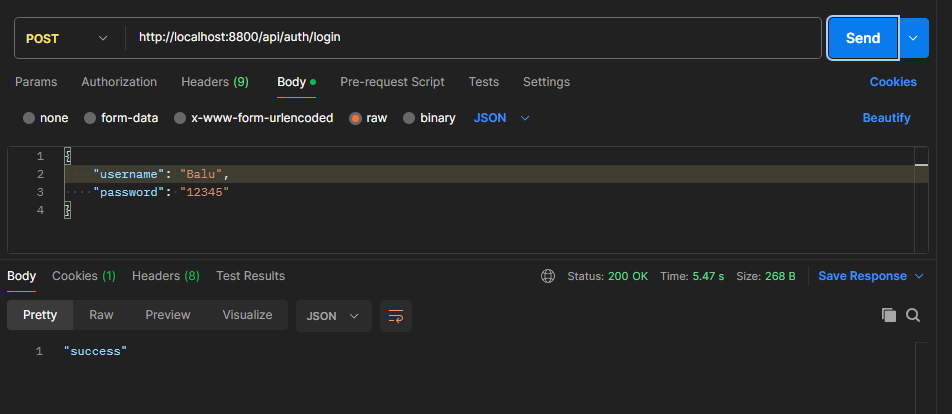
### **Login Route:**



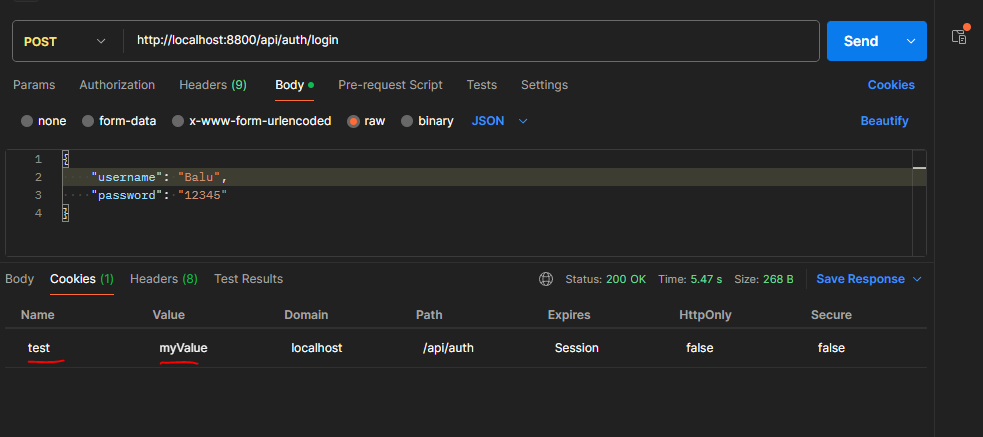


Error caused due to user instead of res



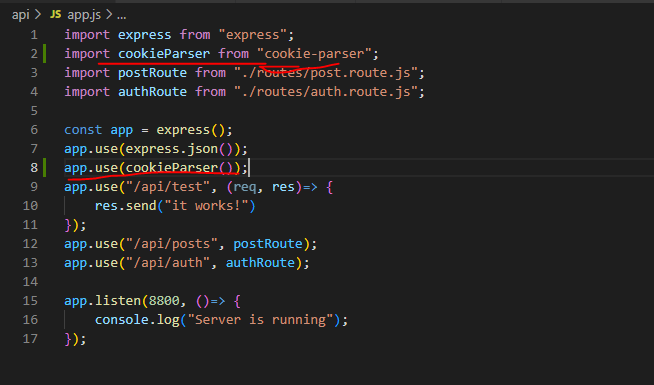


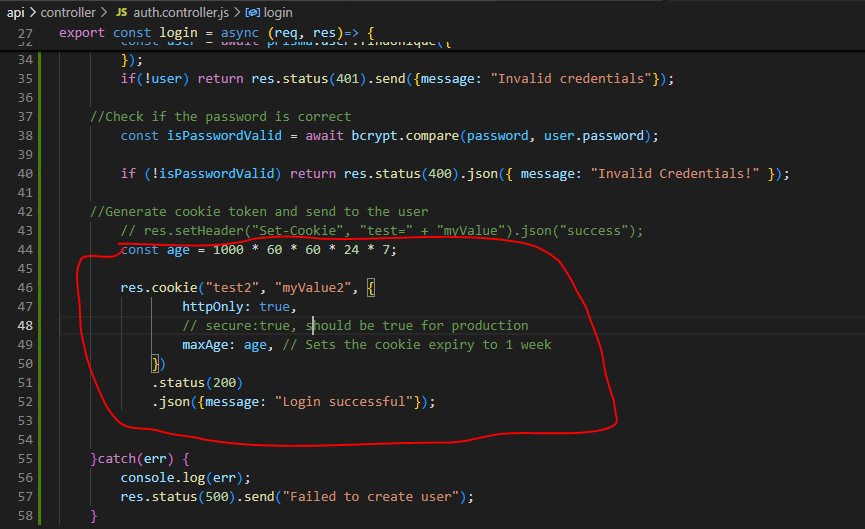
We receive the cookie as well

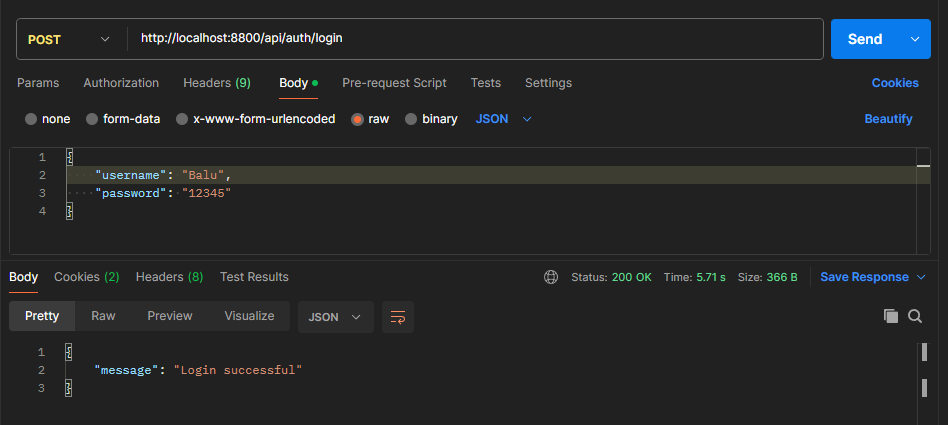


