Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

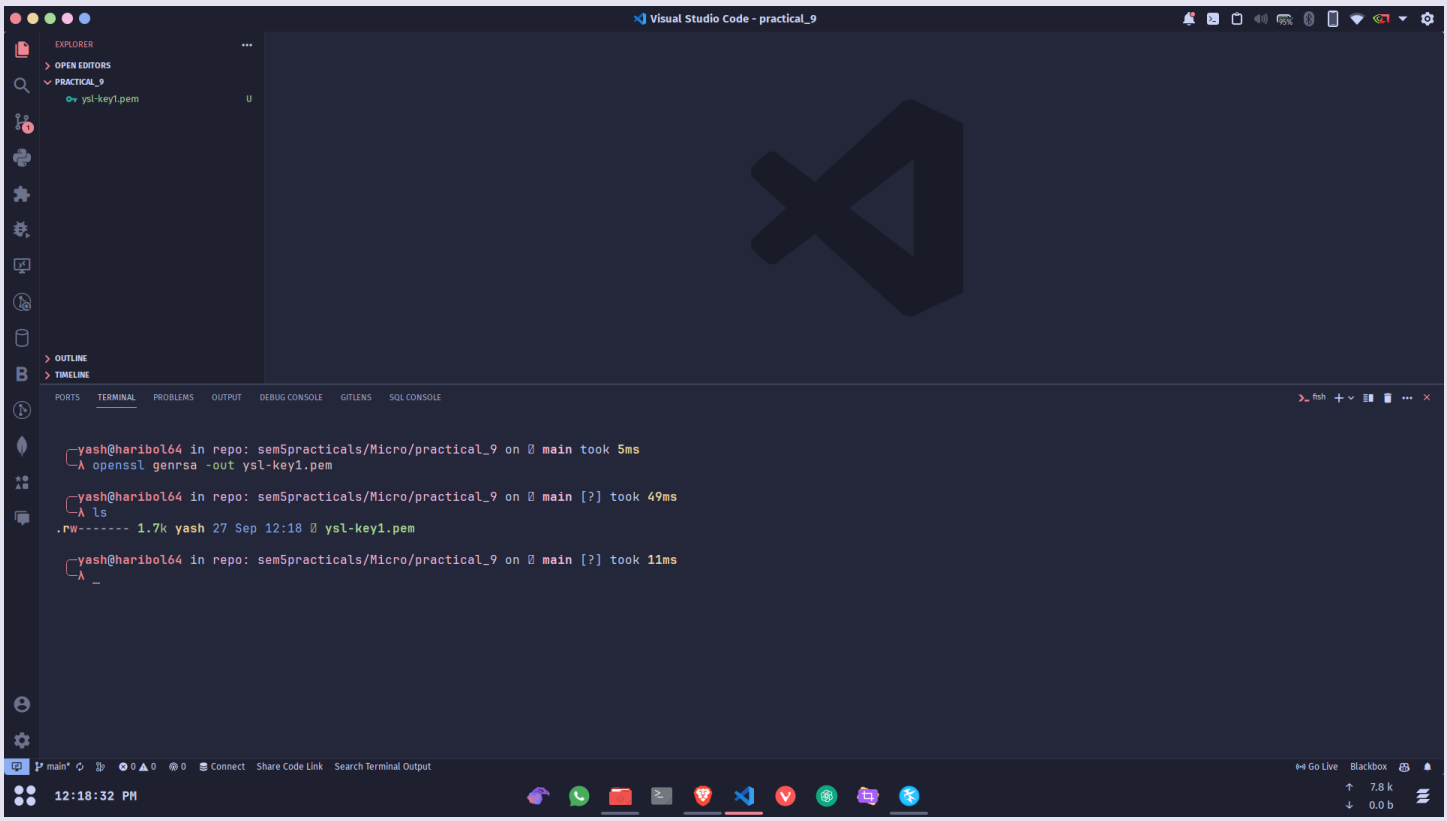
Scenario : Demonstrate the secured HTTP through SSL. Symbiosis Pvt Ltd makes NodeJS-based websites and deploys as well. At the time of deployment, websites require some authentication, and encryption policy for security purposes. Apply the following technique and secure said company's website while deploying :

Tasks, Code and Screenshots :

