Day 47

"Web Development + Security"

ejs Template Engine in Express:

What is EJS?

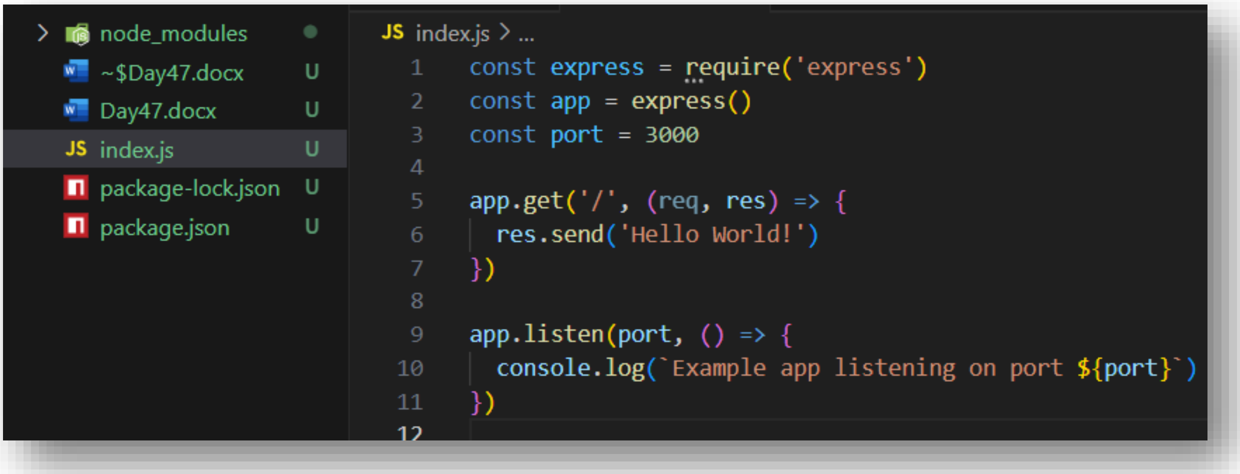
EJS (Embedded JavaScript) is a template engine for Node.js that allows you to generate dynamic HTML using JavaScript.

It lets you:

- Write HTML pages that can include variables, loops, and conditions
- Embed JavaScript code inside HTML using <%= %> or <% %>
- Separate your server logic (Express) from your frontend presentation (HTML)

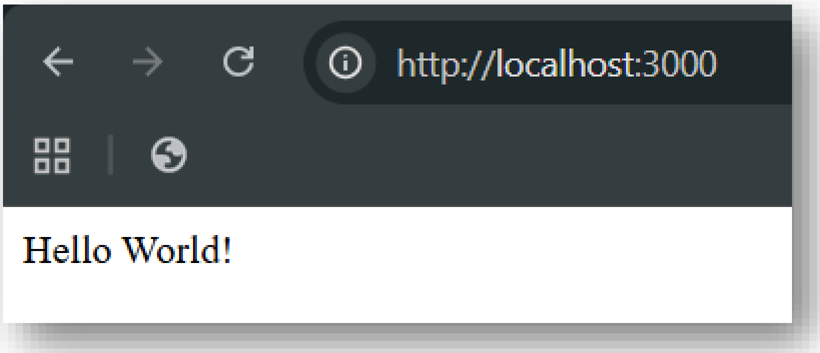
Example: simplest template we are using since last few express js videos

Index.js:



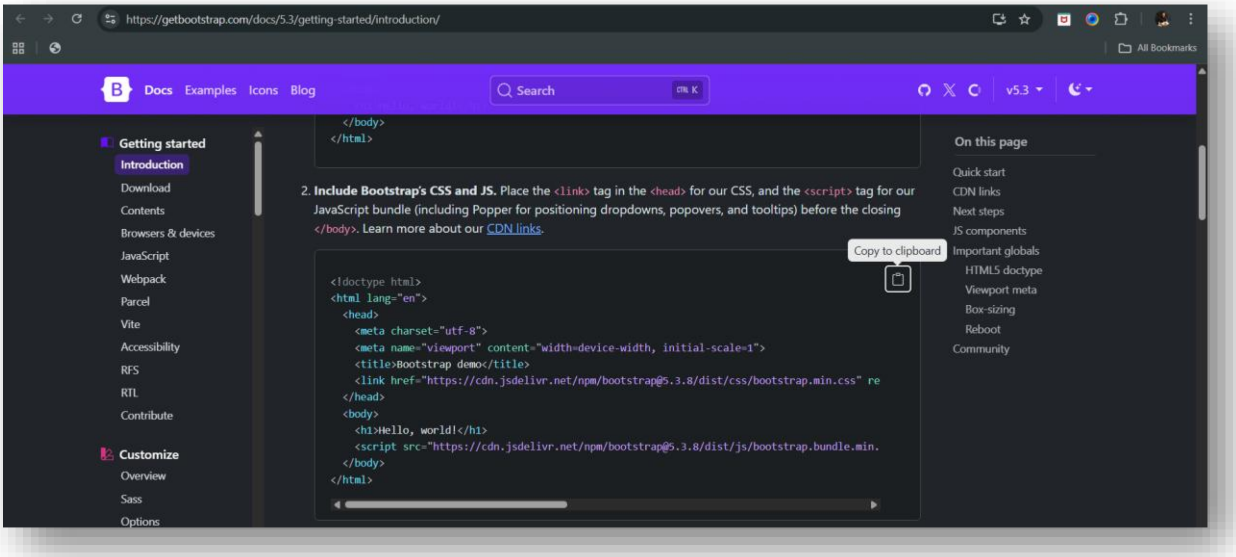
```javascript
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

Output:



Hello World!

Now, we are introducing Bootstrap in the code. For which we will copy the template from the bootstrap docs. Which be this:

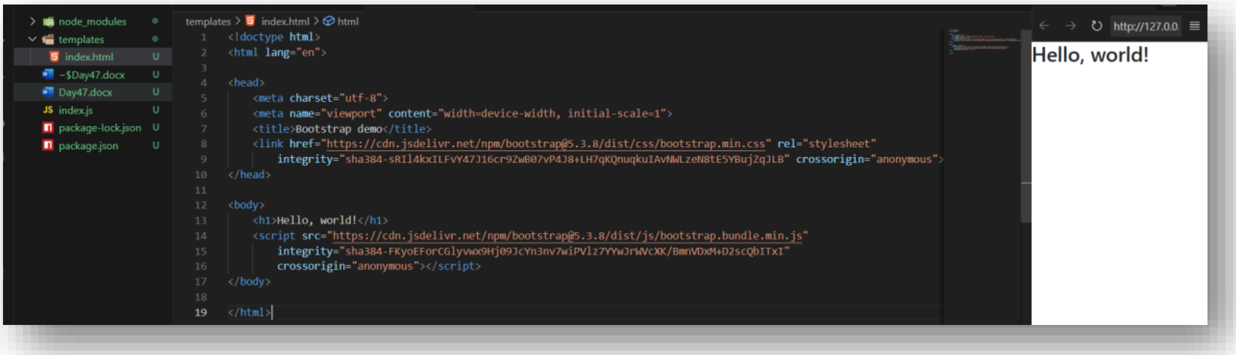


```
  </body>
</html>
```

2. **Include Bootstrap's CSS and JS.** Place the `<link>` tag in the `<head>` for our CSS, and the `<script>` tag for our JavaScript bundle (including Popper for positioning dropdowns, popovers, and tooltips) before the closing `</body>`. Learn more about our CDN links.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" re
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.
  </body>
</html>
```

