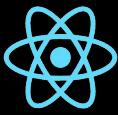


React

node > air-bnb > app.js > ...

```
1 // External Modules
2 const express = require('express');
3 // Local Modules
4 const hostRouter = require('./routes/host');
5 const userRouter = require('./routes/user');
6
7 const app = express();
8
9 app.use(express.urlencoded({extended:true}));
10 app.use(hostRouter);
11 app.use(userRouter);
12
13 const PORT = 3000;
14 app.listen(PORT, () => {
15   | console.log(`Server running at http://localhost:${PORT}/` );
16 });
```

EXPRESS



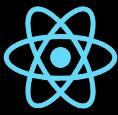
React

10.2 Express Router

EXPRESS Js

node > air-bnb > routes > host.js > ...

```
1 // External Modules
2 const express = require('express');
3
4 const router = express.Router();
5
6 router.get('/add-home', (req, res, next) => {
7   res.send(`
8     <h1>Register Your Home on AirBnB</h1>
9     <form action="/submit-home" method="POST">
10       <input type="text" name="houseName" placeholder="Enter your House Name"><br>
11       <br><input type="submit" value="Submit">
12     </form>
13   `);
14 });
15
16 router.post('/submit-home', (req, res, next) => {
17   console.log("Post Request details", req.url, req.method, req.body);
18   res.redirect('/');
19 });
20
21 module.exports = router;
```



React

10.2 Express Router

EXPRESS Js

node > air-bnb > routes > `js` user.js > ...

```
1 // External Modules
2 const express = require('express');
3 const userRouter = express.Router();
4
5 userRouter.use('/', (req, res, next) => {
6   console.log("Second Middleware", req.url, req.method);
7   res.send(`
8     <h2>Welcome to AirBnb</h2>
9     <a href="/add-home">Register Your Home on AirBnb</a>
10    `);
11 });
12
13 module.exports = userRouter;
```

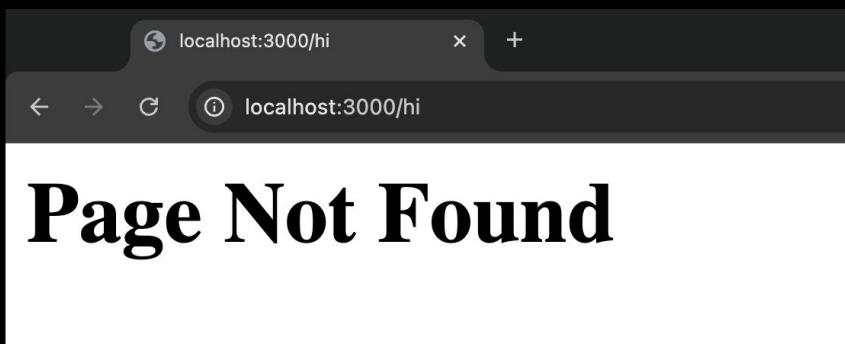


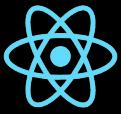
React

10.3 Adding 404

EXPRESS Js

```
app.use(express.urlencoded({extended:true}));  
  
app.use(hostRouter);  
app.use(userRouter);  
  
app.use((req, res, next) => {  
  res.status(404).send('<h1>Page Not Found</h1>');  
});  
  
const PORT = 3000;  
app.listen(PORT, () => {  
  console.log(`Server running at http://localhost:\${PORT}/`);  
});
```





React

10.4 Common Paths

EXPRESS Js

```
// GET path
router.get('/host/add-home', (req, res, next) => {
  res.send(`<h1>Register Your Home on AirBnB</h1>
<form action="/add-home" method="POST">
  <input type="text" name="houseName"
    placeholder="Enter your House Name"><br>
  <br><input type="submit" value="Submit">
</form>
`);
});
```

```
// POST path
router.post('/host/add-home', (req, res, next) => {
  console.log("Post Request details", req.url, req.
  method, req.body);
  res.redirect('/');
});
```

or

```
const app = express();

app.use(express.urlencoded({extended:true}));

app.use("/host", hostRouter);
app.use("/user", userRouter);

app.use((req, res, next) => {
  res.status(404).send('<h1>Page Not Found</h1>');
});
```



