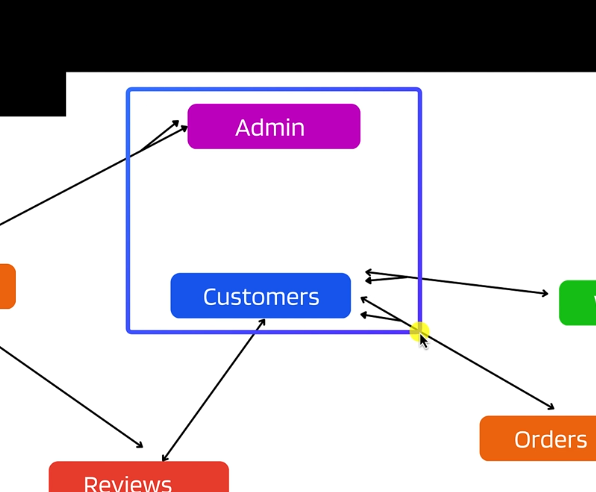
1. 
2. USER DATA MODEL
   1. User.model.js
3. CONNECTION WITH MONGODB
   1. 
   2. BEFORE THE DATABASE WE PROVIDE THE NAME OF THE DATABASE
   3. NAMED TEST DATABASE

Server.js

import http from 'http';

import dotenv from 'dotenv';

import app from './app/app.js';

import dbConnect from './config/dbConnect.js';

dotenv.**config**();

*// create the server*

const PORT = process.env.PORT || 3000;

const server = http.**createServer**(app);

