Name: Patel Shivam S.
Enroll no: 21162101019
Sem: 5 Batch: 51 Branch: CBA

Sub: Microservices

Practical 9

AIM: **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:**

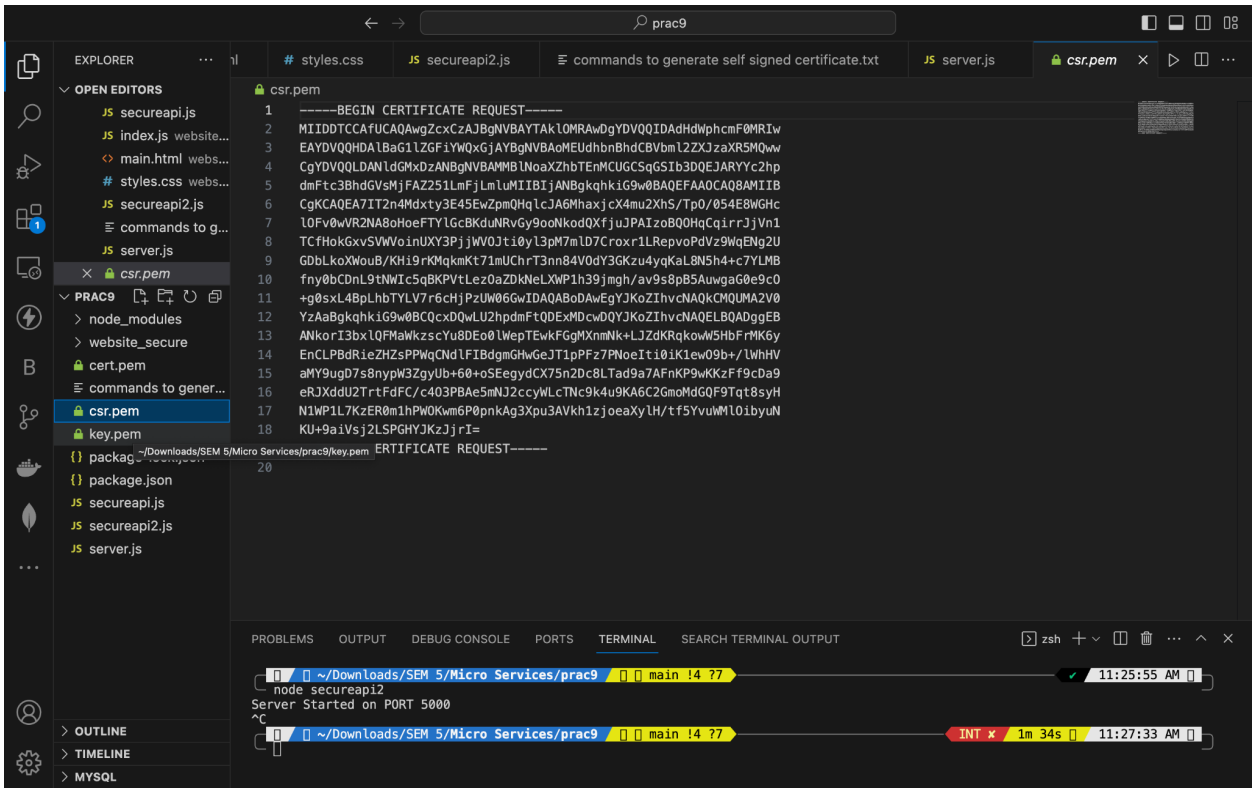
1. **Generate a Public certificate with a public key and certificate**
2. **Connect it with the website using the HTTPS library**
3. **Secure application using Username and password.**
4. **Secure your REST APIs using the Bearer token technique.**
5. **Secure your REST APIs created in previous practicals using username and password**

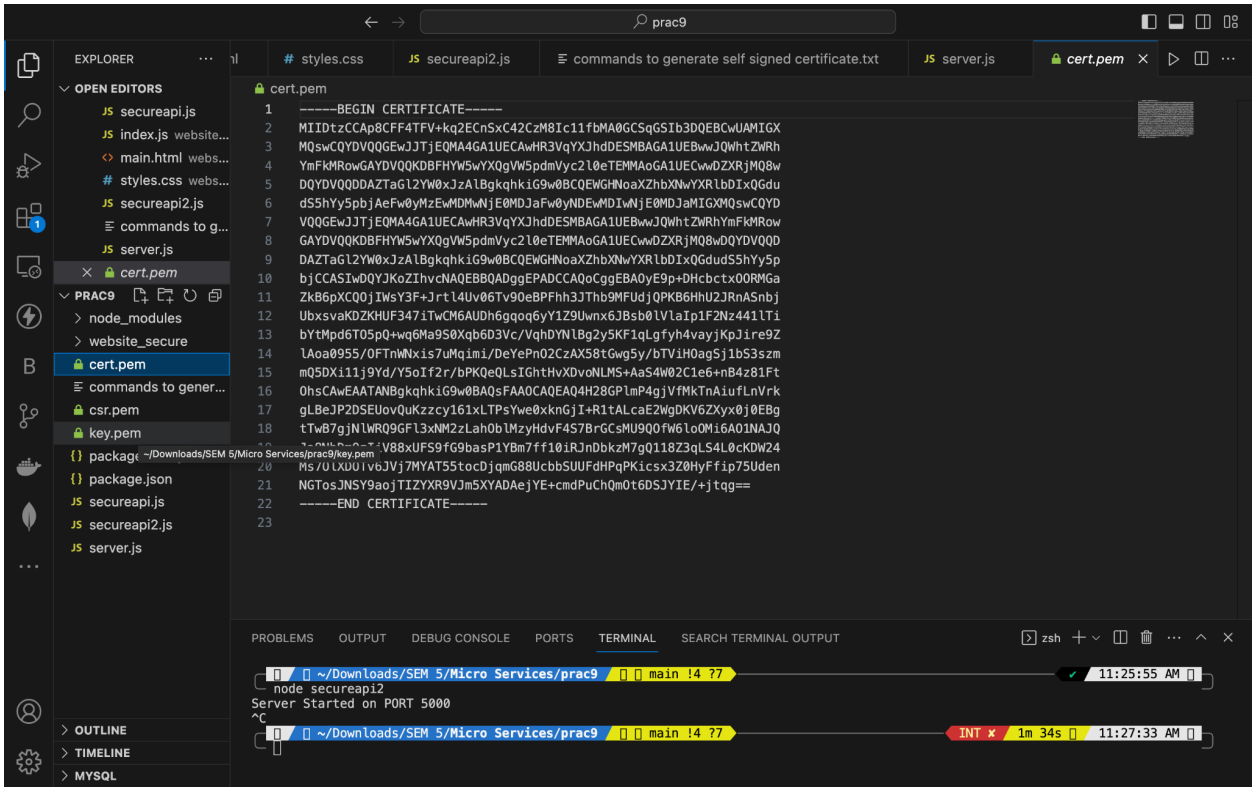
GITHUB LINK:
https://github.com/Shivam3783/microservice_practicals/tree/main/prac9