React

10.5 Adding HTML Files

EXPRESS Js

air-bnb > views > add-home.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
  initial-scale=1.0" />
  <title>airbnb</title>
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="/">Go to Home</a></li>
        <li><a href="/host/add-home">Add Home</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <h1>Register Your Home on AirBnB</h1>
    <form action="/host/add-home" method="POST">
      <input
        type="text"
        name="houseName"
        placeholder="Enter your House Name"
      /><br />
      <br /><input type="submit" value="Submit" />
    </form>
  </main>
</body>
</html>
```

air-bnb > views > user.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
  initial-scale=1.0" />
  <title>airbnb</title>
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="/">Go to Home</a></li>
        <li><a href="/host/add-home">Add Home</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <h2>Welcome to AirBnb</h2>
  </main>
</body>
</html>
```

air-bnb

> node_modules

> routes

host.js

user.js

> views

404.html

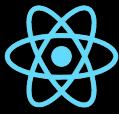
add-home.html

user.html

app.js

package-lock.json

package.json



React

10.6 Serving HTML Files

EXPRESS Js

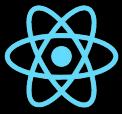
air-bnb > routes > `host.js` > ...

```
// Core Modules
const path = require('path');
```

```
// External Modules
const express = require('express');
```

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

```
// GET path
router.get('/add-home', (req, res, next) => {
  res.sendFile(path.join(__dirname, '../', 'views', 'add-home.html'));
});
```



React

airbnb > util > `path.js` > ...

```
// Core Modules
const path = require('path');
```

```
module.exports = path.dirname(require.main.filename);
```

```
// Local Modules
```

```
const rootDir = require('../util/path');
```

```
userRouter.get('/', (req, res, next) => {
  console.log("Second Middleware", req.url, req.method);
  res.sendFile(path.join(rootDir, 'views', 'user.html'));
});
```

EXPRESS Js

A file tree visualization for the 'airbnb' project. The root folder contains 'node_modules', 'routes', 'util', and 'views'. 'util' contains 'host.js' and 'user.js'. 'views' contains 'app.js'. Additionally, there are 'package-lock.json' and 'package.json' files at the root level.

```
airbnb
├── node_modules
└── routes
    ├── host.js
    └── user.js
└── util
    ├── host.js
    └── user.js
└── views
    ├── app.js
    └── package-lock.json
    └── package.json
```



React

Practise Set

Reuse the app from the last assignment

1. Parse the body of the contact-us request and log it to console.
2. Move the code to separate local modules and use the Express router to import and use them in app.js
3. Move all the html code to html files and serve them using the file helper.
4. Also add a 404 page for this app.

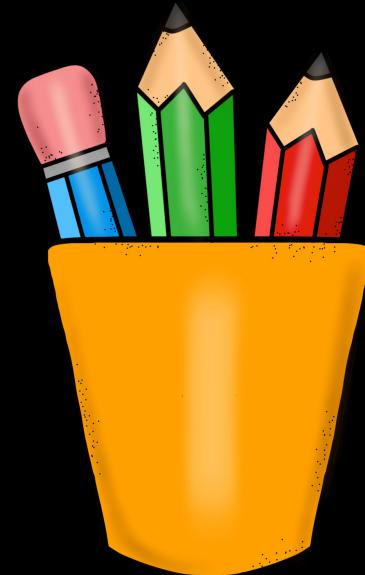


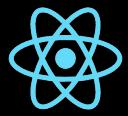


React

Revision

1. Parsing Requests
2. Express Router
3. Adding 404
4. Common Paths
5. Adding HTML Files
6. Serving HTML Files
7. Using File Helper



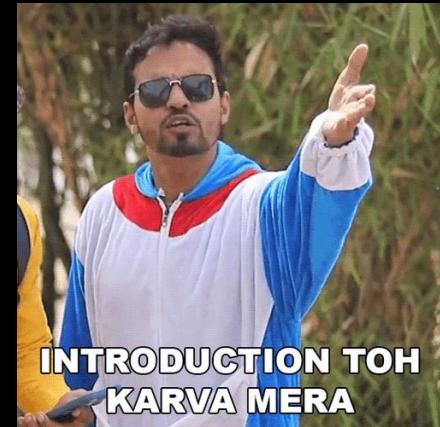


React

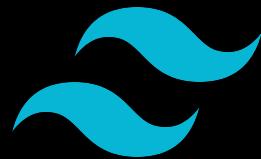


11. Styling using tailwindcss

1. Serving Static Files
2. Introduction to Tailwind CSS
3. Utility Classes
4. Installing Extension
5. Including Tailwind CSS
6. Installing Tailwind CSS
7. Using Tailwind CSS
8. Building Tailwind Automatically



