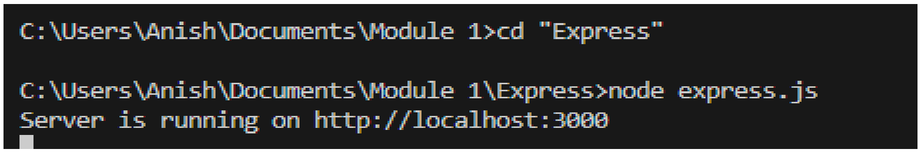
Practical No : 12

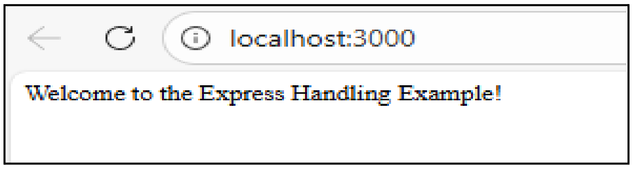
AIM : Write a program to demonstrate error handling in Express.js

```
const express = require ("express");
const app = express();

app.use(express.json());

// Normal route
app.get("/", (req, res) => {
  res.send("Welcome to the Express Handling Example!");
});
// Route that throws an error
app.get("/error", (req, res, next) => {
  //Create an intentional error
  throw new Error("Something went wrong on the server!");
});
// Error-handling middleware (must have 4 args)
app.use((err, req, res, next) => {
  console.error("Error Message:", err.message);
  res.status(500).send({
    success:false,
    message: "Internal Server Error",
    error: err.message
  });
});
app.listen(3000, () => {
  console.log("Server is running on http://localhost:3000");
});
```

Output:-



```
C:\Users\Anish\Documents\Module 1>cd "Express"

C:\Users\Anish\Documents\Module 1\Express>node express.js
Server is running on http://localhost:3000
```



Welcome to the Express Handling Example!

Practical No : 13

AIM : Write a program to demonstrate environment configuration in Express.js using dotenv.

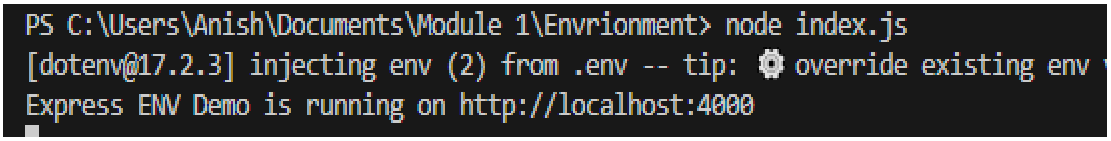
1 index.js

```
require ('dotenv').config(); // Load .env file

const express = require('express');
const app = express();

const PORT = process.env.PORT || 3000;
const APP_NAME = process.env.APP_NAME;

app.get('/', (req, res) => {
  res.send(`Welcome to ${APP_NAME}!Running on port ${PORT}.`);
});

app.listen(PORT, () => {
  console.log(`${APP_NAME} is running on http://localhost:${PORT}`);
});
```

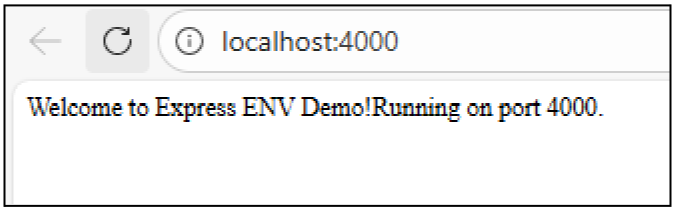
2 . env

```
PORT=4000
APP_NAME= Express ENV Demo
```

Output : -



```
PS C:\Users\Anish\Documents\Module 1\Envrionment> node index.js
[dotenv@17.2.3] injecting env (2) from .env -- tip: ⚙ override existing env
Express ENV Demo is running on http://localhost:4000
```



Welcome to Express ENV Demo!Running on port 4000.