Practical 9.1: Generate a Public certificate with a public key and certificate

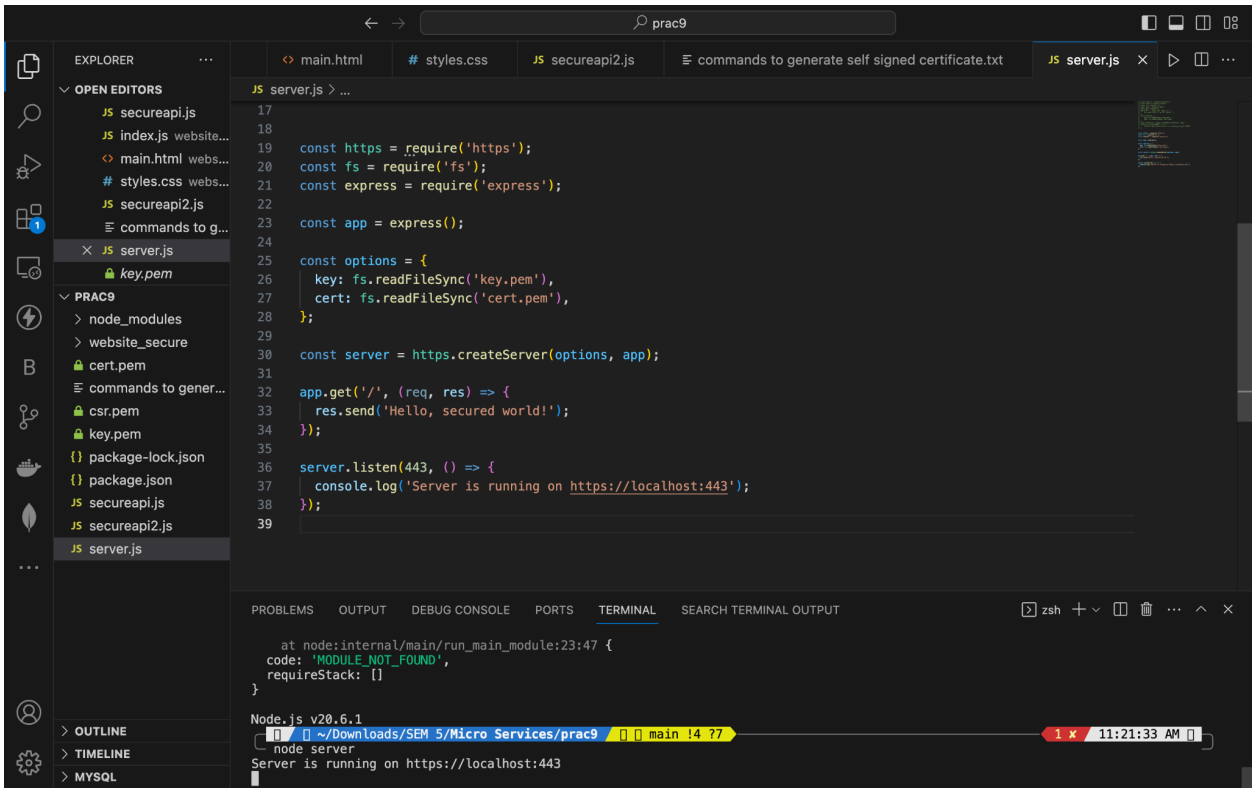
```
openssl genrsa -out key.pem
openssl req -new -key key.pem -out csr.pem
openssl x509 -req -days 365 -in csr.pem -signkey key.pem -out cert.pem
```

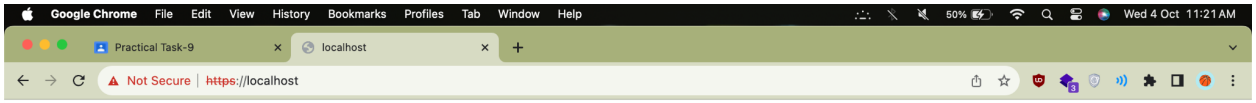
After run this code we will get public key and certificate.


prac9
EXPLORER
styles.css
secureapi2.js
commands to generate self signed certificate.txt
server.js
csr.pem
OPEN EDITORS
secureapi.js
index.js website...
main.html webs...
styles.css webs...
secureapi2.js
commands to g...
server.js
csr.pem
PRAC9
node_modules
website_secure
cert.pem
commands to gener...
csr.pem
key.pem
package.json
secureapi.js
secureapi2.js
server.js
-----BEGIN CERTIFICATE REQUEST-----
MIIDDTCCAfUCAQAwgZcxCzAJBgNVBAYTAkl0MRAwDgYDVQQIDAdHdWphcmF0MRIw
EAYDVQQHDAlBaG1lZGFiYWQxGjAYBgNVBAoMEUdhbnBhdCBVbml2ZXJzaXR5MQww
CgYDVQQLDANldGMxDzANBgNVBAMMBlNoaXZhbTEnMCUGCSqGSIb3DQEJARYYc2hp
dmFtc3BhdGVsMjFAZ251LmFjLmluMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA7IT2n4Mdxty3E45EwZpmQHqlcJA6MhaxjcX4mu2XhS/TpO/054E8WGHc
lOFv0wVR2NA8oHoeFTYlGcBKduNRvGy9ooNkodQXfjuJPAIzoBQOHqCqirrJjVn1
TCfHokGxvSVWVoinUXY3PjjWVOJti0yl3pM7mlD7Croxr1LRepvoPdVz9WqENg2U
GDbLkoXWouB/KHi9rKMqkmKt71mUChrT3nn84VOdY3GKzu4yqKaL8N5h4+c7YLMB
fny0bCDnL9tNWIc5qBKPVtLezOaZDkNeLXWP1h39jmgh/av9s8pB5AuwgaG0e9cO
+g0sxL4BpLhbTYLV7r6cHjPzUW06GwIDAQABoDAwEgYJKoZIhvcNAQkCMQUMA2V0
YzAaBgkqhkiG9w0BCQcxDQwLU2hpdmFtQDExMDcwDQYJKoZIhvcNAQELBQADggEB
ANkorI3bxlQFMaWkzscYu8DEo0lWepTEwkFGgMXnmNk+LJZdKRqkowW5HbFrMK6y
EnCLPBdRieZHZsPPWqCNdlFIBdgmGHwGeJT1pPFz7PNoeIti0iK1ewO9b+/lWhHV
aMY9ugD7s8nypW3ZgyUb+60+oSEegydCX75n2Dc8LTad9a7AFnKP9wKKzFf9cDa9
eRJXddU2TrtFdFC/c4O3PBAe5mNJ2ccyWLcTNc9k4u9KA6C2GmoMdGQF9Tqt8syH
N1WP1L7KzER0m1hPWOKwm6P0pnkAg3Xpu3AVkh1zjoeaXylH/tf5YvuWMlOibyuN
KU+9aiVsj2LSPGHYJKzJjrI=
-----END CERTIFICATE REQUEST-----
~/Downloads/SEM 5/Micro Services/prac9/key.pem
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
TERMINAL
SEARCH TERMINAL OUTPUT
zsh
~/Downloads/SEM 5/Micro Services/prac9
main !4 ?7
11:25:55 AM
node secureapi2
Server Started on PORT 5000
^C
INT
1m 34s
11:27:33 AM
OUTLINE
TIMELINE
MYSQL

