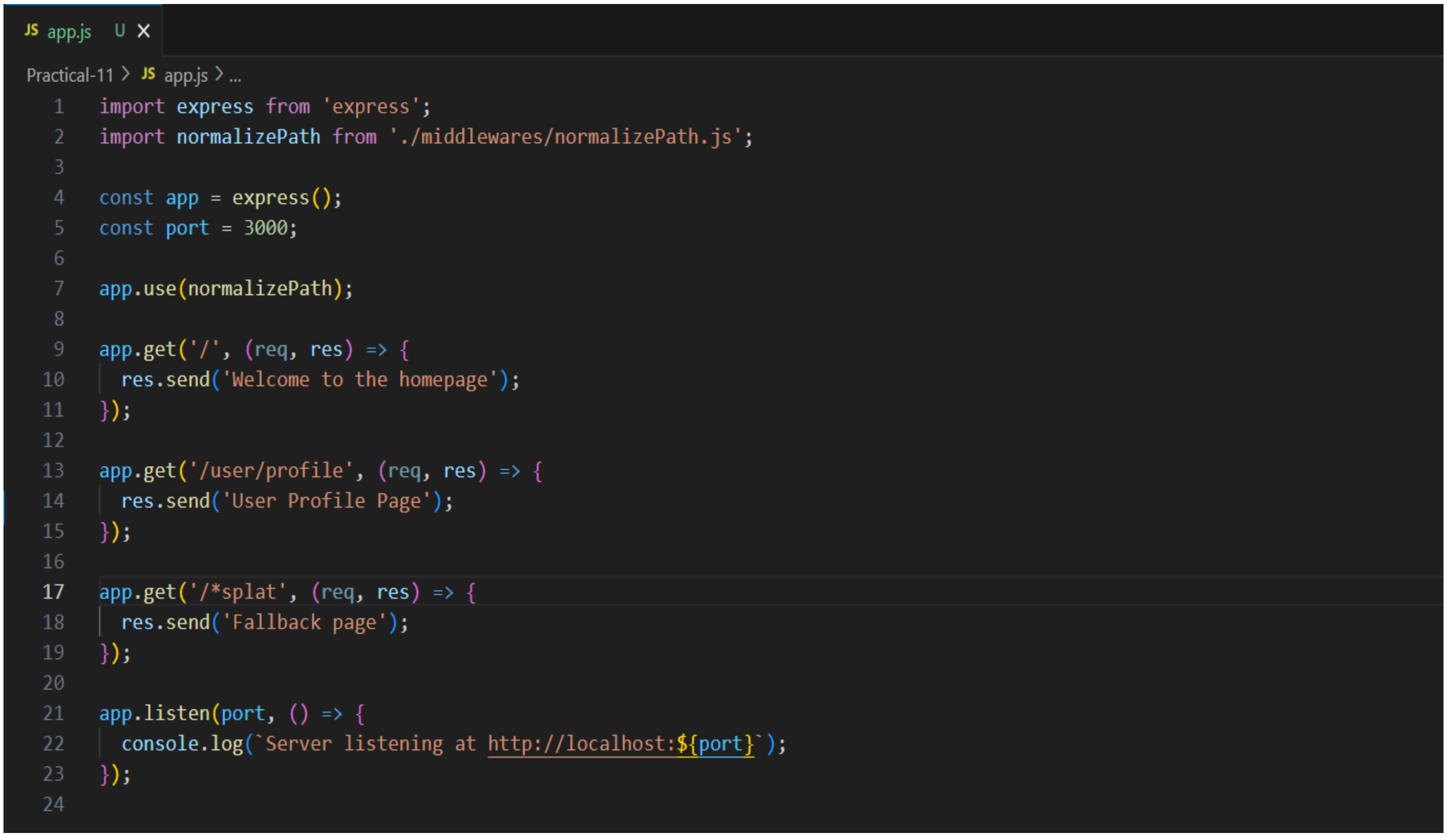
Practical – 11

GitHub Link: https://github.com/Devsharma511/NodeJs.git

Create the website and track the patterns. If there is redundancy, resolve it using the different path module and if no redundancy, which path module you will be executing and why? Answer with the code and along with your opinion.