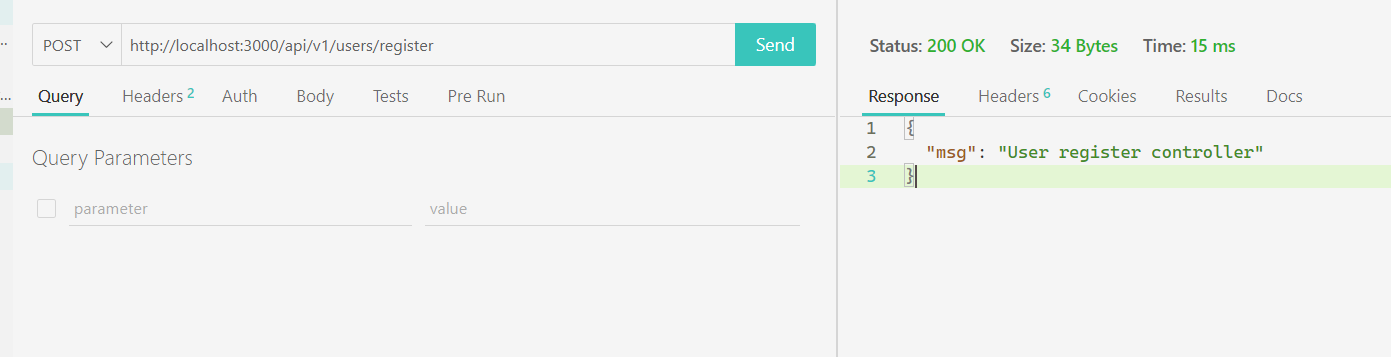
server.**listen**(PORT, () => {

**dbConnect**();

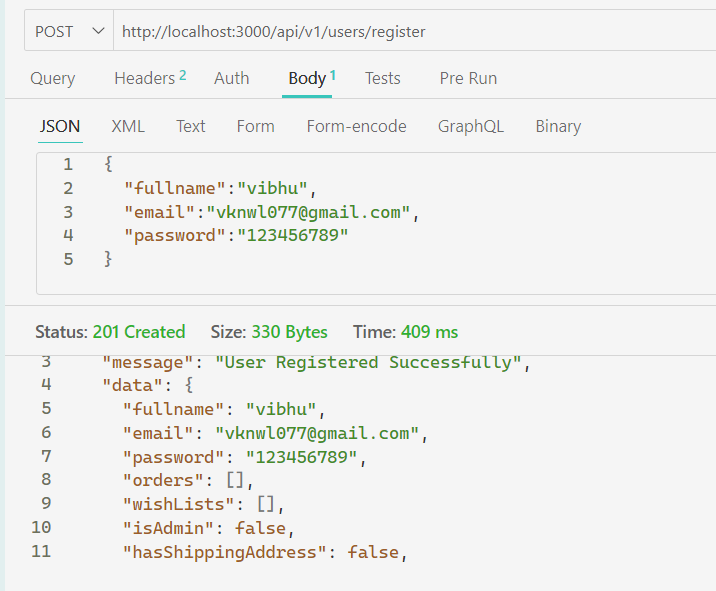
  console.**log**(`Server is up and running on port ${PORT}`);