11.1 Serving Static Files



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>airbnb</title>
    <link rel="stylesheet" href="airbnb.css">
  </head>
  <body>
    <header>
      <nav>
        <ul>
          <li><a href="/">Go to Home</a></li>
          <li><a href="/host/add-home">Add Home</a></li>
        </ul>
      </nav>
    </header>
    <main>
      <h2>Welcome to AirBnb</h2>
    </main>
  </body>
</html>
```

```
body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f5f5f5;
}

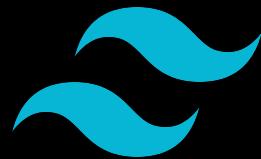
header {
  background-color: #ff5a5f;
  padding: 1rem 2rem;
  box-shadow: 0px 2px 8px rgba(0, 0, 0, 0.1);
}

nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

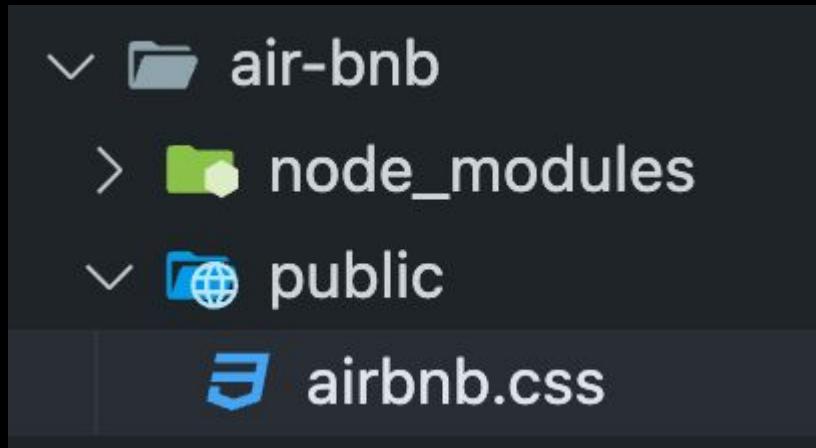
nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
}

nav li {
  margin-left: 20px;
}
```

11.1 Serving Static Files



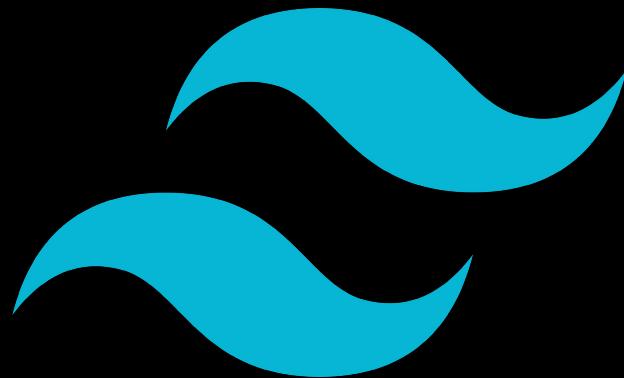
```
nav a {  
    text-decoration: none;  
    color: black;  
    font-weight: bold;  
    padding: 0.5rem 1rem;  
    border-radius: 4px;  
    transition: background-color 0.3s ease;  
}  
  
nav a:hover {  
    background-color: #e54e52;  
}  
  
main {  
    text-align: center;  
    padding: 5rem 2rem;  
    background-color: white;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    margin: 2rem auto;  
    max-width: 600px;  
    border-radius: 8px;  
}  
  
h2 {  
    color: black;  
    font-size: 2rem;  
}
```



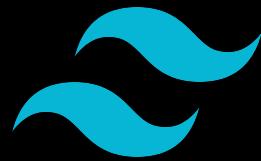
```
// Granting access to public folder  
app.use(express.static(path.join(rootDir, "public")));
```

11.2 Introduction to Tailwind CSS

1. **Responsive:** Mobile-first design for all device sizes.
2. **Utility-First:** Provides low-level utility classes for building custom designs.
3. **Highly Customizable:** Easily extendable through a config file.
4. **Responsive Design:** Built-in responsive utilities (e.g., sm:, md:).
5. **No Predefined Components:** Focuses on building custom components.
6. **Purge CSS:** Removes unused styles in production for smaller files.
7. **Fast Development:** Style elements directly in markup for speed.



11.3 Utility Classes



