Name : Avinash
Roll number : 21011102019

Routing Implementation using ExpressJS

server/index.js:

```javascript
const http = require('http');
const express = require('express');
const bodyParser = require('body-parser');
// const server = http.createServer((req, res) => {
//    res.writeHead(200, { 'Content-Type': 'text/plain' });
//    res.end('Hello, Node.js Server!\n');
// });

const PORT = process.env.PORT || 3001;
const app = express();
app.use(bodyParser.json());
app.use(express.json());
app.use(express.urlencoded({ extended: false }));


app.get('/', (req, res) => {
  res.send('this is root endpoint')
})

app.post('/', (req, res) => {
  console.log(req.body)
  res.send('POST request to the homepage')
})




app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
// server.listen(PORT, () => {
//    console.log(`Server is running on port ${PORT}`);
// });
```
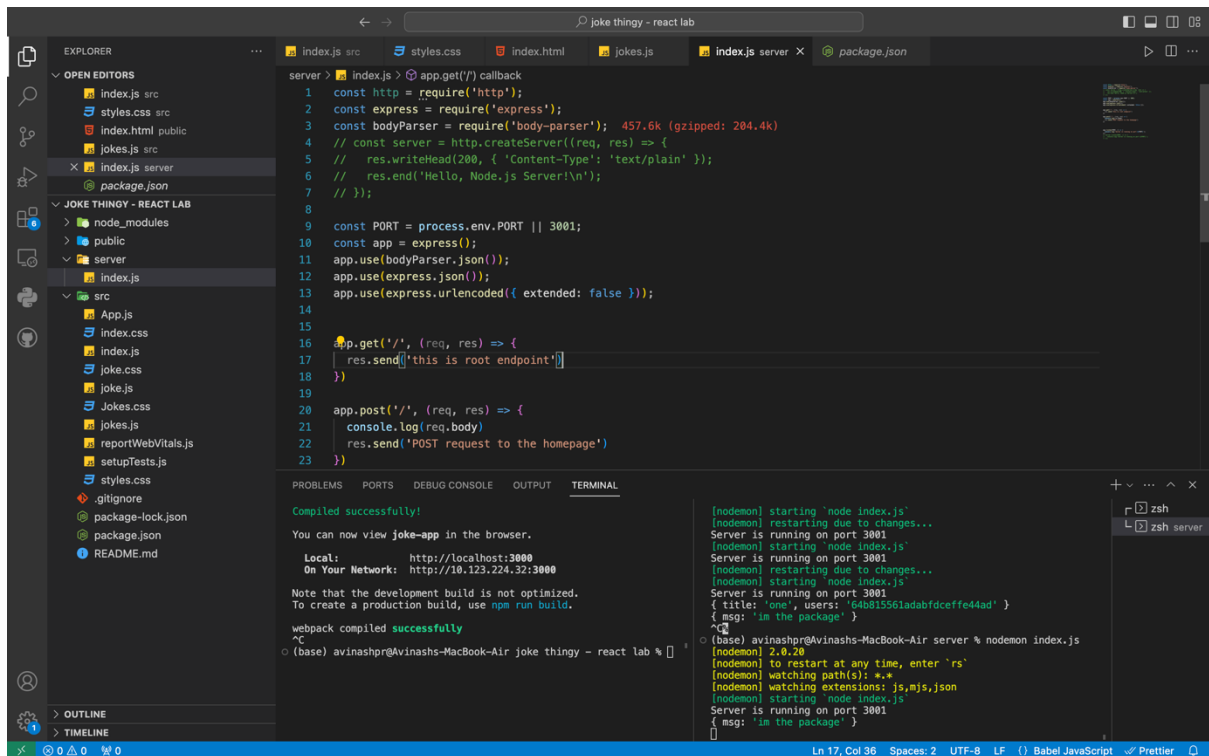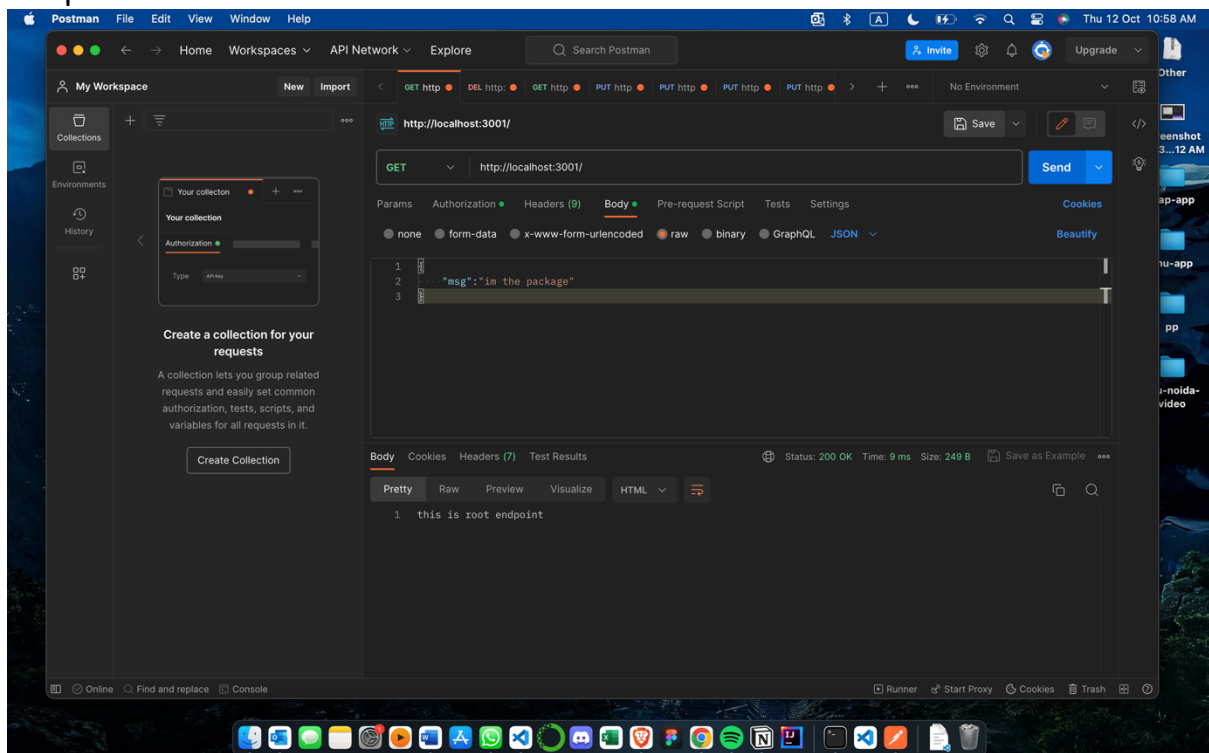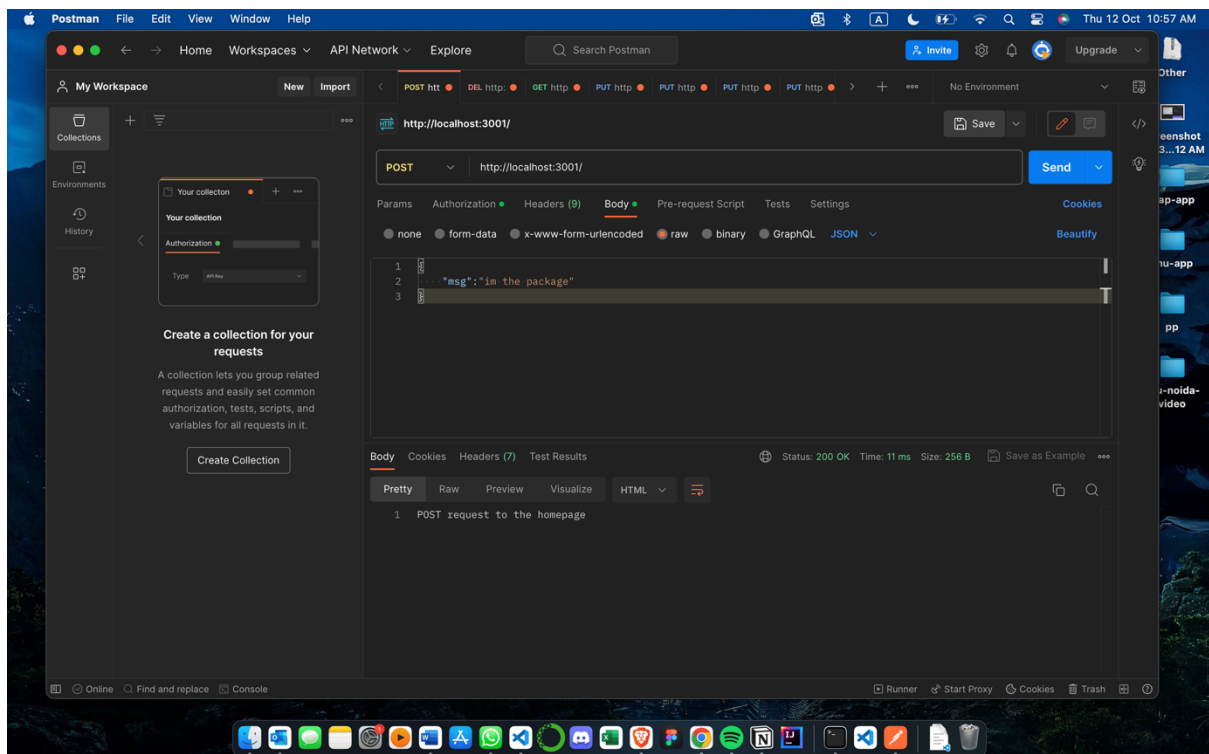
Output in Vs code terminal:

Output for GET in localhost:3001

Output for POST in localhost:3001

My Workspace

Collections

Environments

History

POST http://localhost:3001/

POST http://localhost:3001/ Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1
2    "msg":"im the package"
3
```

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 11 ms Size: 256 B Save as Example

Pretty Raw Preview Visualize HTML

```
1  POST request to the homepage
```

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

```
(base) avinashpr@Avinashs-MacBook-Air server % nodemon index.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server is running on port 3001
{ msg: 'im the package' }
```