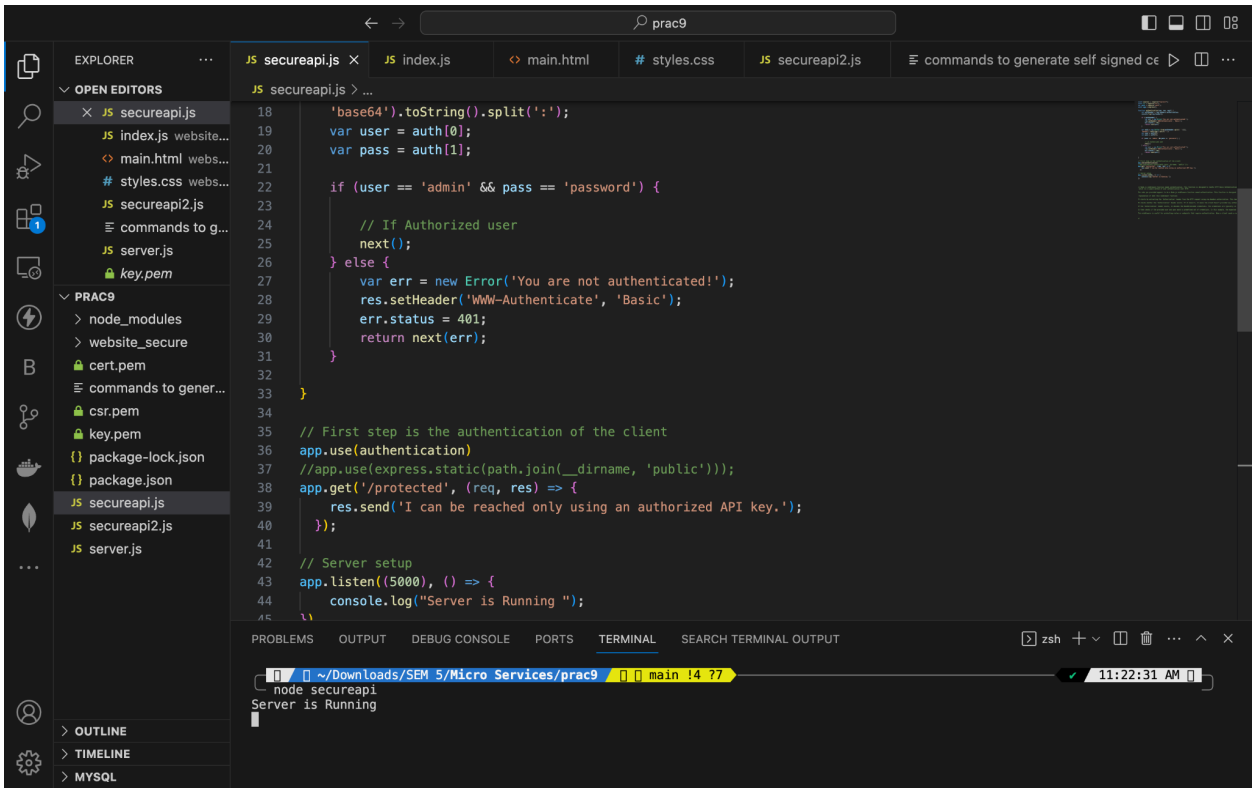

prac9
EXPLORER
styles.css
secureapi2.js
commands to generate self signed certificate.txt
server.js
cert.pem
OPEN EDITORS
secureapi.js
index.js website...
main.html webs...
styles.css webs...
secureapi2.js
commands to g...
server.js
cert.pem
PRAC9
node_modules
website_secure
cert.pem
commands to gener...
csr.pem
key.pem
package.json
secureapi.js
secureapi2.js
server.js
-----BEGIN CERTIFICATE-----
MIIDtzCCAp8CFF4TFV+kq2ECnSxC42CzM8Ic11fbMA0GCSqGSIb3DQEBCwUAMIGX
MQswCQYDVQQGEwJJTjEQMA4GA1UECAwHR3VqYXJhdDESMBAGA1UEBwwJQWhtZWRh
YmFkMRowGAYDVQQKDBFHYW5wYXQgVW5pdmVyc2l0eTEMMAoGA1UECwwDZXRjMQ8w
DQYDVQQDDAZTaGl2YW0xJzAlBgkqhkiG9w0BCQEWGHNoaXZhbXNwYXRlbDIxQGdu
dS5hYy5pbjAeFw0yMzEwMDMwNjE0MDJaFw0yNDEwMDIwNjE0MDJaMIGXMQswCQYD
VQQGEwJJTjEQMA4GA1UECAwHR3VqYXJhdDESMBAGA1UEBwwJQWhtZWRhYmFkMRow
GAYDVQQKDBFHYW5wYXQgVW5pdmVyc2l0eTEMMAoGA1UECwwDZXRjMQ8wDQYDVQQD
DAZTaGl2YW0xJzAlBgkqhkiG9w0BCQEWGHNoaXZhbXNwYXRlbDIxQGdudS5hYy5p
bjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOyE9p+DHcbctxOORMGa
ZkB6pXCQOjIWsY3F+Jrtl4Uv06Tv9OeBPFhh3JThb9MFUdjQPKB6HhU2JRnASnbj
UbxsvaKDZKHUF347iTwCM6AUDh6gqoq6yY1Z9Uwnx6JBsb0lVlaIp1F2Nz441lTi
bYtMpd6TO5pQ+wq6Ma9S0Xqb6D3Vc/VqhDYNlBg2y5KF1qLgfyh4vayjKpJire9Z
lAoa0955/OFTnWNxis7uMqimi/DeYePnO2CzAX58tGwg5y/bTViHOagSj1bS3szm
mQ5DXi11j9Yd/Y5oIf2r/bPKQeQLsIGhtHvXDvoNLMS+AaS4W02C1e6+nB4z81Ft
OhsCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEAQ4H28GPlmP4gjVfMkTnAiufLnVrk
gLBeJP2DSEUovQuKzzcy161xLTPsYwe0xknGjI+R1tALcaE2WgDKV6ZXyx0j0EBg
tTwB7gjNlWRQ9GFl3xNM2zLahOblMzyHdvF4S7BrGCsMU9QOfW6loOMi6AO1NAJQ
V88xUFS9fG9basP1YBm7ff10iRJnDbkzM7gQ118Z3qLS4L0cKDW24
Ms/OlXUUIv6JVj7MYAT55tocDjqmG88UcbbSUUFdHPqPKicsx3Z0HyFfip75Uden
NGTosJNSY9aojTIZYXR9VJm5XYADAejYE+cmdPuChQmOt6DSJYIE/+jtqg==
-----END CERTIFICATE-----
~/Downloads/SEM 5/Micro Services/prac9/key.pem
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
TERMINAL
SEARCH TERMINAL OUTPUT
zsh
~/Downloads/SEM 5/Micro Services/prac9
main !4 ?7
11:25:55 AM
node secureapi2
Server Started on PORT 5000
^C
INT
1m 34s
11:27:33 AM
OUTLINE
TIMELINE
MYSQL

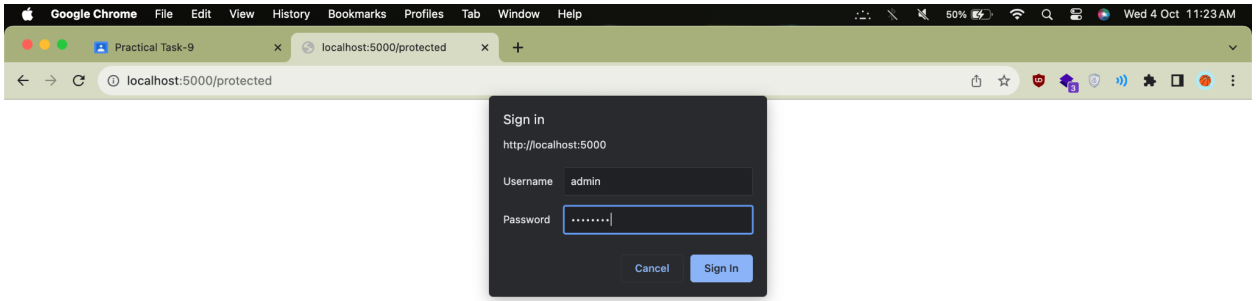
Practical 9.2: Connect it with the website using the HTTPS library