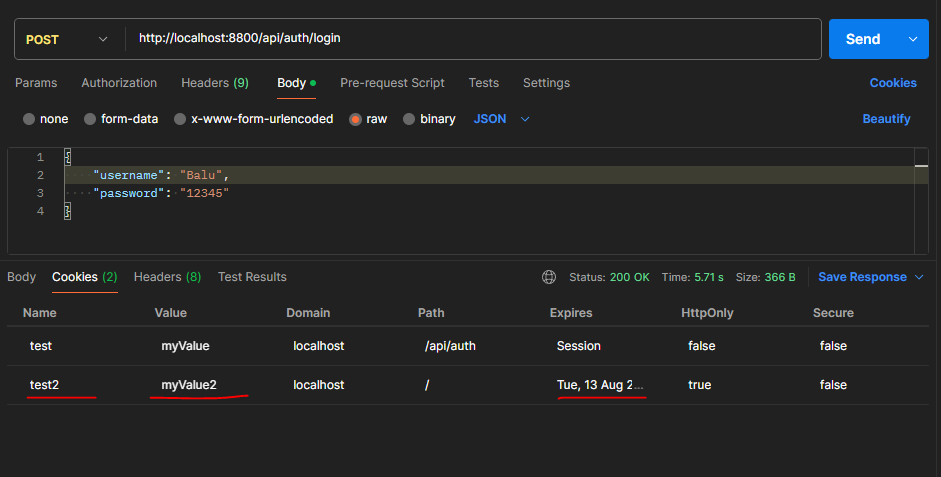
#### **Cookie Parser**

Npm I cookie-parser







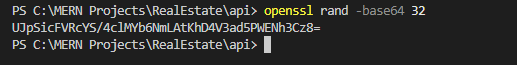


#### **JWT:**

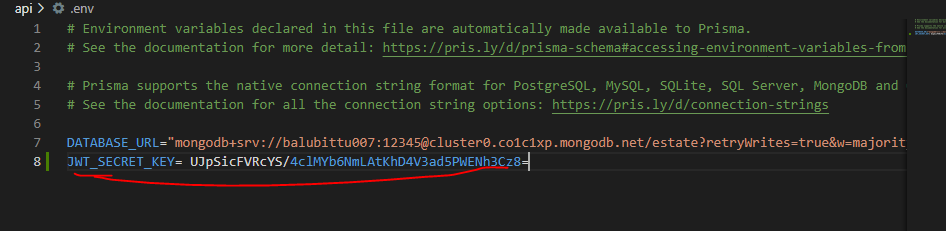
Npm I jsonwebtoken

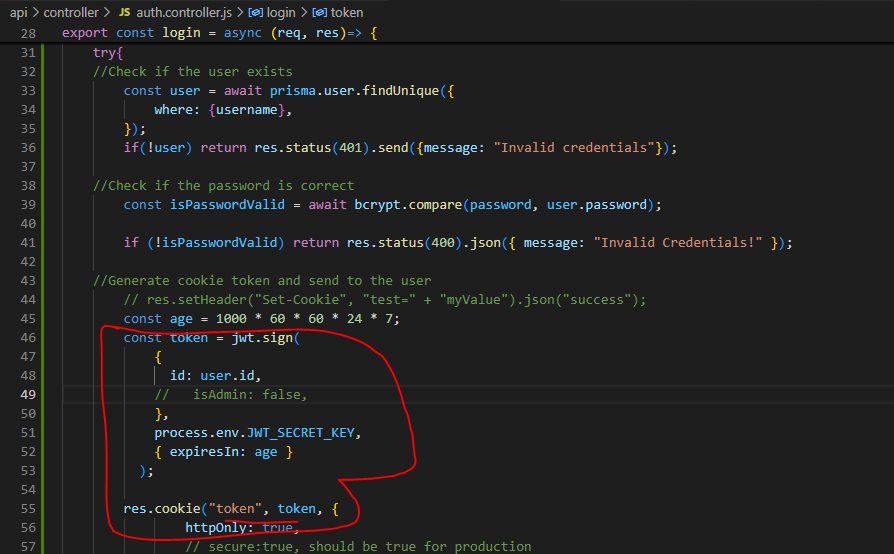
To generate random secret key run below

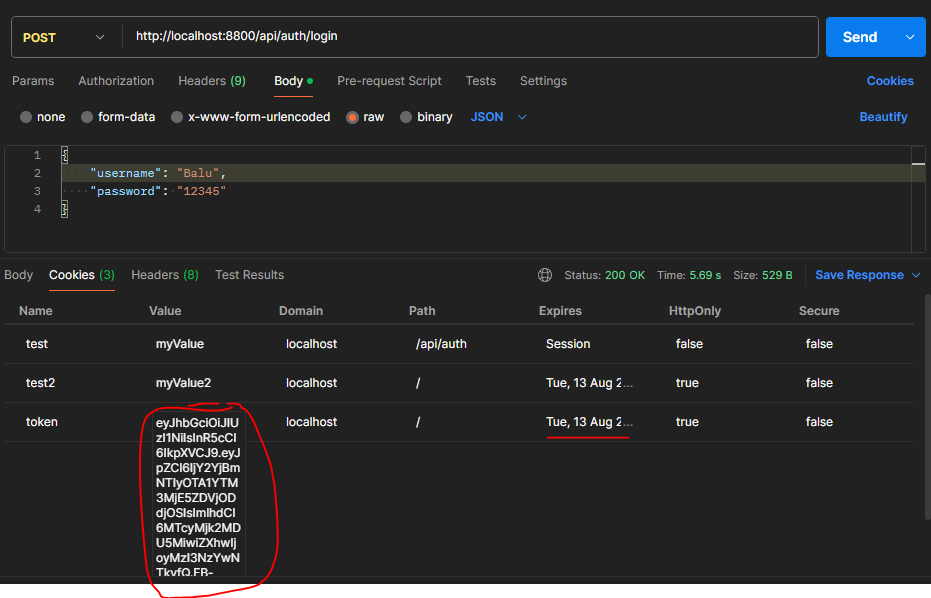
