The project includes 3 functionalities

1. Authentication
2. Image thumb nail generation
3. JSON Patching

**Steps followed:**

Have used Angular 7 for frontend.

The nodejs code is in “server” folder.

**Authentication:**

1. Created login form in angular. Give the command ng serve to run angular.
2. Open the URL “<http://localhost:4200/login>” for login form. Given any user name and password, it sends http request to <http://localhost:3000/auth/login>. Here the jwt token is generated using module “jsonwebtoken” and “jwt.sign”. The token created is returned back to frontend and is saved in localstorage for further validation. (You can find the code in login.component.ts)
3. I have used “JWTINterceptor” for setting headers i.e, to add token as header to http request. You can find the code in “src/app/\_helpers/jwt.interceptor.ts and in “app.module.ts”
4. To test the token validation you can browse the url “http://localhost:3000/home/” which shows the message “Not an authorized user” if there is no token.

If there is a token then it will display token along with login details in console.

For this functionality I have used middleware. I have added below code in routing

router.get('/',authservice.checkToken, function(req,res) {

console.log(req.headers) ;

});

You can find the token validation in authservice -> checktoken function

**Image Thumb nail generation**

For this functionality first I am giving a Google image URL. I am downloading the image in project folder in local in “/download”. I am reading the downloaded file and resizing it and again saving in project folder in “/thumbnail”.

For this I have used frontend again, where I am sending google image in request. You can find the code in imageprocessing.component.ts.Hit the URL “http://localhost:4200/imageprocessing” The request goes to <http://localhost:3000/home/imagedownload>, where the image download and resize functionality is written. You can find the code in “controllers/homecontroller.js” under the routes

router.get('/imagedownload',authservice.checkToken, function(req,res){

})

**JSON Patching**

I have used motcha to test JSON patching. You can find the code for JSON Patching in “homecontroller.js” under the routes

router.get('/jsonpatch',function(req,res){

I have taken a jsonobject and applied json patching and returning the json object after patching