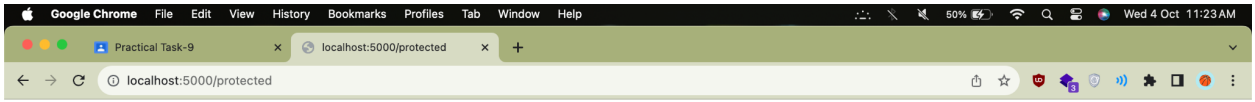




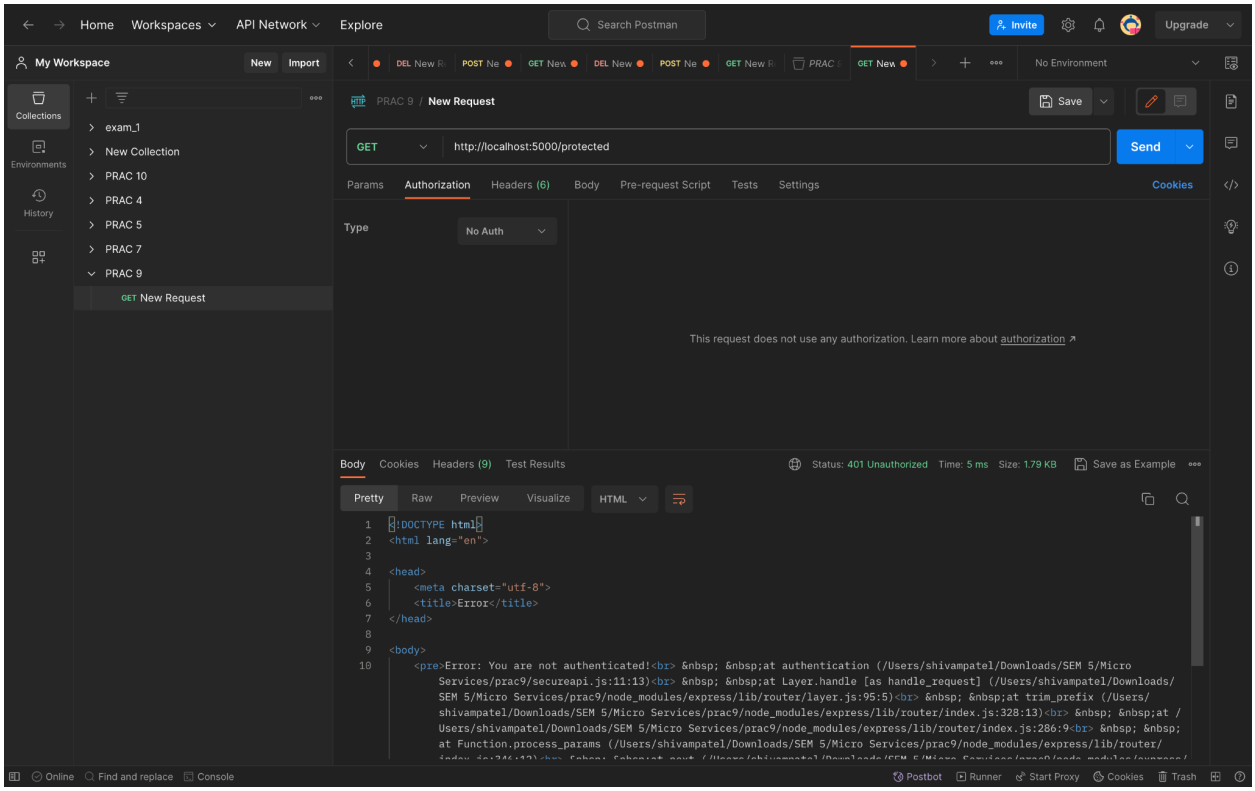
Practical 9.3: Secure application using Username and password.