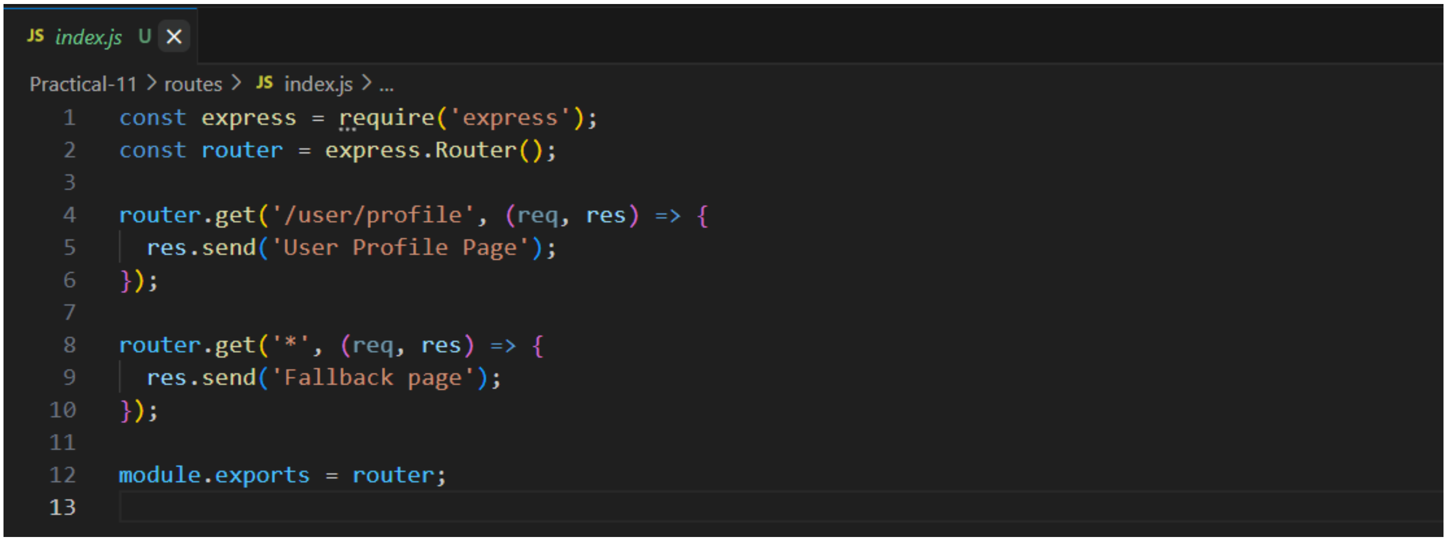
Project Structure:

app.js:



