**ASE LAB-9 REPORT**

**Lab id: 5\_2**

**Sandeep Reddy Salkuti (24)**

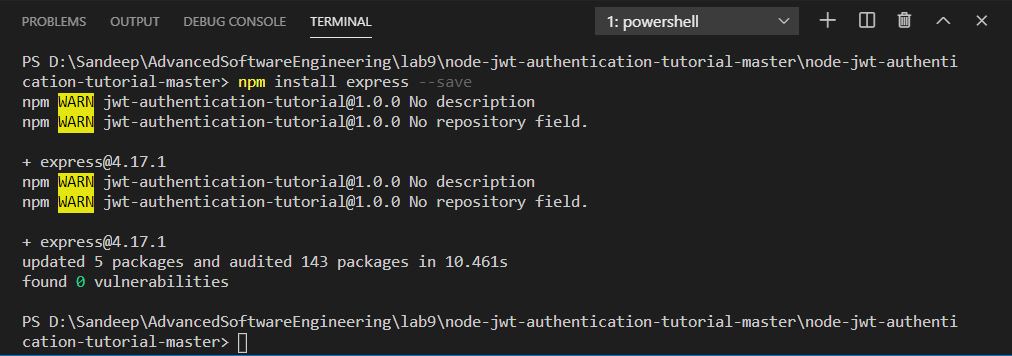
**Pranathi Nalluri (22)**

The main goal of the task is to add authentication to nodejs app using JWTs . For doing this we will create REST API using ExpressJS which is combination of both authenticated and unauthenticated JSON endpoints and a middleware function which will wrap this endpoints and a simple JWT validation function which will check whether incoming header is appropriate authorized header set or not and value of this header is JSON Web Token.

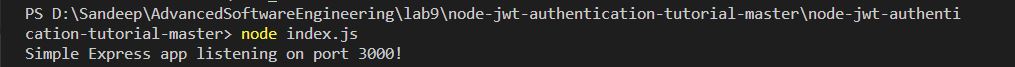
Structure of JWT is featured with “ Header, Payload, Signature ”

In this task we create simple JWT that features with payload and for signing JWT we create private RSA key.

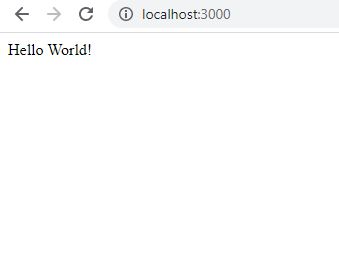
1. For going ahead we need to install dependencies express js node module using “ npm install express –save” like this below



1. On launching the application in v s code editor with running command as node index.js applications runs on specified port 3000.

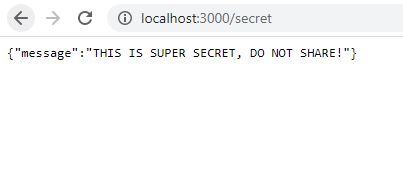


1. After successfully running application below page is displayed as output on the port which is hello world node app.

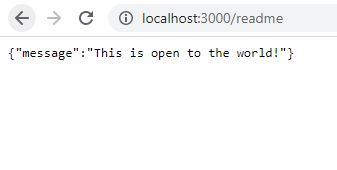


**Adding endpoints:**

1. Next we will add endpoints and how we protect them. We will add few http get endpoints like secret end point which will be protecting one and screenshot is attached below.



A readme endpoint which will be open to world like to anyone and screenshot is attached below.

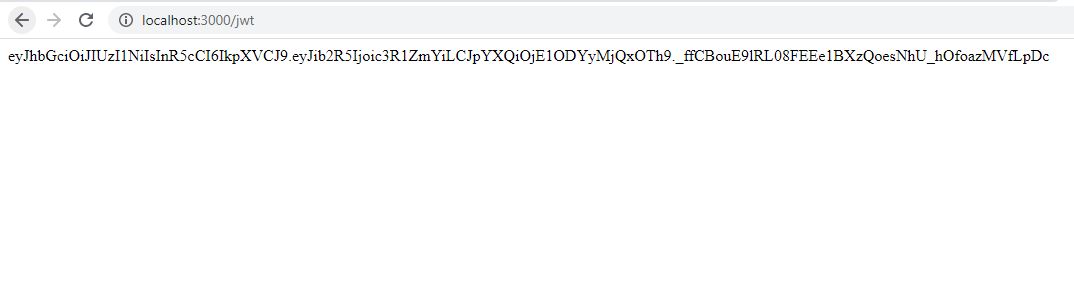


**Generating valid JWT:**

As we have created two end points we will create one additional endpoint for application that will return JWT token to test authentication. So in order to generate and validate JWT we need to import jsonwebtoken node module using following command.

**“ $ npm install –save jsonwebtoken”**

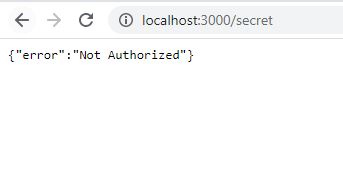
1. Once module is installed we will create a custom jwt endpoint and is signed using a private key named as private.pem in folders. So we will receive a valid JWT which is used for testing in authentications and screenshot is attached below.



**Authentication Middleware:**

In this we will create authorization function which will take incoming request and parse it. If JWT is valid it allows to hit valid endpoint if not error message is displayed as “ not authorized”.

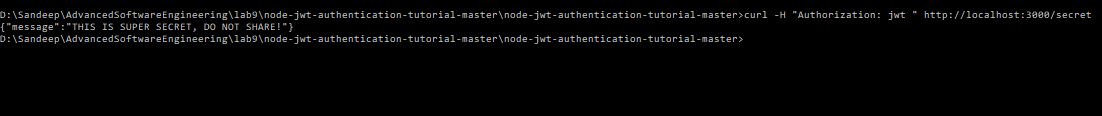
1. Now we will use authorization function to one of our endpoint namely secret and check whether we can authorize it without proper authorization header set and it is denied and screenshot is attached below.

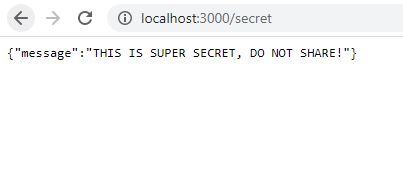


1. As it is denied now we will authorize it using curl command where we will add authorization header and set it to valid jwt got from above jwt endpoint. The curl command is shown below

**curl -H "Authorization: jwt**" <http://localhost:3000/secret>

By giving command we can access endpoint as we have given valid JWT and screenshot is below.



****