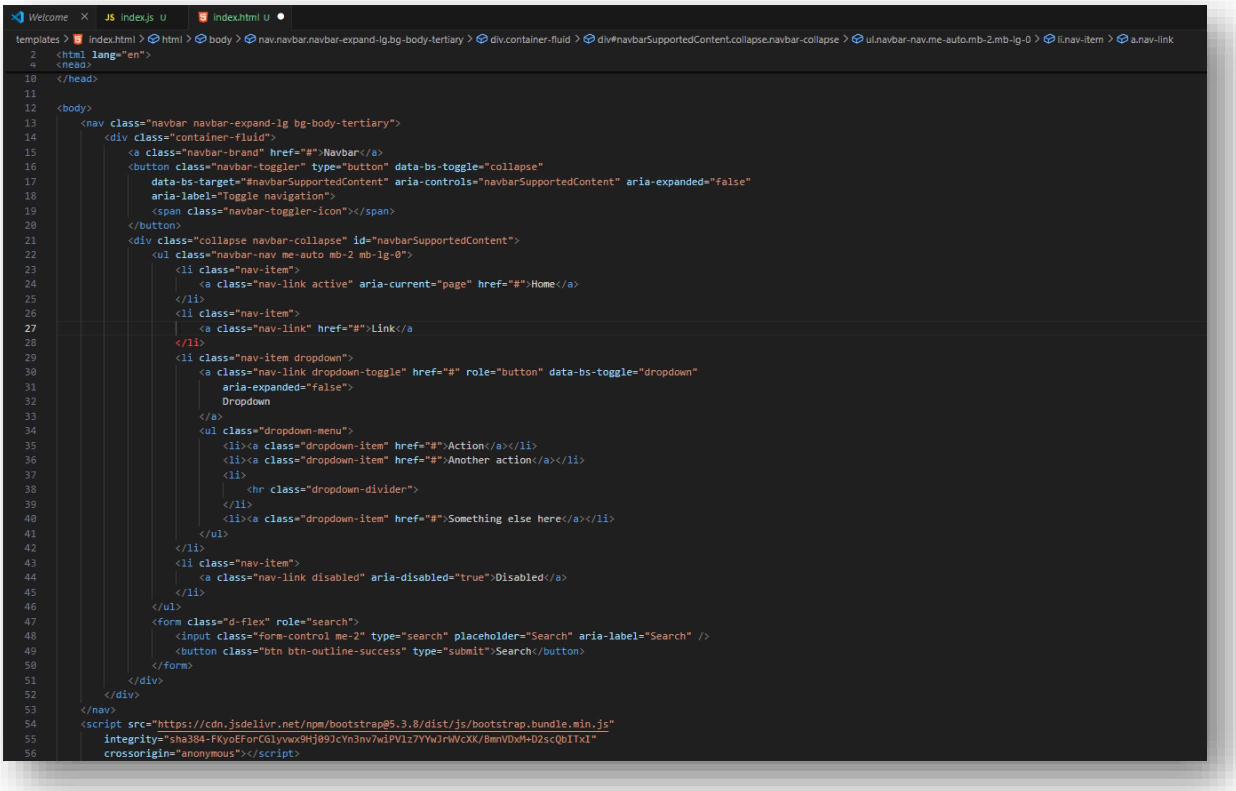
Now, in order to include it, we will create a folder named “templates” and inside I will create an “index.html” page: and just paste the copied code. As shown below:



```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-sRIl4kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkuIAvNWLzeN8tE5YBujZqJLB" crossorigin="anonymous">
</head>

<body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-FKyoEForCGlyvwx9Hj09JcYn3nv7wiPVlz7YYwJrWVcXK/BmnVDxM+D2scQbITxI"
        crossorigin="anonymous"></script>
</body>

</html>
```

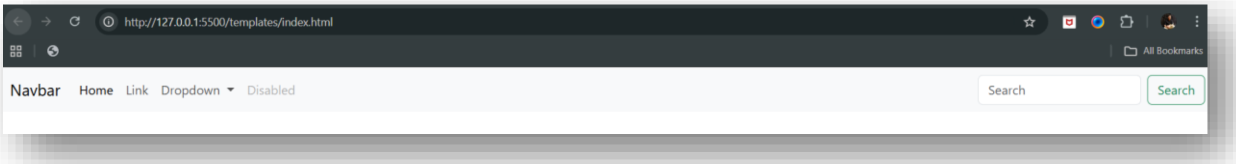
Hello, world!

Also, we can just copy paste the navbar code from the docs of bootstrap and run:


Welcome
JS index.js U
index.html U
templates > index.html > html > body > nav.navbar.navbar-expand-lg.bg-body-tertiary > div.container-fluid > div#navbarSupportedContent.collapse.navbar-collapse > ul.navbar-nav.me-auto.mb-2.mb-lg-0 > li.nav-item > a.nav-link

```html
<html lang="en">
<head>
</head>

<body>
    <nav class="navbar navbar-expand-lg bg-body-tertiary">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">Navbar</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="#">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Link</a
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown"
                            aria-expanded="false">
                            Dropdown
                        </a>
                        <ul class="dropdown-menu">
                            <li><a class="dropdown-item" href="#">Action</a></li>
                            <li><a class="dropdown-item" href="#">Another action</a></li>
                            <li>
                                <hr class="dropdown-divider">
                            </li>
                            <li><a class="dropdown-item" href="#">Something else here</a></li>
                        </ul>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link disabled" aria-disabled="true">Disabled</a>
                    </li>
                </ul>
                <form class="d-flex" role="search">
                    <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search" />
                    <button class="btn btn-outline-success" type="submit">Search</button>
                </form>
            </div>
        </div>
    </nav>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-FKyoEForCGlyvwx9Hj09JcYn3nv7wiPVlz7YYwJrWVcXK/BmnVDxM+D2scQbITxI"
        crossorigin="anonymous"></script>
```