1. Generate a Public certificate with a public key and certificate

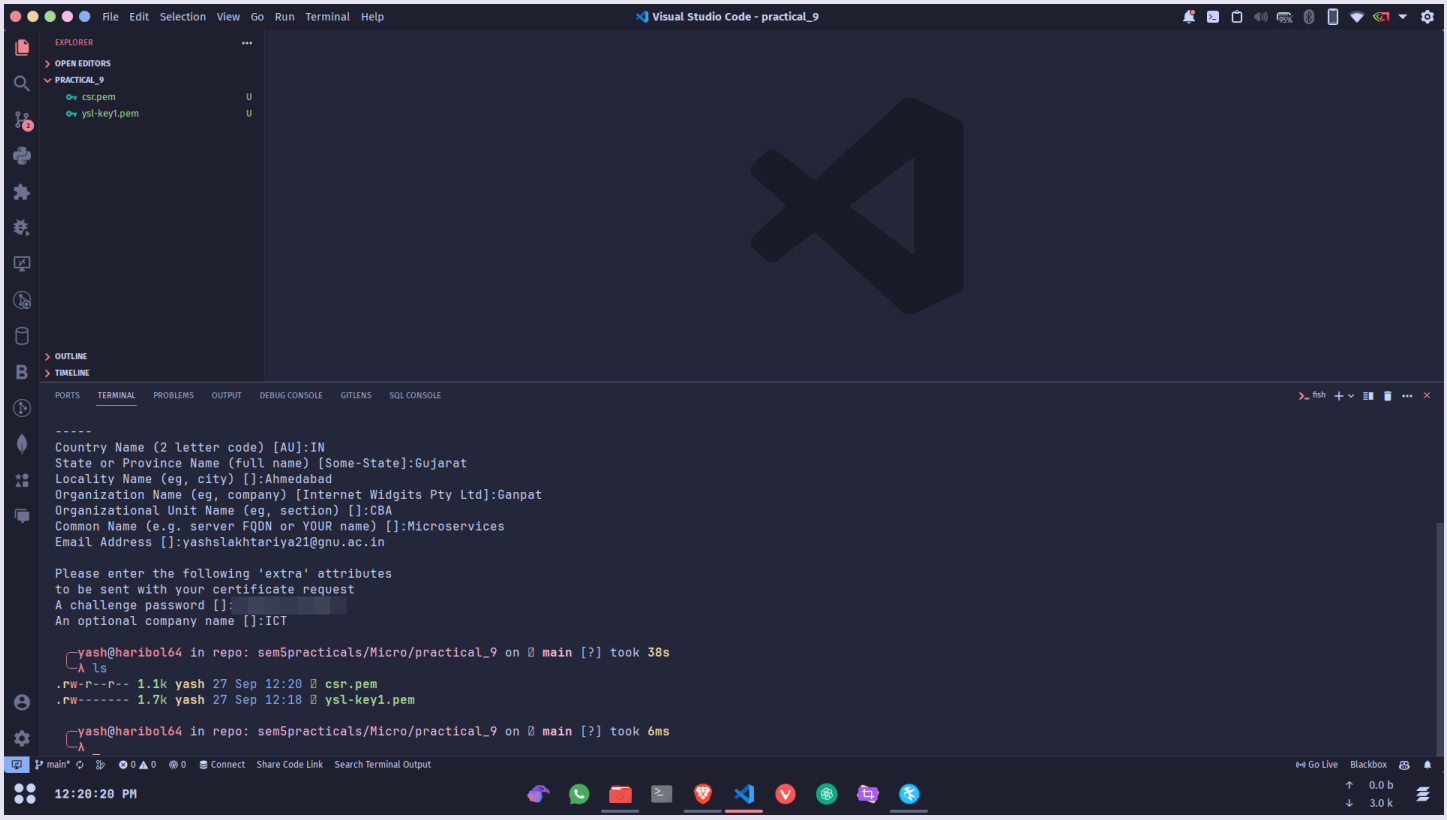
Command :

```
openssl genrsa -out ysl-key1.pem
```