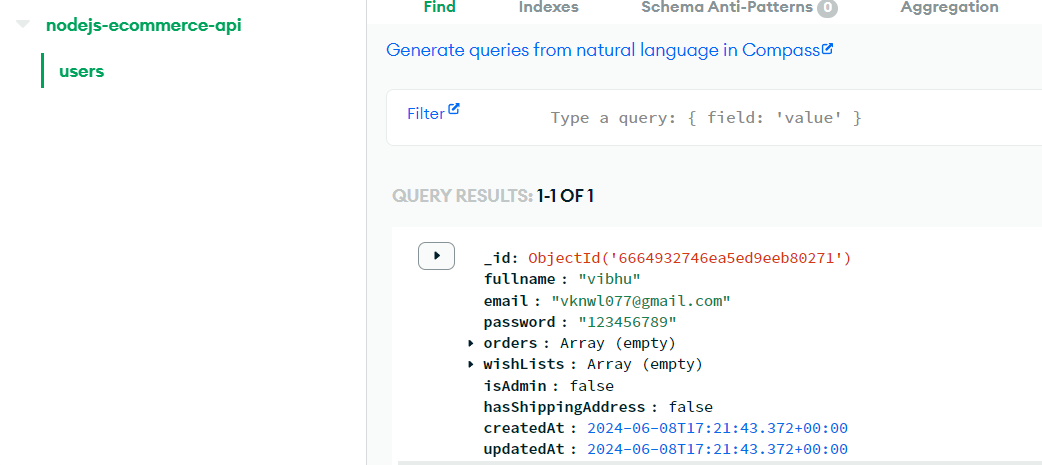
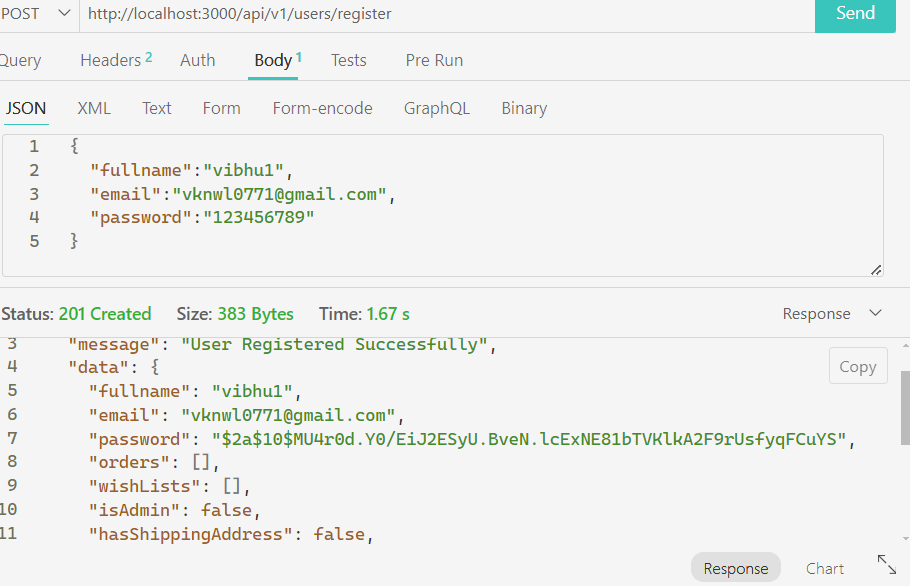
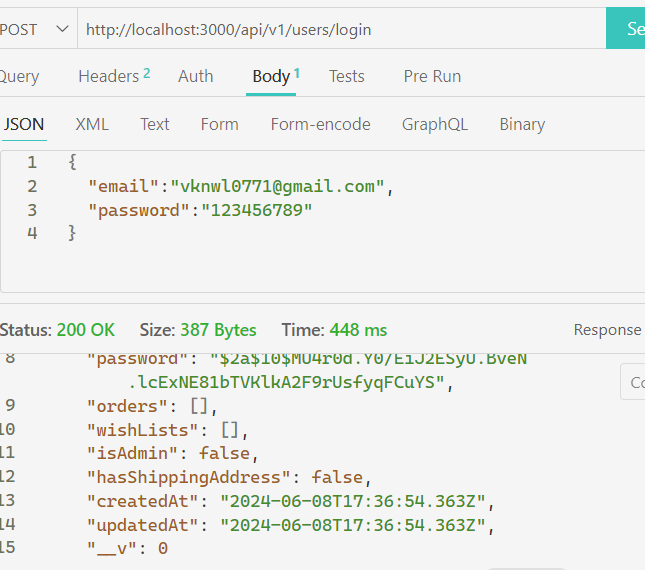