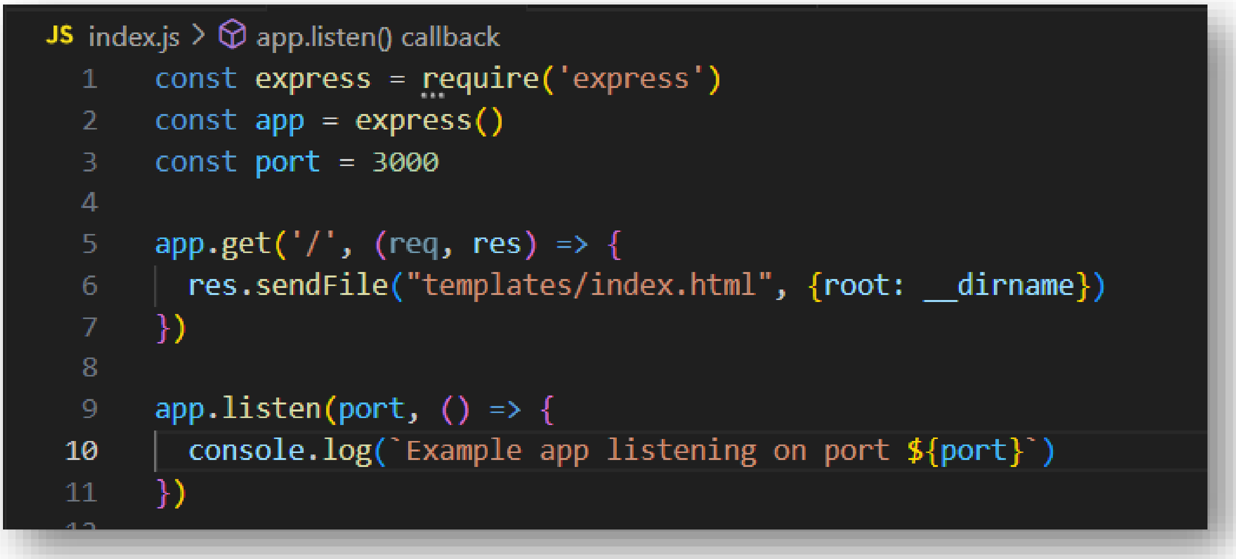
Output: clearly it is on different server.



Now, suppose we want that instead of content of index.js we want that index.html should be shown at the express server, then we will do by including the res.sendFile(). As shown below:

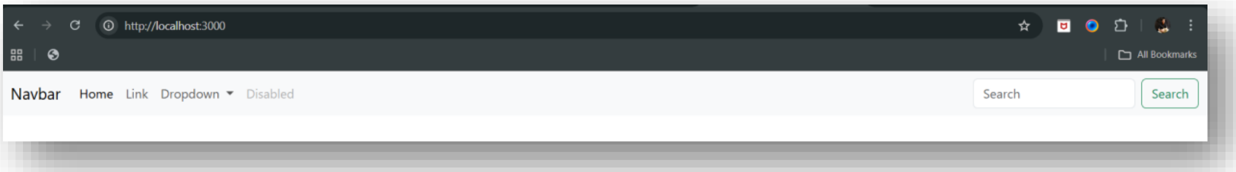
Index.html: as same as above

Index.js:



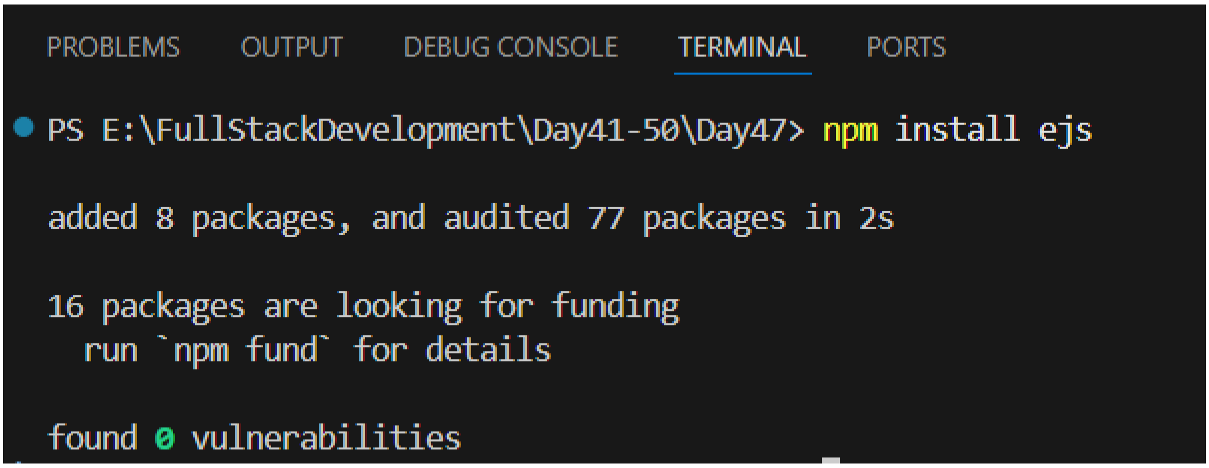
```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.sendFile("templates/index.html", {root: __dirname})
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

Output: clearly now navbar is on the express js server



Now, suppose we fetched something from the database (which we will understand later) then we want that we shall be able to make it enter in the template. So, does this means we have to make index.html for every page? No. We will use EJS here.

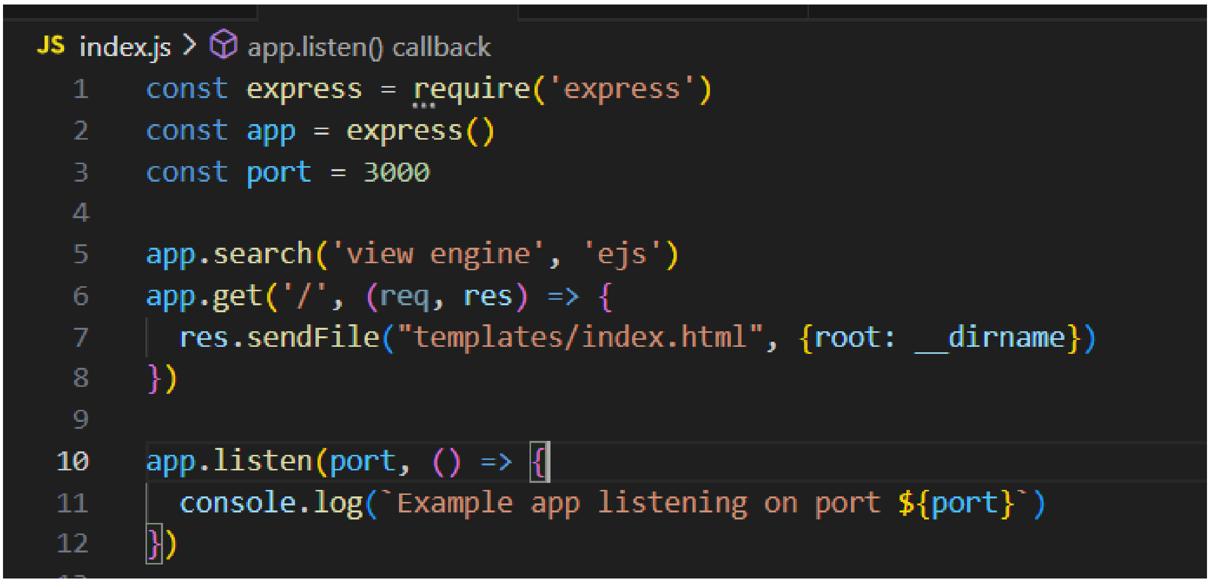
Now, to install ejs we will use the command: npm install ejs



```
PS E:\FullStackDevelopment\Day41-50\Day47> npm install ejs

added 8 packages, and audited 77 packages in 2s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Now, to specify that we want to use ejs template engine, we will write the code:

app.set('view engine', 'ejs');



```
const express = require('express')
const app = express()
const port = 3000

app.search('view engine', 'ejs')
app.get('/', (req, res) => {
  res.sendFile("templates/index.html", {root: __dirname})
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

But is this enough? No, we need to make some major changes in the index.js such that it will start sending the content to the index.html.

---The End--