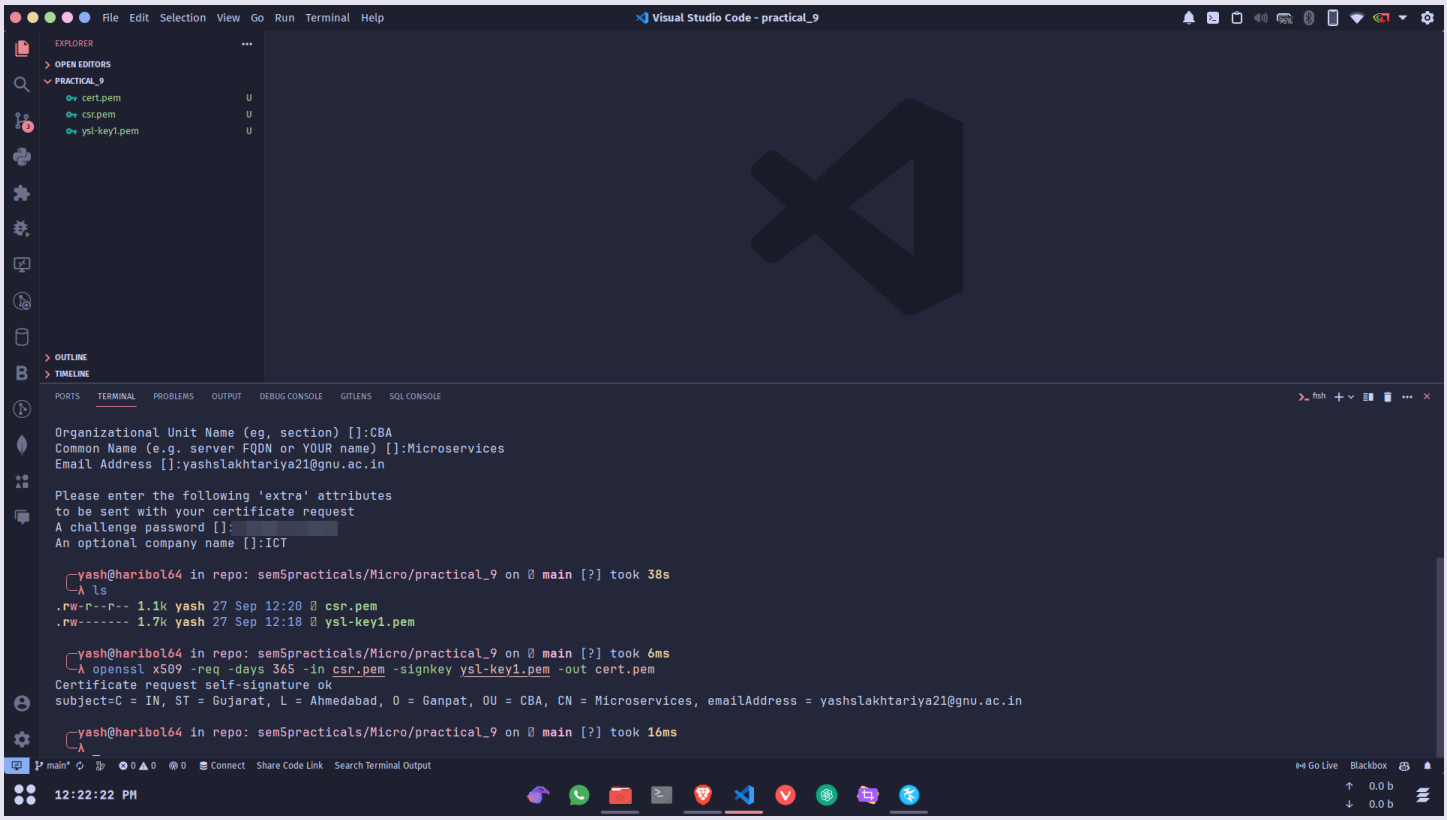
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
openssl req -new -key ysl-key1.pem -out csr.pem
```



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
openssl x509 -req -days 365 -in csr.pem -signkey ysl-key1.pem -out cert.pem
```

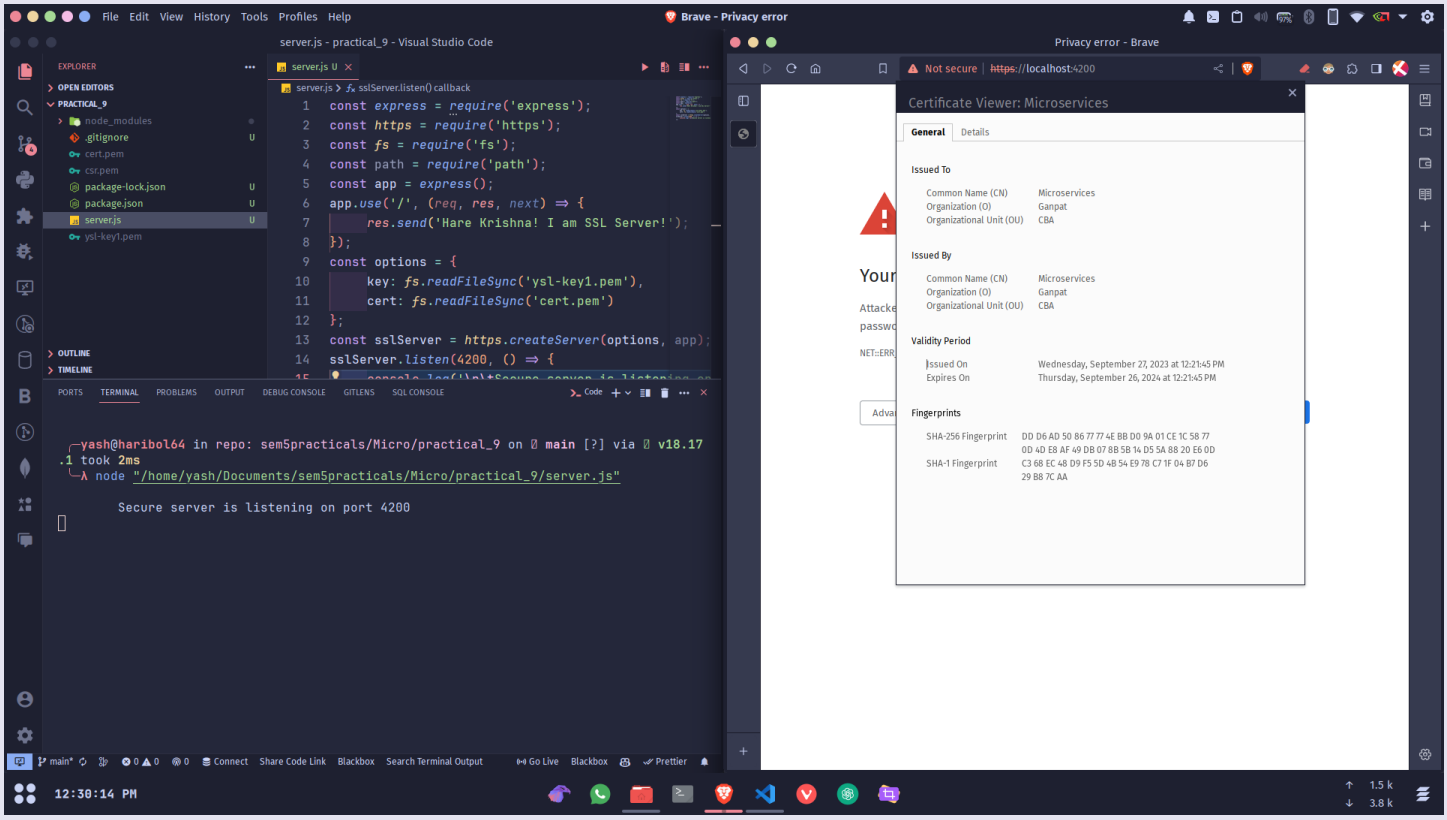


Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

2. Connect it with the website using the HTTPS library

```
const express = require('express');
const https = require('https');
const fs = require('fs');
const path = require('path');
const app = express();
app.use('/', (req, res, next) => {
res.send('Hare Krishna! I am SSL Server!');
});
const options = {
key: fs.readFileSync('ysl-key1.pem'),
cert: fs.readFileSync('cert.pem')
};
const sslServer = https.createServer(options, app);
sslServer.listen(4200, () => {
console.log('\n\tSecure server is listening on port 4200');
});
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9



3. Secure application using Username and password.

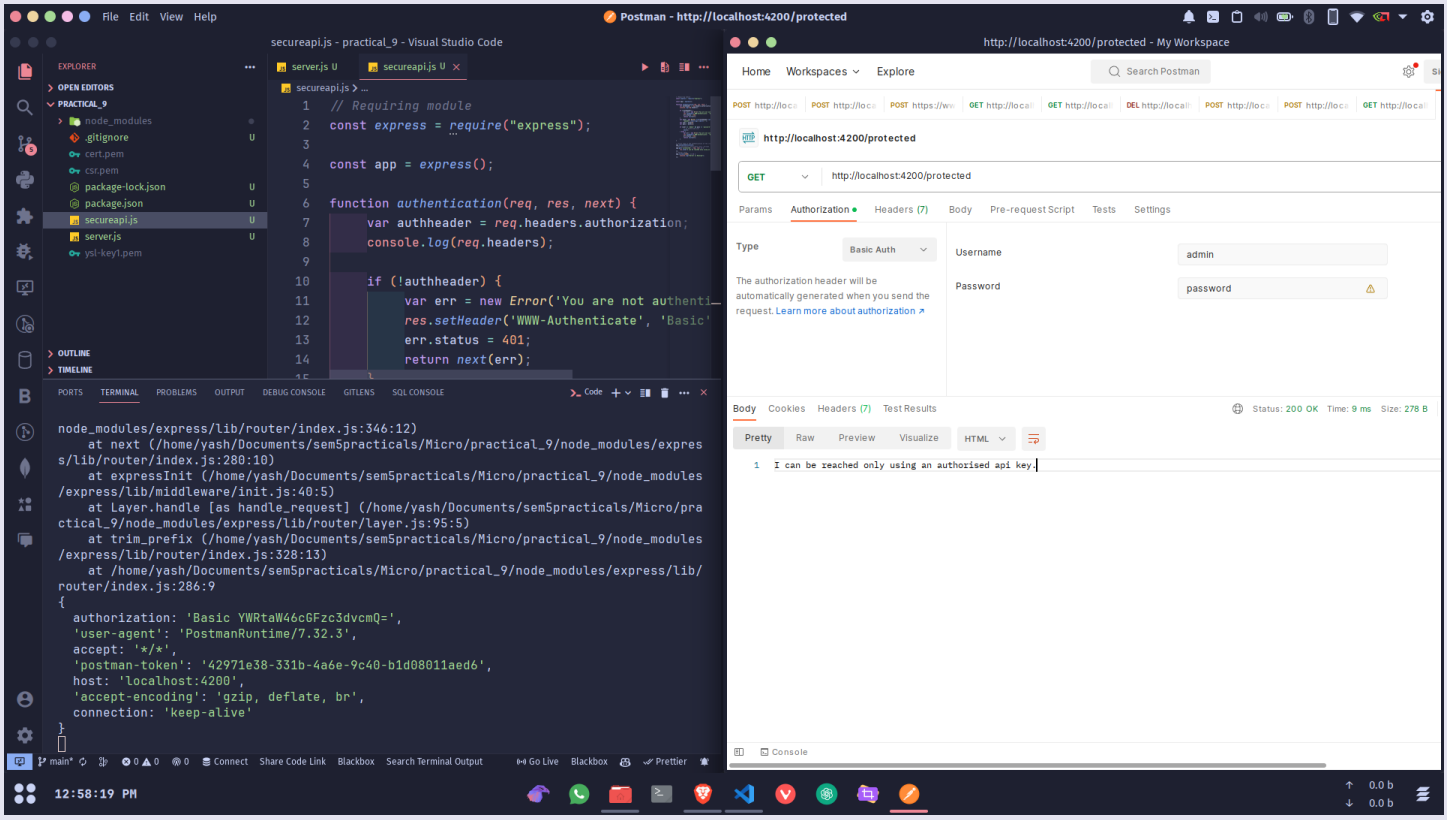
```
// Requiring module
const express = require("express");

const app = express();

function authentication(req, res, next) {
var authheader = req.headers.authorization;
console.log(req.headers);

if (!authheader) {
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
var err = new Error('You are not authenticated!');
res.setHeader('WWW-Authenticate', 'Basic');
err.status = 401;
return next(err);
}
var auth = new Buffer.from(authheader.split(' ')[1],
'base64').toString().split(':');
var user = auth[0];
var pass = auth[1];

if (user == 'admin' && pass == 'password') {
// If Authorized user
next();
} else {
var err = new Error('You are not authenticated!');
res.setHeader('WWW-Authenticate', 'Basic');
err.status = 401;
return next(err);
}
}

// First step is the authentication of the client
app.use(authentication);
//app.use(express.static(path.join(__dirname, 'public')));
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
app.get('/protected', (req, res) => {
res.send('I can be reached only using an authorised api key.');
});
// Server setup
app.listen((4200), () => {
console.log("Server is Running");
});
```



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

4. Secure your REST APIs using the Bearer token technique.

```
const express = require("express");
const jwt = require("jsonwebtoken");

const app = express();
app.use(express.json());

app.post("/api/login", (req, res) => {
//you can do this either synchronously or asynchronously
//if synhronously, you can set a variable to jwt sign and pass it into the
payload with secret key
//if async => call back

//Mock user
const user = {
id: Date.now(),
userEmail: "yashlakhtariya1010@gmail.com",
password: "harekrishna",
};

//send abpve as payload
jwt.sign(
{ user },
"mysecret",
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

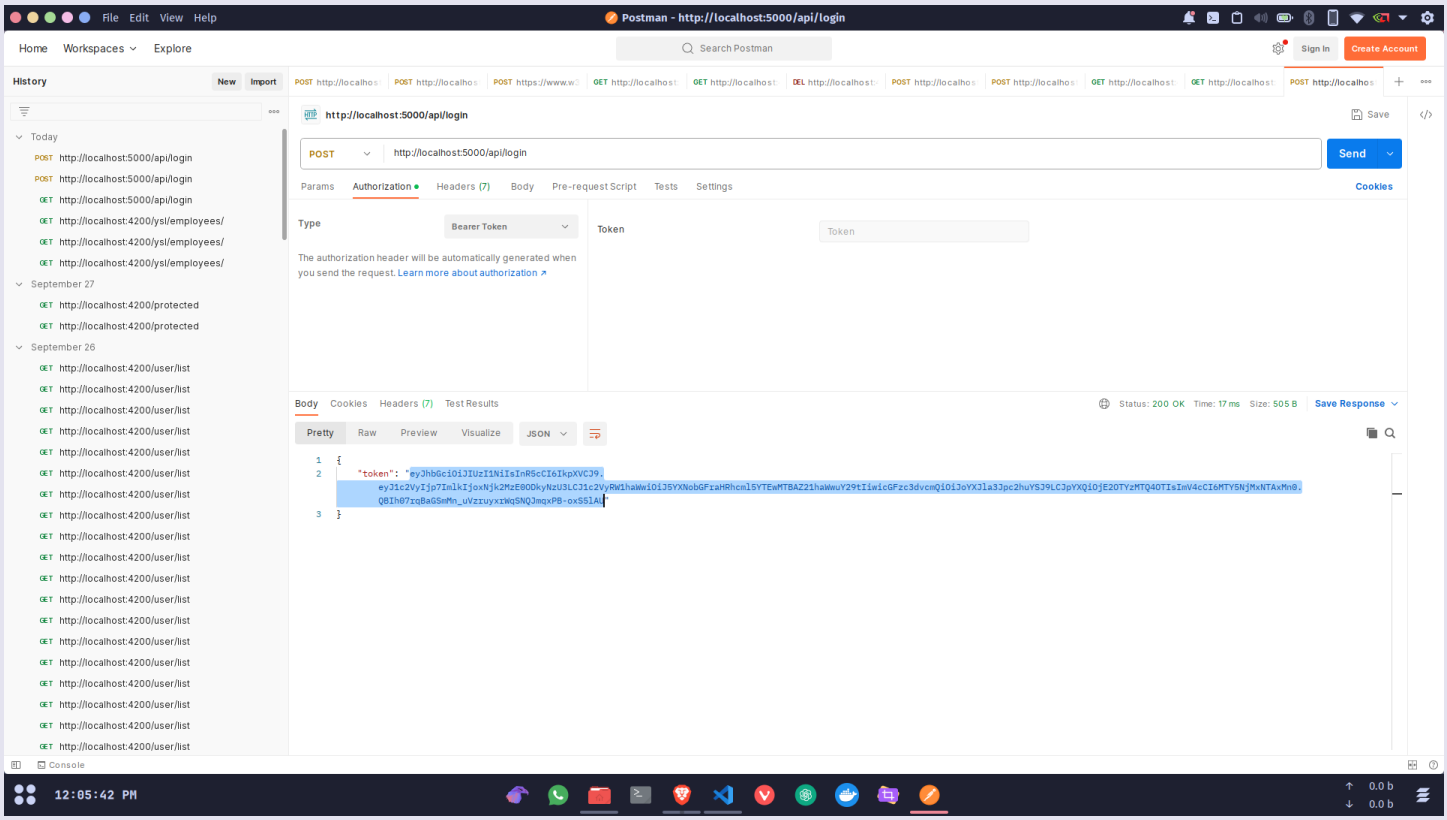