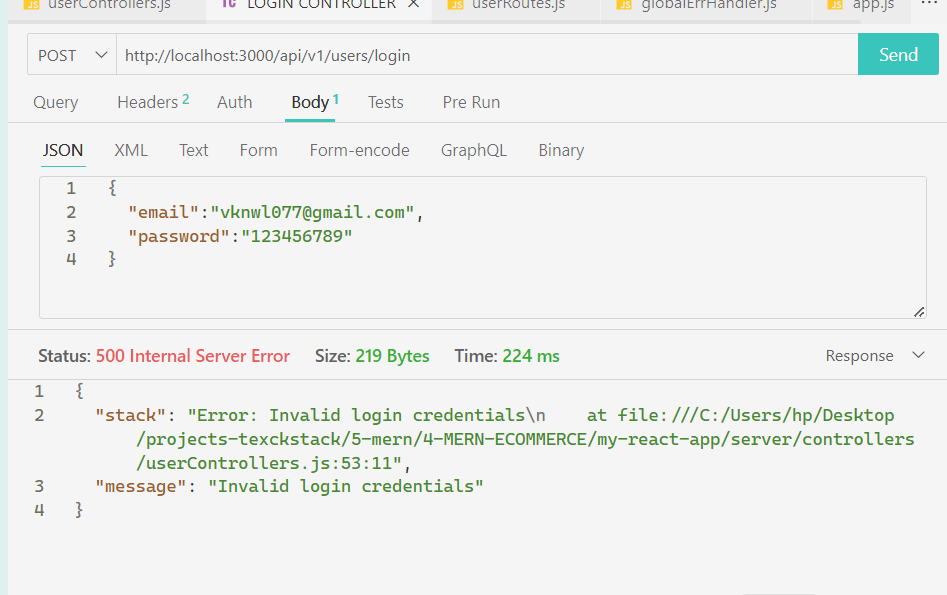
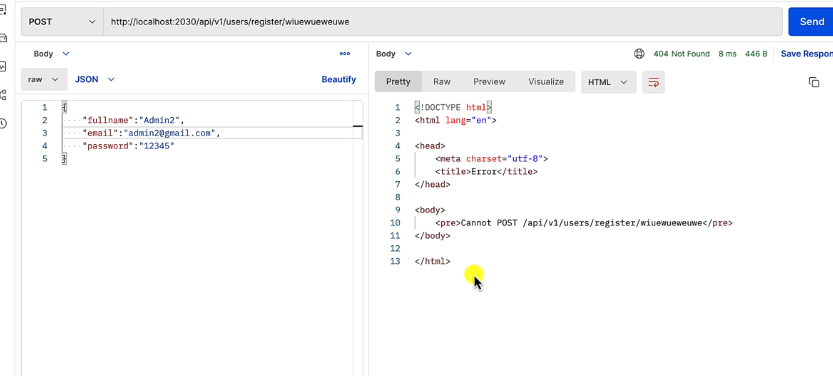
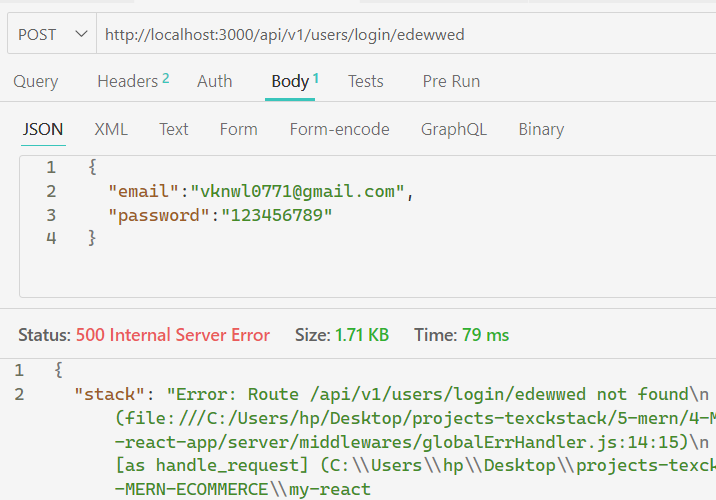
});

* 1. userController🡺userRoutes🡺 app.js



1. CONTROLLERS
   1. Controllers🡺userControllers.js
   2. Routes🡺 userRoutes.js
   3. App.js
   4. Make the USER CONTROLLER

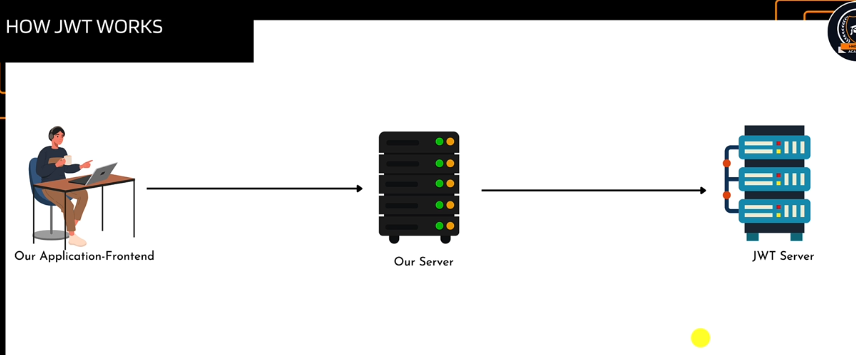


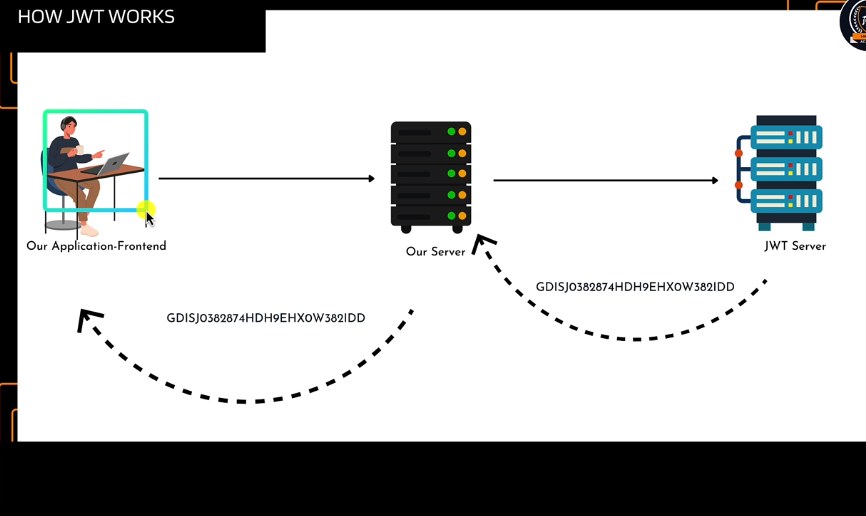
* 1. 
  2. The database name is before the “?” is in the mongo\_uri
  3. 
  4. Hashed the password using bcryptjs
  5. LOGIN CONTROLLER
     + 1. 
       2. ERROR HANDLING
          1. 
          2. INVALID OF STATUS OF 200
          3. We are going to make the global middleware which are below the routes which are catch or block
          4. App.use is the middlewares
          5. First make the middleware globalErrHandler.js and t install the express-error-handelr to wrap the login and register
          6. 
          7. And then below the routes we call the globalErrHandler.js as the middleware
       3. ERRO HANDLING IN THE END POINTS
          1. 
          2. IN THE globalErrHandler we are exiting the middleware that’s why we didn’t invoke the next() but in the not found we are passing It In the next function
          3. The flow in the globalErrHandler is that first it hit the route of the user with login and register and then if there is some error it pass the next which is app.use(globalErrHandler) with the err
          4. So we use the next() in the not found and placed it before the globalErrHandler.js
          5. 
          6. Create the function notFound in the globalErrHandler.js
       4. AUTHENTICATION FLOW