```
1 /* Colors */
2 .text-primary { color: #007bff; }
3 .bg-primary { background-color: #007bff; }
4
5 /* Sizing */
6 .w-full { width: 100%; }
7 .h-full { height: 100%; }
8
9 /* Typography */
10 .text-center { text-align: center; }
11 .font-bold { font-weight: bold; }
12
13 /* Spacing */
14 .m-1 { margin: 0.25rem; }
15 .m-2 { margin: 0.5rem; }
16 .p-1 { padding: 0.25rem; }
17 .p-2 { padding: 0.5rem; }
```

```
19 /* Layout */
20 .d-flex { display: flex; }
21 .flex-col { flex-direction: column; }
22 .items-center { align-items: center; }
23 .justify-center { justify-content: center; }
24
25 /* Misc */
26 .rounded { border-radius: 0.25rem; }
27 .hidden { display: none; }
```

11.4 Installing Extension



 tailwindcss

INTELLISENSE

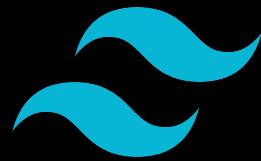
FOR



Visual Studio Code

```
class="bg-t| md:flex md:items-center md:  
v class="fl □ bg-teal-50  
<h2 class="t □ bg-teal-100  
    Back End D □ bg-teal-200  
</h2> □ bg-teal-300      background  
</div> □ bg-teal-400  
div class="mt □ bg-teal-500  
<span class= □ bg-teal-600  
    <button ty □ bg-teal-700  
        Edit □ bg-teal-800  
    </button> □ bg-teal-900  
</span> □ bg-transparent  
<span class= □ bg-top  
    <button type="button" class="in □ bg-line-t  
        Publish □  
    </button>  
</span>
```

11.4 Installing Extension



Tailwind CSS IntelliSense v0.12.10

Tailwind Labs

tailwindcss.com

7,381,739

★★★★★ (99)

Intelligent Tailwind CSS tooling for VS Code

Install



Auto Update



DETAILS

FEATURES

CHANGELOG

A screenshot of the Visual Studio Code interface. A code editor window shows a snippet of Tailwind CSS code with syntax highlighting and hover documentation. A callout box highlights the 'bg-teal' color class, showing a list of color variants: bg-transparent, bg-teal-50, bg-teal-100, bg-teal-200, bg-teal-300, bg-teal-400, bg-teal-500, bg-teal-600, bg-teal-700, bg-teal-800, and bg-teal-900. The background of the code editor has a subtle grid pattern. At the bottom, there are logos for Tailwind CSS and Visual Studio Code.

Categories

Linters

Resources

Marketplace

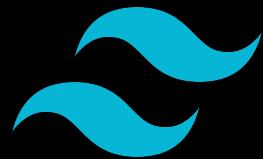
Issues

Repository

License

Tailwind Labs

11.5 Including Tailwind CSS



Installation

Get started with Tailwind CSS

Tailwind CSS works by scanning all of your HTML files, JavaScript components, and any other templates for class names, generating the corresponding styles and then writing them to a static CSS file.

It's fast, flexible, and reliable — with zero-runtime.

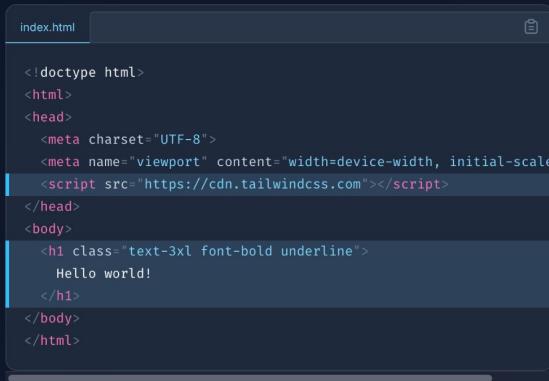
Installation

[Tailwind CLI](#) [Using PostCSS](#) [Framework Guides](#) [Play CDN](#)

Use the Play CDN to try Tailwind right in the browser without any build step. The Play CDN is designed for development purposes only, and is not the best choice for production.

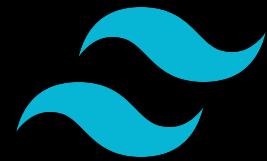
1 Add the Play CDN script to your HTML

Add the Play CDN script tag to the `<head>` of your HTML file, and start using Tailwind's utility classes to style your content.



```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
  <h1 class="text-3xl font-bold underline">
    Hello world!
  </h1>
</body>
</html>
```

11.6 Installing Tailwind CSS



1. Install:

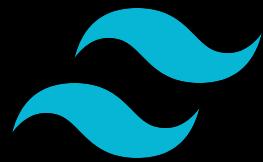
```
npm init -y
```

```
npm install -D tailwindcss postcss autoprefixer
```

2. Initialize Tailwind CSS Config

```
npx tailwindcss init
```

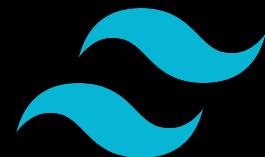
11.6 Installing Tailwind CSS



Configure Tailwind in the Configuration Files (tailwind.config.js)

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  ...
  content: ["./views/*.html"],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

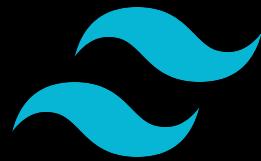
11.6 Installing Tailwind CSS



Add Tailwind Directives to views/input.css

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

11.6 Installing Tailwind CSS



5. Include public/output.css into your index.html

6. Run Command

```
npx tailwindcss -i ./views/input.css -o ./public/output.css --watch
```

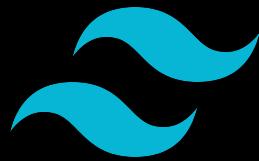
7. Declare Shortcut:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "tailwind": "npx tailwindcss -i ./src/input.css -o ./src/output.css --watch"  
},
```

8. Run Command

```
npm run tailwind
```

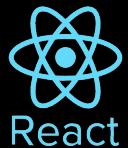
11.7 Using Tailwind CSS



```
<body class="■bg-gray-100 font-sans">
<header class="■bg-red-500 ■text-white p-4 shadow-md">
  <nav class="container mx-auto flex justify-between items-center">
    <ul class="flex space-x-4">
      <li>
        <a href="/" class="■hover:bg-red-400 py-2 px-4 rounded transition duration-300">Go to Home</a>
      </li>
      <li>
        <a href="/host/add-home" class="■hover:bg-red-400 py-2 px-4 rounded transition duration-300">Add Home</a>
      </li>
    </ul>
  </nav>
</header>
<main class="container mx-auto ■bg-white shadow-lg rounded-lg p-8 mt-10 max-w-xl">
  <h2 class="text-2xl ■text-gray-800 font-bold text-center mb-6">
    Welcome to Airbnb
  </h2>
</main>
</body>
```

11.8 Building Tailwind Automatically

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "tailwind": "tailwindcss -i ./views/input.css -o ./public/output.css --watch",  
  "start": "nodemon app.js & npm run tailwind"  
},
```



React

Practise Set

Style other page using
Tailwind.

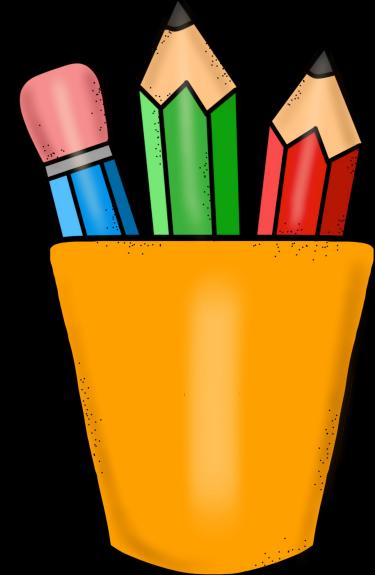


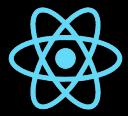


React

Revision

1. Serving Static Files
2. Introduction to Tailwind CSS
3. Utility Classes
4. Installing Extension
5. Including Tailwind CSS
6. Installing Tailwind CSS
7. Using Tailwind CSS





React