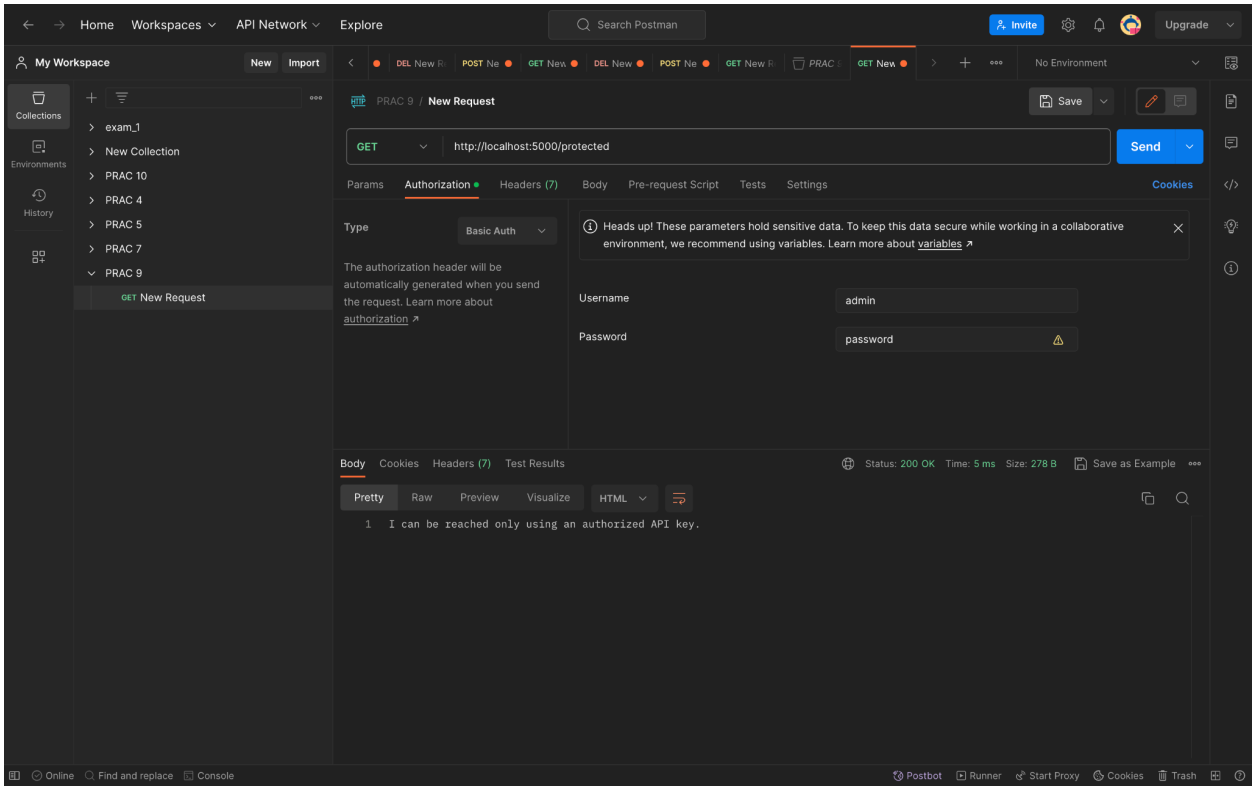




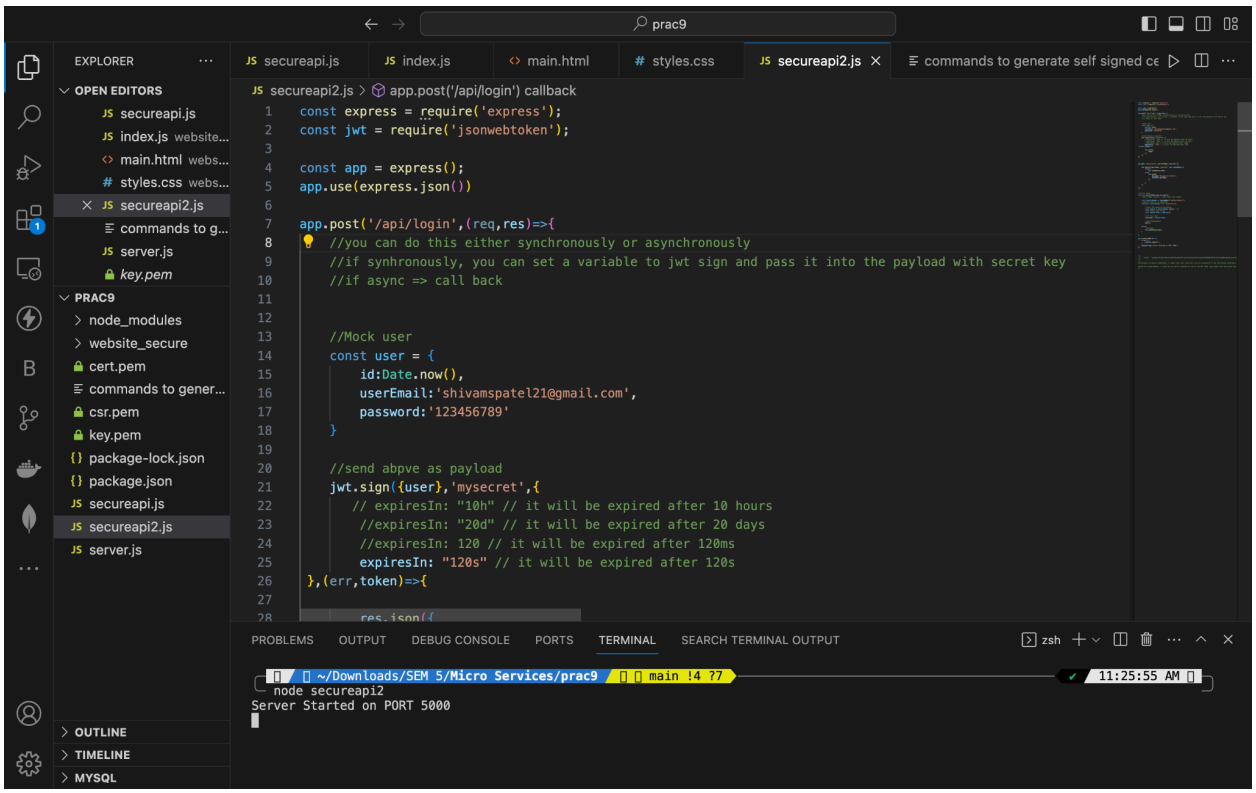


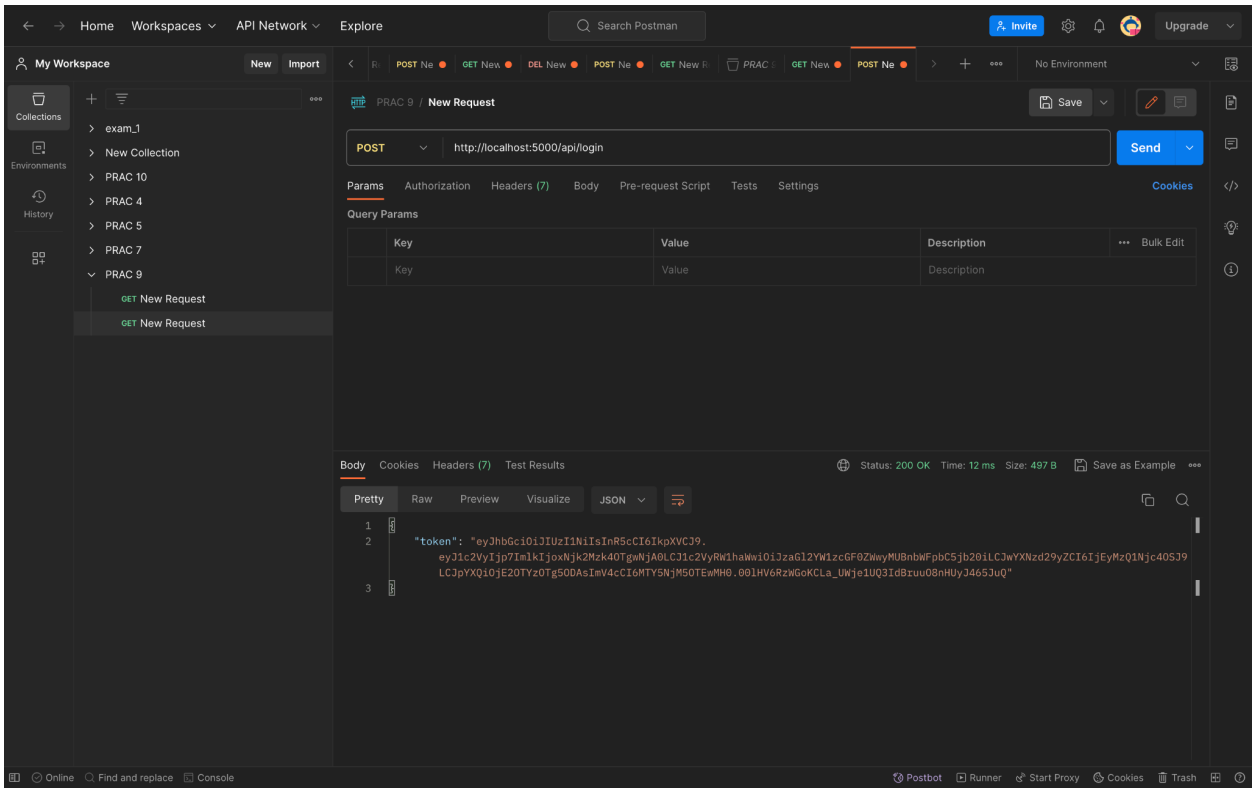
I can be reached only using an authorized API key.