TYPES IS PASSPORT, COOKIE BASED, USE SESSIONS , USE TOKEN

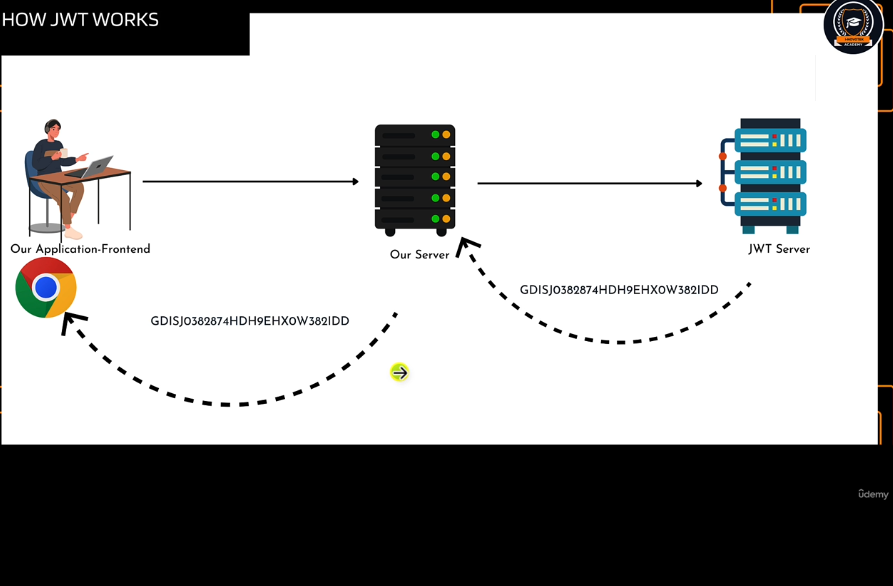
JSON WEB TOKEN

Token based authentication

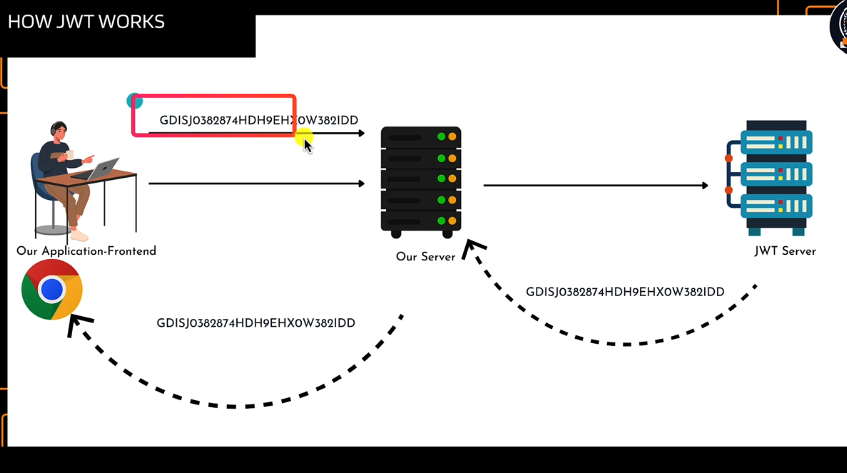




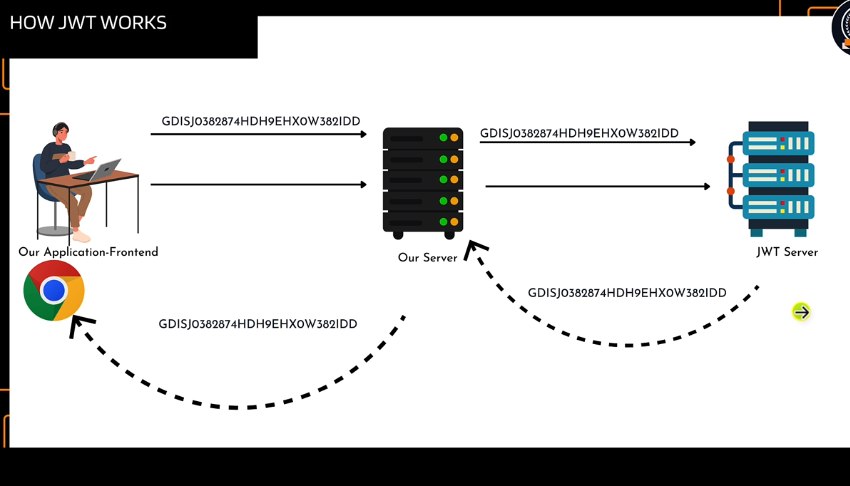
The customer or user is represention the token



Saved in browser local storage

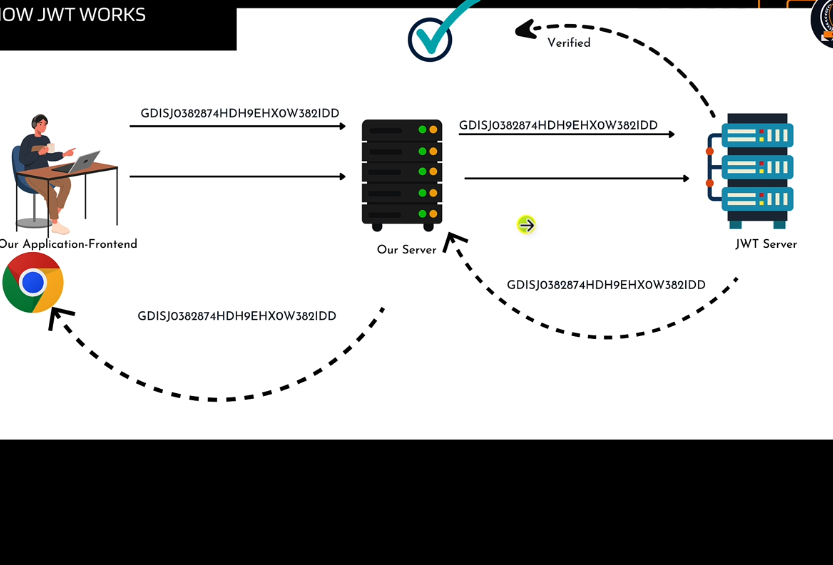
i

If the user performs some actions it verifies the token which send from the server and token which requests the server



Implement some logic and send back to the third party jsonweb token

Check validity of the token same taken at generating



Verified and performs some actions