```
{
// expiresIn: "10h" // it will be expired after 10 hours
//expiresIn: "20d" // it will be expired after 20 days
//expiresIn: 120 // it will be expired after 120ms
expiresIn: "3600s", // it will be expired after 120s
},
(err, token) => {
res.json({
token,
});
}
);
});

app.get("/api/profile", verifyToken, (req, res) => {
jwt.verify(req.token, "mysecret", (err, authData) => {
if (err) res.sendStatus(403);
else {
res.json({
message: "Welcome to Profile",
userData: authData,
});
}
});
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

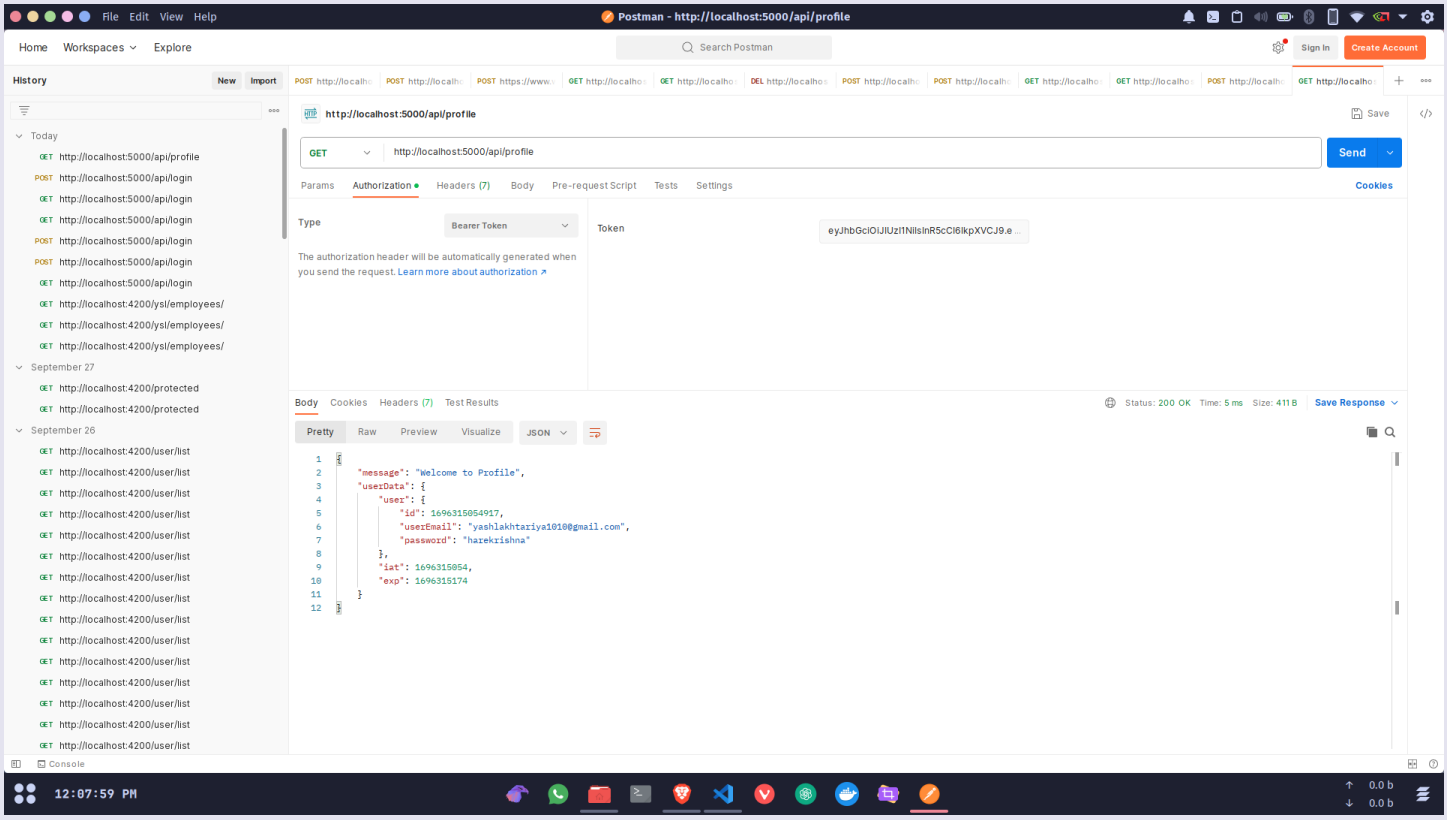
```
});

//Verify Token
function verifyToken(req, res, next) {
//Auth header value = > send token into header
const bearerHeader = req.headers["authorization"];
//check if bearer is undefined
if (typeof bearerHeader !== "undefined") {
//split the space at the bearer
const bearer = bearerHeader.split(" ");
//Get token from string
const bearerToken = bearer[1];
//set the token
req.token = bearerToken;
//next middleweare
next();
} else {
//Fobidden
res.sendStatus(403);
}
}

app.listen(5000, (err) => {
if (err) {
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
console.log(err);
}
console.log("\n\tServer Started on PORT 5000");
});
```



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9



5. Secure your REST APIs created in previous practicals using username and password

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const app = express();
app.use(cors());

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());
let port = 4200;
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
function authentication(req, res, next) {
var authheader = req.headers.authorization;
console.log(req.headers);

if (!authheader) {
var err = new Error('You are not authenticated!');
res.setHeader('WWW-Authenticate', 'Basic');
err.status = 401;
return next(err);
}
var auth = new Buffer.from(authheader.split(' ')[1],
'base64').toString().split(':');
var user = auth[0];
var pass = auth[1];

if (user == 'yash' && pass == 'harekrishna') {
// If Authorized user
next();
} else {
var err = new Error('You are not authenticated!');
res.setHeader('WWW-Authenticate', 'Basic');
err.status = 401;
return next(err);
```

```
}
}

let emps = [
{
id: "1",
name: "Madhav",
email: "madhav@haribol.com",
phone: "1234567890",
designation: "CEO",
},
{
id: "2",
name: "Gopal",
email: "gopal@haribol.com",
phone: "0123456789",
designation: "HR",
},
{
id: "3",
name: "Damodar",
email: "damodar@haribol.com",
phone: "9876543210",
designation: "Finance",
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
},
{
id: "4",
name: "Keshav",
email: "keshav@haribol.com",
phone: "9630147852",
designation: "HR",
},
];

app.use(authentication);

app.get("/", (req, res) => {
res.send("Welcome to YSL Company REST API!");
});

app.get("/ysl/employees", (req, res) => {
res.send(emps);
});

app.get("/ysl/employees/:id", (req, res) => {
const emp = emps.find(({ id }) => id === req.params.id);

if (!emp) res.status(404).send("Not found!");
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
res.send(emp);
});

app.post("/ysl/employees", (req, res) => {
let emp = {
id: req.body.id,
name: req.body.name,
email: req.body.email,
phone: req.body.phone,
dept: req.body.dept,
};
emps.push(emp);
res.send(emp);
});

app.put("/ysl/employees/:id", (req, res) => {
let emp = emps.find(({ id }) => id === req.params.id);
if (!emp) res.status(404).send("Not found!");
emp.id = req.body.id;
emp.name = req.body.name;
emp.email = req.body.email;
emp.ln = req.body.ln;
emp.phone = req.body.phone;
emp.designation = req.body.designation;
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9

```
res.send(emp);
});

app.delete("/ysl/employees/:id", (req, res) => {
const emp = emps.find(({ id }) => id === req.params.id);
if (!emp) res.status(404).send("Not found!");

const index = emps.indexOf(emp);
emps.splice(index, 1);

res.send(emp);
});

app.listen(port, () => {
console.log(`\nThe server is running on port ${port}\n`);
});
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 9