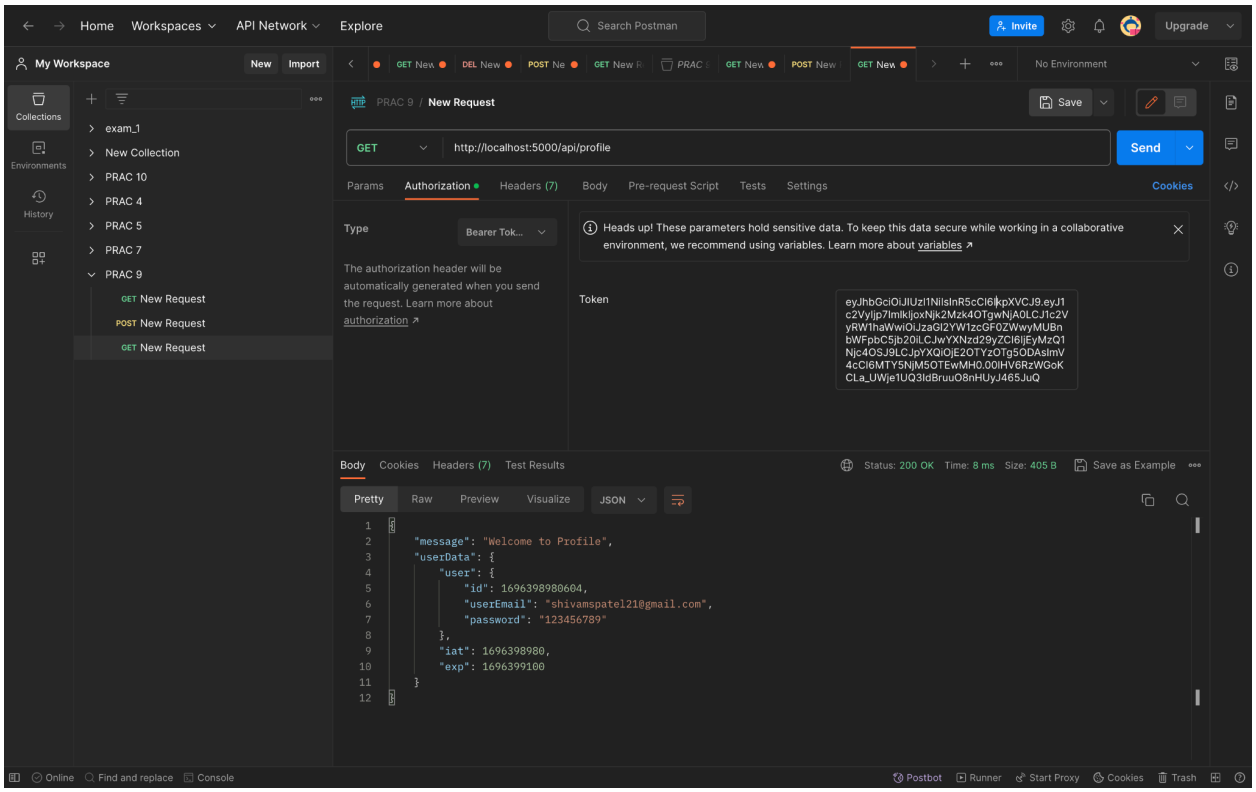




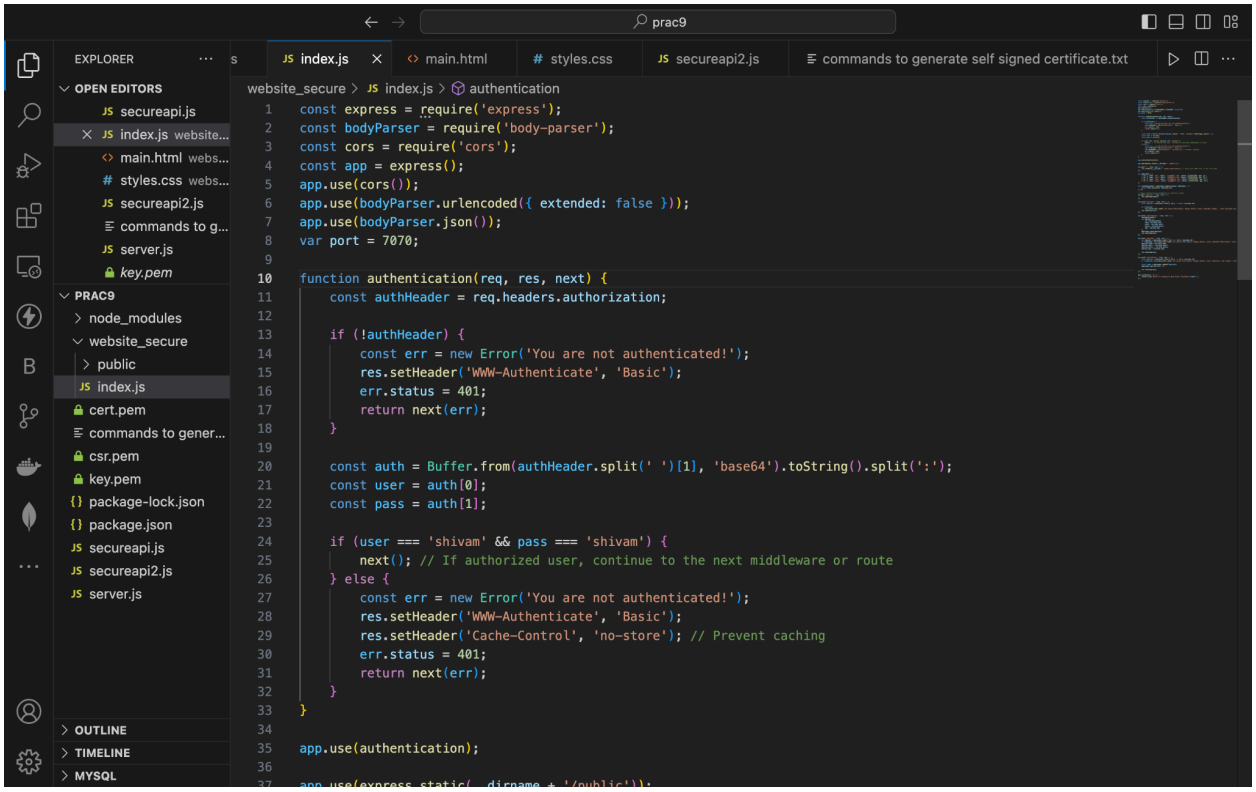
Practical 9.4: Secure your REST APIs using the Bearer token technique.



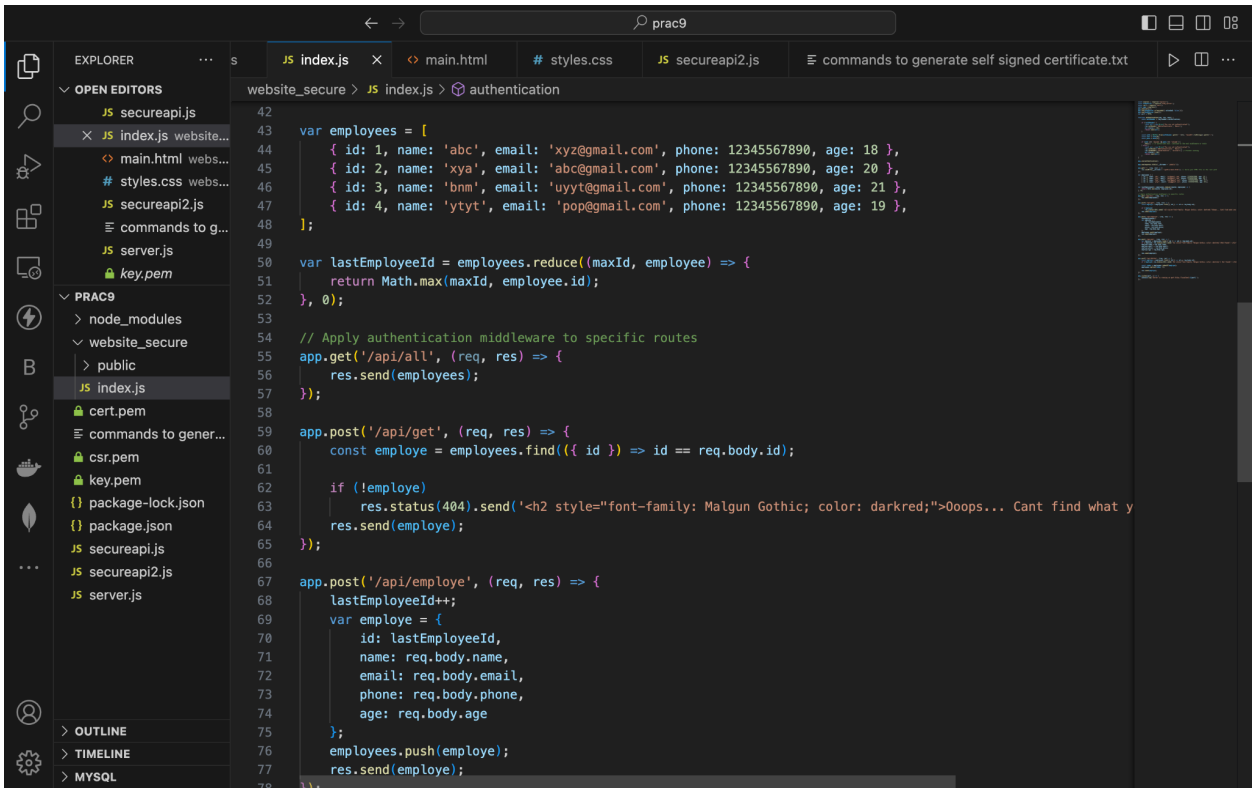

Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
exam_1
New Collection
PRAC 10
PRAC 4
PRAC 5
PRAC 7
PRAC 9
GET New Request
GET New Request
No Environment
PRAC 9 / New Request
Save
POST
http://localhost:5000/api/login
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 12 ms
Size: 497 B
Save as Example
Pretty
Raw
Preview
Visualize
JSON
{
"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ1c2VyIjp7ImlkIjoxNjk2Mzk4OTgwNjA0LCJ1c2VyRW1haWwiOiJzaGl2YW1zcGF0ZWwyMUBnbWFpbC5jb20iLCJwYXNzd29yZCI6IjEyMzQ1Njc4OSJ9
LCJpYXQiOjE2OTYzOTg5ODAsImV4cCI6MTY5NjM5OTEwMH0.00lHV6RzWGoKCLa_UWje1UQ3IdBruuO8nHUyJ465JuQ"
}
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
exam_1
New Collection
PRAC 10
PRAC 4
PRAC 5
PRAC 7
PRAC 9
GET New Request
POST New Request
GET New Request
No Environment
PRAC 9 / New Request
Save
GET
http://localhost:5000/api/profile
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Type
Bearer Tok...
The authorization header will be automatically generated when you send the request. Learn more about authorization
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables
Token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImlkIjoxNjk2Mzk4OTgwNjA0LCJ1c2VyRW1haWwiOiJzaGl2YW1zcGF0ZWwyMUBnbWFpbC5jb20iLCJwYXNzd29yZCI6IjEyMzQ1Njc4OSJ9LCJpYXQiOjE2OTYzOTg5ODAsImV4cCI6MTY5NjM5OTEwMH0.00lHV6RzWGoKCLa_UWje1UQ3IdBruuO8nHUyJ465JuQ
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 8 ms
Size: 405 B
Save as Example
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Welcome to Profile",
"userData": {
"user": {
"id": 1696398980604,
"userEmail": "shivamspatel21@gmail.com",
"password": "123456789"
},
"iat": 1696398980,
"exp": 1696399100
}
}
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash

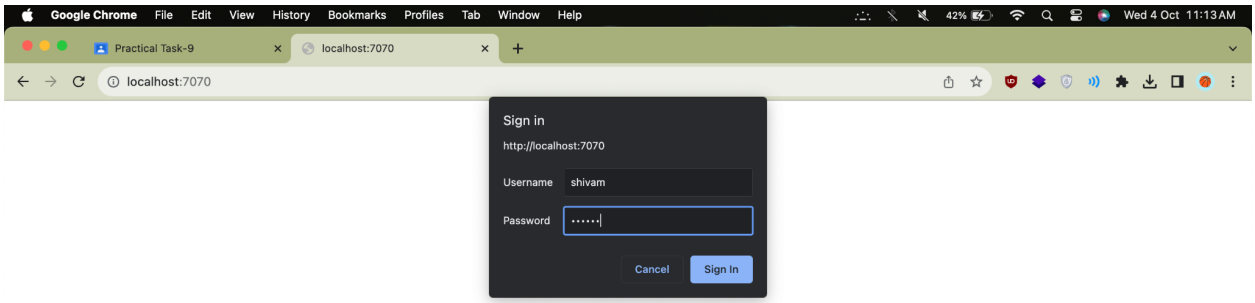
Practical 9.5: Secure your REST APIs created in previous practicals using username and password.

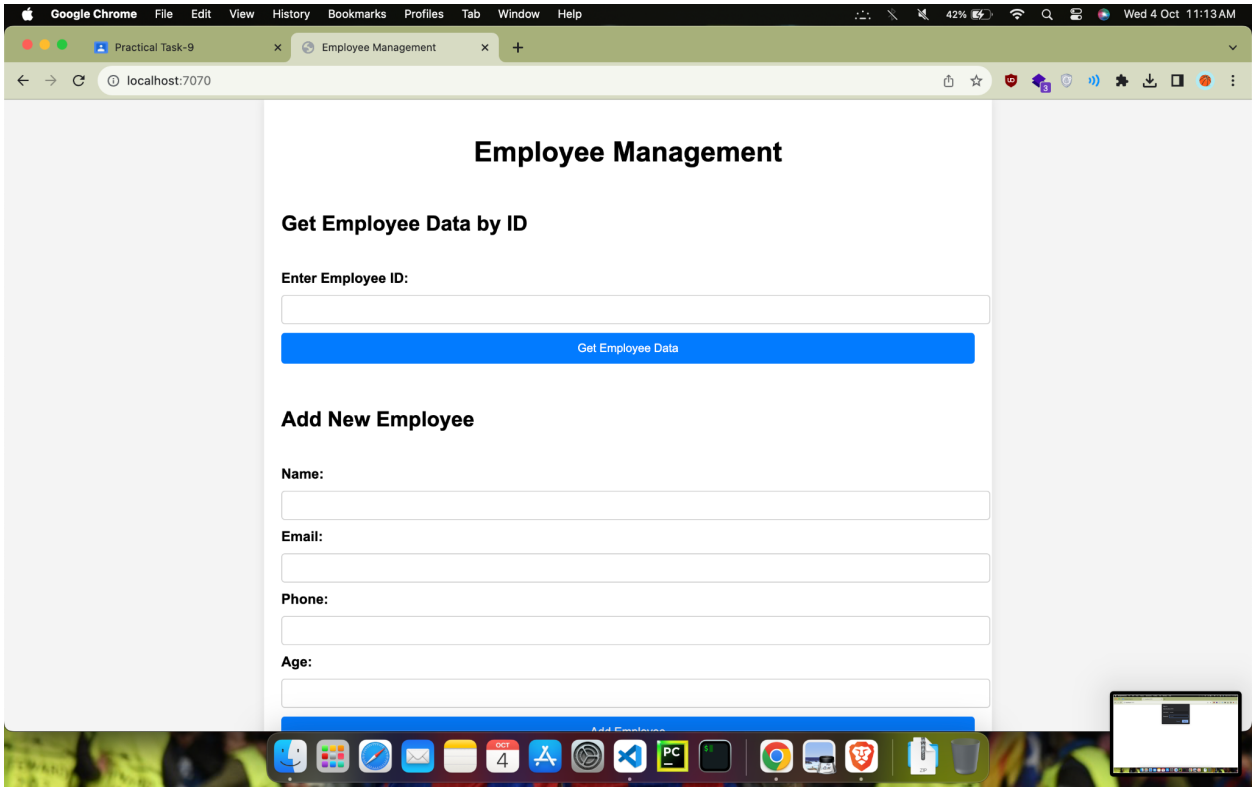


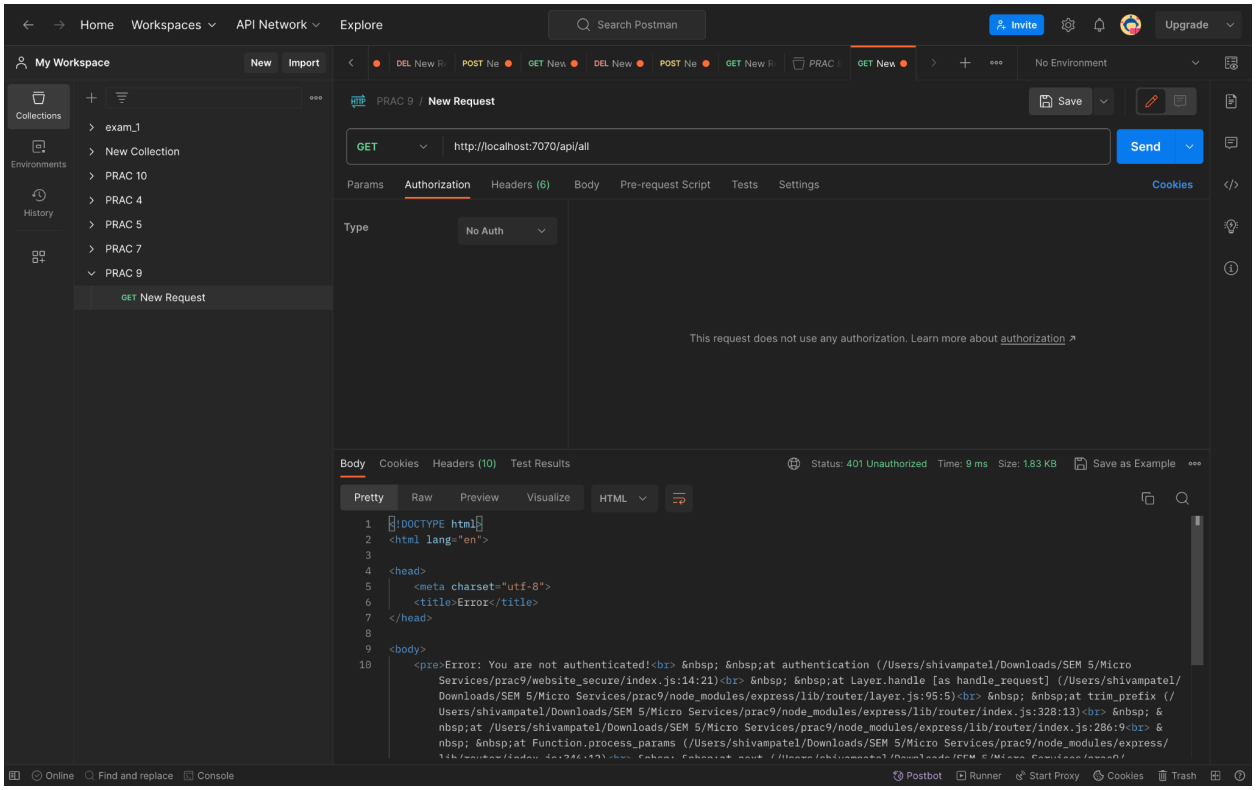
```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');
const app = express();
app.use(cors());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());
var port = 7070;

function authentication(req, res, next) {
    const authHeader = req.headers.authorization;

    if (!authHeader) {
        const err = new Error('You are not authenticated!');
        res.setHeader('WWW-Authenticate', 'Basic');
        err.status = 401;
        return next(err);
    }

    const auth = Buffer.from(authHeader.split(' ')[1], 'base64').toString().split(':');
    const user = auth[0];
    const pass = auth[1];

    if (user === 'shivam' && pass === 'shivam') {
        next(); // If authorized user, continue to the next middleware or route
    } else {
        const err = new Error('You are not authenticated!');
        res.setHeader('WWW-Authenticate', 'Basic');
        res.setHeader('Cache-Control', 'no-store'); // Prevent caching
        err.status = 401;
        return next(err);
    }
}

app.use(authentication);
```