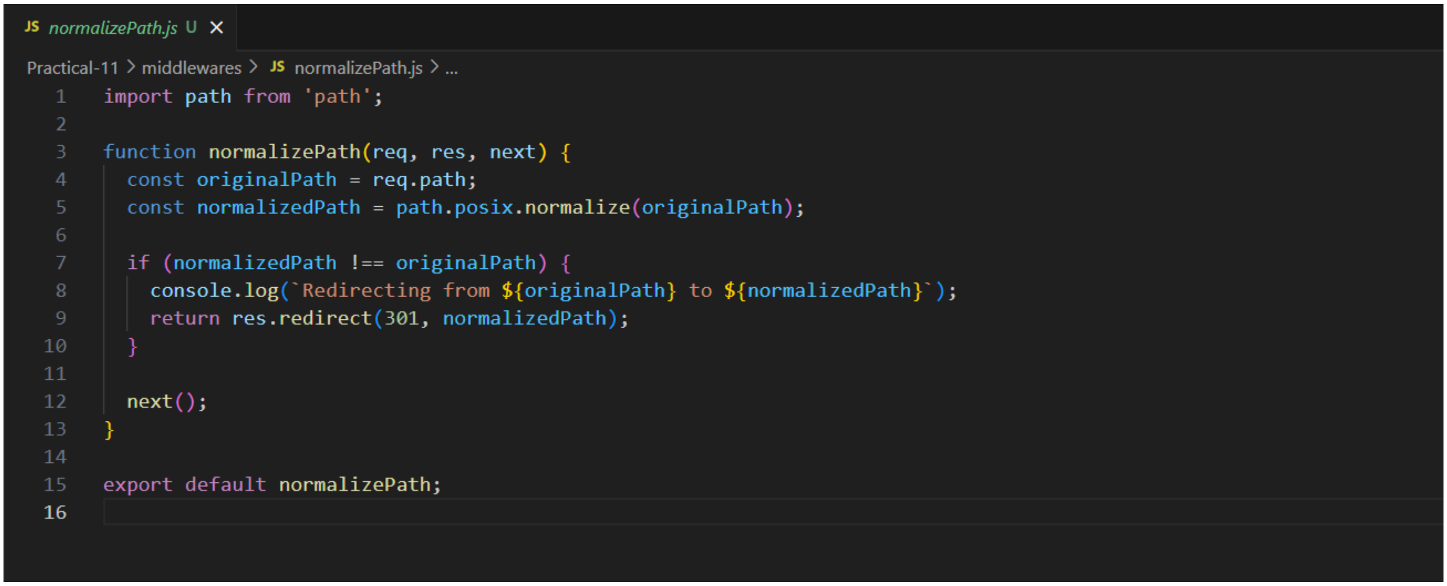
```javascript
import express from 'express';
import normalizePath from './middlewares/normalizePath.js';

const app = express();
const port = 3000;

app.use(normalizePath);

app.get('/', (req, res) => {
  res.send('Welcome to the homepage');
});

app.get('/user/profile', (req, res) => {
  res.send('User Profile Page');
});

app.get('/*splat', (req, res) => {
  res.send('Fallback page');
});

app.listen(port, () => {
  console.log(`Server listening at http://localhost:${port}`);
});

```

routes/index.js:



```
const express = require('express');
const router = express.Router();

router.get('/user/profile', (req, res) => {
  res.send('User Profile Page');
});

router.get('*', (req, res) => {
  res.send('Fallback page');
});

module.exports = router;

```

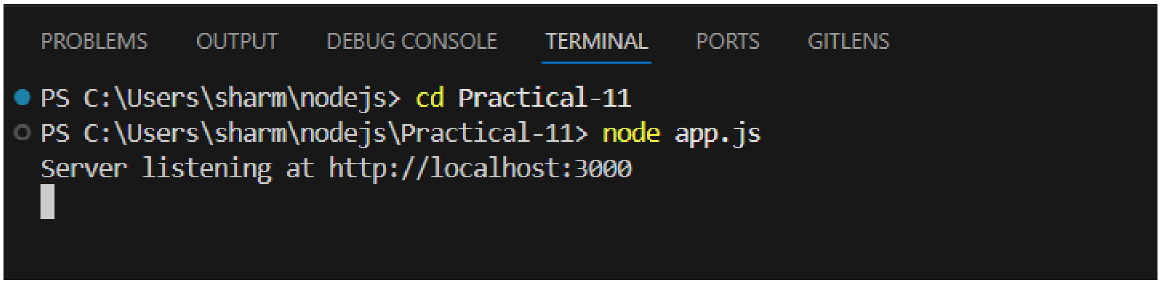
middlewares/normalizePath.js:



```
import path from 'path';

function normalizePath(req, res, next) {
  const originalPath = req.path;
  const normalizedPath = path.posix.normalize(originalPath);

  if (normalizedPath !== originalPath) {
    console.log(`Redirecting from ${originalPath} to ${normalizedPath}`);
    return res.redirect(301, normalizedPath);
  }

  next();
}

export default normalizePath;

```

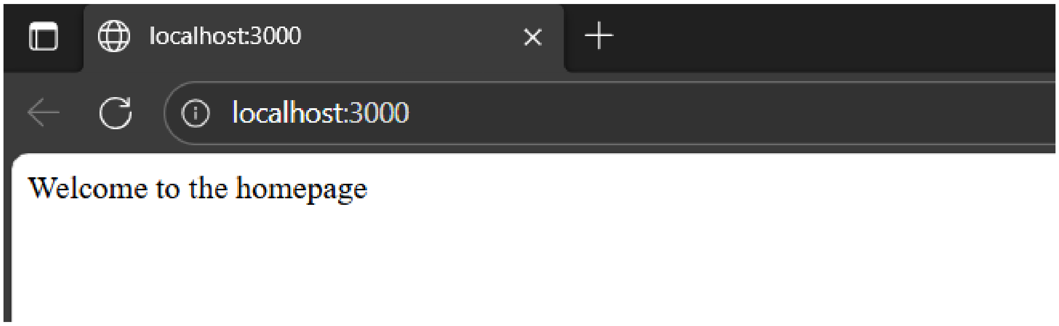
Output:

Terminal:



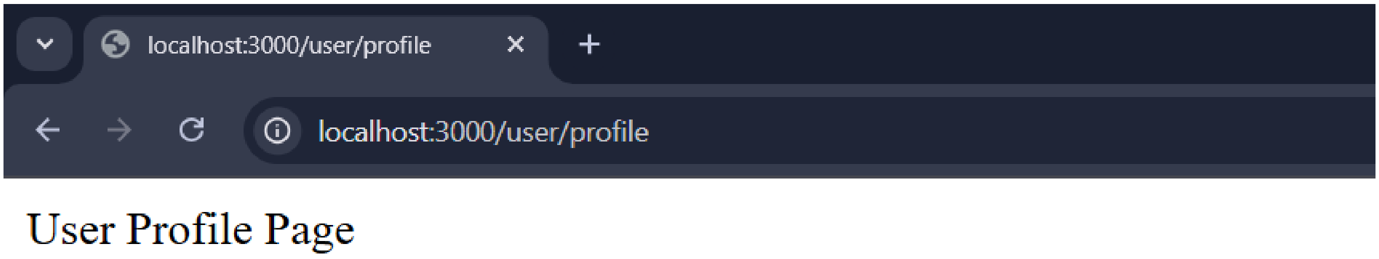
```
PS C:\Users\sharm\nodejs> cd Practical-11
PS C:\Users\sharm\nodejs\Practical-11> node app.js
Server listening at http://localhost:3000
```

Homepage (localhost:3000):



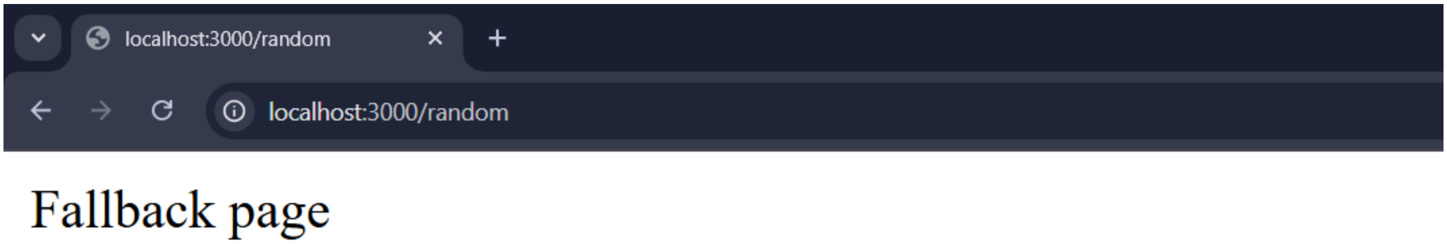
Welcome to the homepage

User Profile (localhost:3000/user/profile):



User Profile Page

Visiting any undefined route like (http://localhost:3000/random) shows:



Fallback page

Opinion on Path Module Usage:

- For resolving URL path redundancies, path.posix.normalize() is used as URLs always use POSIX-style forward slashes (/). This method efficiently removes redundant segments like multiple slashes.
- When there is no redundancy, path.posix is used for consistency and cross-platform compatibility since URLs follow POSIX path conventions regardless of the operating system.
- Modules like path.resolve() are mainly for filesystem paths and can produce incorrect results for URLs, so avoid them for URL path manipulation.

Explanation:

- The Node.js path module is a built-in utility that handles file and URL paths, abstracting differences between Windows and Unix-style separators.
- The path.posix variant provides POSIX-compliant path functions suitable for URLs, ensuring consistent forward-slash use.
- The middleware uses this to normalize every incoming URL path, redirecting redundant URLs to their canonical form, improving SEO and user experience.