openssl rand -base64 32



Store it inside env file

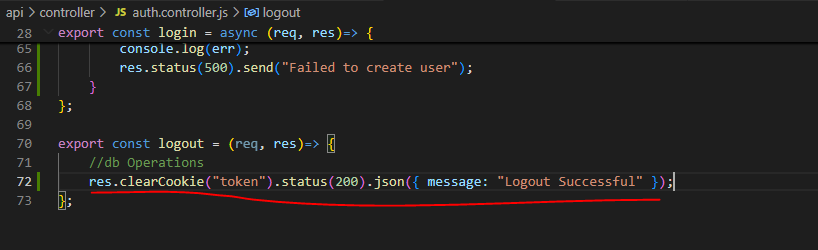


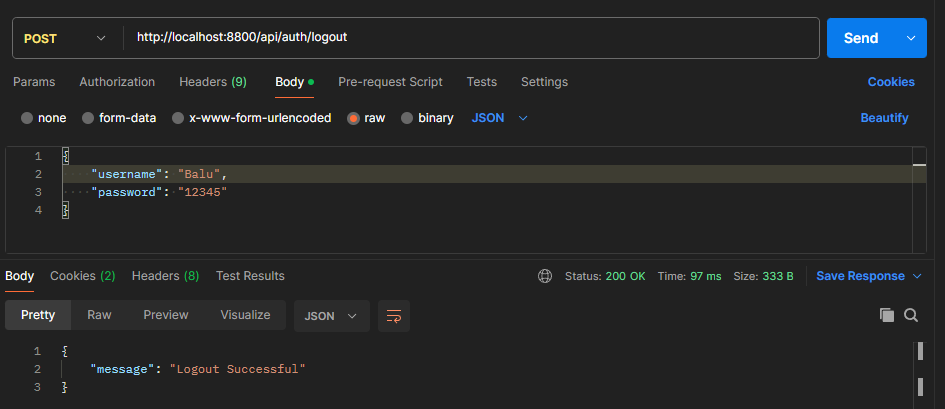




So, right now token consists of user id. On making different api calls we need to decrypt token and get the user id.

### **Logout:**





It will also clear token from cookie