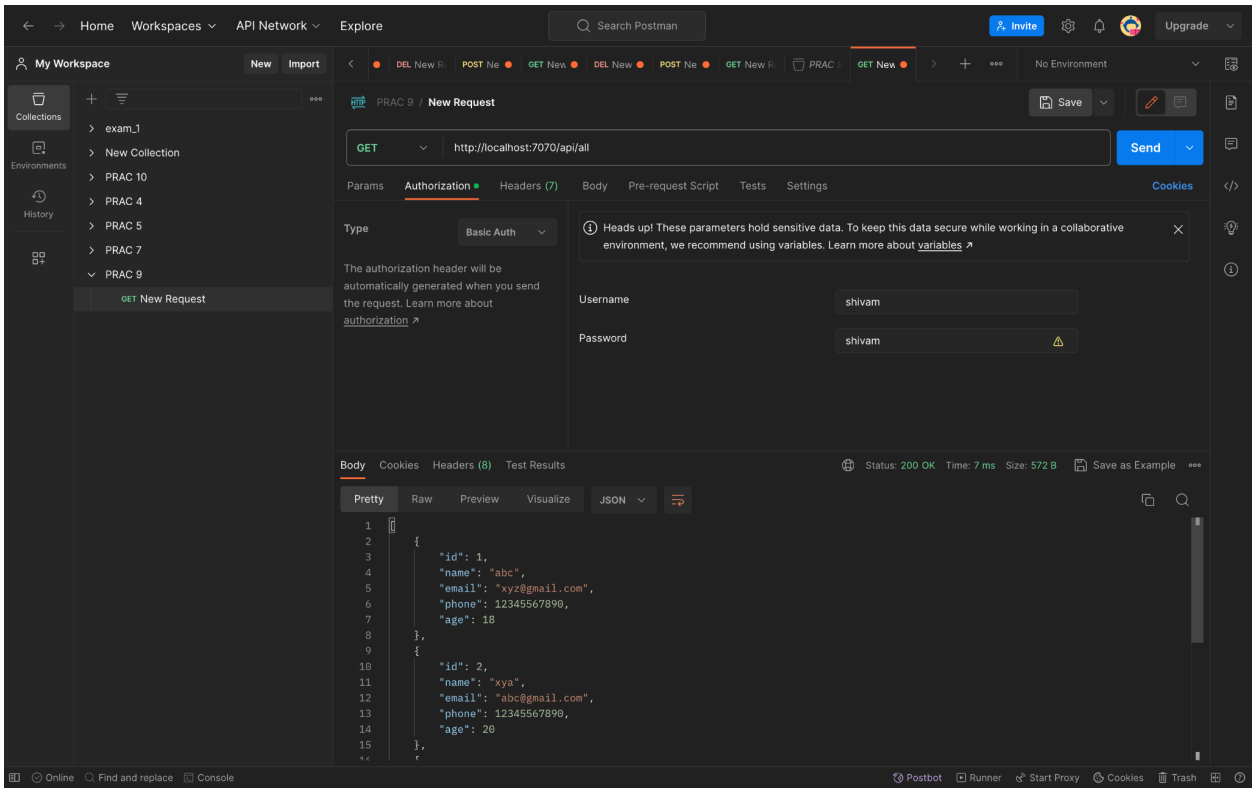


```
var employees = [
    { id: 1, name: 'abc', email: 'xyz@gmail.com', phone: 12345567890, age: 18 },
    { id: 2, name: 'xya', email: 'abc@gmail.com', phone: 12345567890, age: 20 },
    { id: 3, name: 'bnm', email: 'uyyt@gmail.com', phone: 12345567890, age: 21 },
    { id: 4, name: 'ytyt', email: 'pop@gmail.com', phone: 12345567890, age: 19 },
];

var lastEmployeeId = employees.reduce((maxId, employee) => {
    return Math.max(maxId, employee.id);
}, 0);

// Apply authentication middleware to specific routes
app.get('/api/all', (req, res) => {
    res.send(employees);
});

app.post('/api/get', (req, res) => {
    const employe = employees.find(({ id }) => id == req.body.id);

    if (!employe)
        res.status(404).send('<h2 style="font-family: Malgun Gothic; color: darkred;">Ooops... Cant find what y
    res.send(employe);
});

app.post('/api/employe', (req, res) => {
    lastEmployeeId++;
    var employe = {
        id: lastEmployeeId,
        name: req.body.name,
        email: req.body.email,
        phone: req.body.phone,
        age: req.body.age
    };
    employees.push(employe);
    res.send(employe);
```


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 4 Oct 11:13 AM
Practical Task-9
localhost:7070
localhost:7070
Sign in
http://localhost:7070
Username
shivam
Password
Cancel
Sign In


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 4 Oct 11:13 AM
Practical Task-9
Employee Management
localhost:7070
Employee Management
Get Employee Data by ID
Enter Employee ID:
Get Employee Data
Add New Employee
Name:
Email:
Phone:
Age:


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
exam_1
New Collection
PRAC 10
PRAC 4
PRAC 5
PRAC 7
PRAC 9
GET New Request
No Environment
PRAC 9 / New Request
Save
GET
http://localhost:7070/api/all
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Type
No Auth
This request does not use any authorization. Learn more about authorization
Body
Cookies
Headers (10)
Test Results
Status: 401 Unauthorized
Time: 9 ms
Size: 1.83 KB
Save as Example
Pretty
Raw
Preview
Visualize
HTML
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Error: You are not authenticated!
    at authentication (/Users/shivampatel/Downloads/SEM 5/Micro Services/prac9/website_secure/index.js:14:21)
    at Layer.handle [as handle_request] (/Users/shivampatel/Downloads/SEM 5/Micro Services/prac9/node_modules/express/lib/router/layer.js:95:5)
    at trim_prefix (/Users/shivampatel/Downloads/SEM 5/Micro Services/prac9/node_modules/express/lib/router/index.js:328:13)
    at /Users/shivampatel/Downloads/SEM 5/Micro Services/prac9/node_modules/express/lib/router/index.js:286:9
    at Function.process_params (/Users/shivampatel/Downloads/SEM 5/Micro Services/prac9/node_modules/express/
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
Environments
History
exam_1
New Collection
PRAC 10
PRAC 4
PRAC 5
PRAC 7
PRAC 9
GET New Request
No Environment
PRAC 9 / New Request
Save
GET
http://localhost:7070/api/all
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Type
Basic Auth
The authorization header will be automatically generated when you send the request. Learn more about authorization
Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables
Username
shivam
Password
shivam
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 7 ms
Size: 572 B
Save as Example
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"name": "abc",
"email": "xyz@gmail.com",
"phone": 12345567890,
"age": 18
},
{
"id": 2,
"name": "xya",
"email": "abc@gmail.com",
"phone": 12345567890,
"age": 20
},
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash