

# Dynamic UI ::::::::::;



## 12.1 Need of Dynamic UI



- **Personalized Content:** Tailors responses based on user profiles, preferences, or behaviors to enhance user experience.
- **Dynamic Data Delivery:** Provides real-time information that updates with each request, such as live scores or stock prices.
- **Security and Access Control:** Delivers different content based on user authentication and authorization levels.
- **Localization and Internationalization:** Adjusts responses to accommodate different languages, cultures, or regional settings.
- **API Versatility:** Supports multiple client types (web, mobile, IoT) by providing appropriate data formats and structures.



## 12.2 Sharing using Global Variables



```
const homes = [];  
  
// POST path  
router.post('/add-home', (req, res, next) => {  
  homes.push({houseName: req.body.houseName});  
  res.redirect('/');  
});  
  
exports.hostRouter = router;  
exports.homes = homes;  
  
const {homes} = require('../routes/host');  
  
userRouter.get('/', (req, res, next) => {  
  console.log('homes', homes);  
  res.sendFile(path.join(rootDir, 'views', 'user.html'));  
});
```

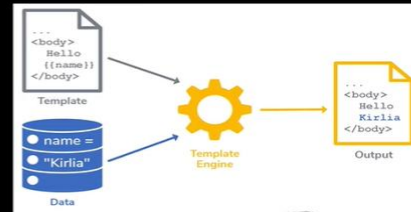
Variables are shared across users



## 12.3 What is EJS



- **HTML with JS:** EJS lets you embed JavaScript code within HTML.
- **Simple Syntax:** Uses `<% %>` for control flow and `<%= %>` for output.
- **Easy to Learn:** Familiar to those who know HTML and JavaScript.
- **Template Reuse:** Supports **partials** for reusing code snippets.
- **Flexible Logic:** Allows full JavaScript expressions in templates.



## 12.4 Installing EJS



```
prashantjain@Prashants-Mac-mini air-bnb % npm install --save ejs
added 6 packages, and audited 232 packages in 879ms
```

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

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



## 12.5 Using EJS Templates



```
views
├── 404.html
├── add-home.html
├── input.css
└── user.ejs
```

```
// Local Modules
const {homes} = require('../routes/host');

userRouter.get('/', (req, res, next) => {
  res.render('user', { homes: homes });
});
```

```
<main class="container mx-auto bg-white shadow-lg rounded-lg p-8 mt-10 max-w-xl">
  <h2 class="text-5xl text-gray-800 font-bold text-center mb-6">
    Available Homes
  </h2>
  <ul class="list-disc list-inside space-y-2">
    <% homes.forEach(function(home) { %>
      <li class="text-gray-700 hover:text-red-500 transition-colors duration-100">
        <%= home.houseName %>
      </li>
    <% }); %>
  </ul>
</main>
```



## 12.6 Working with Partials



- views
  - partials
    - head.ejs
    - nav.ejs
    - 404.ejs
    - add-home.ejs
    - input.css
    - user.ejs

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title><%= title %></title>
7     <link rel="stylesheet" href="/output.css" />
8
```



## 12.6 Working with Partials



```
1 <header class="bg-red-500 text-white p-4 shadow-md">
2   <nav class="container mx-auto flex justify-between items-center">
3     <ul class="flex space-x-4">
4       <li>
5         <a
6           href="/"
7           class="hover:bg-red-400 py-2 px-4 rounded transition duration-300"
8         >Go to Home</a>
9       </li>
10      <li>
11        <a
12          href="/host/add-home"
13          class="hover:bg-red-400 py-2 px-4 rounded transition duration-300"
14        >Add Home</a>
15      </li>
16    </ul>
17  </nav>
18 </header>
```

- views
  - partials
    - head.ejs
    - nav.ejs
    - 404.ejs
    - add-home.ejs



## 12.6 Working with Partials



air-bnb > views > 404.ejs > ?

```
<%= include('partials/head') %>
</head>
<body class="bg-gray-100 font-sans">
  <%= include('partials/nav') %>
  <main class="container mx-auto mt-20 text-center">
    <h2 class="text-6xl font-bold text-red-500 mb-4">404</h2>
    <p class="text-2xl text-gray-700 mb-8">Oops! Page Not Found</p>
    <a href="/" class="bg-red-500 text-white py-2 px-6 rounded-lg
      hover:bg-red-600 transition duration-300">Go Back Home</a>
  </main>
</body>
</html>
```

```
app.use((req, res, next) => {
  res.status(404).render('404', {title: 'Page Not Found'})
});
```