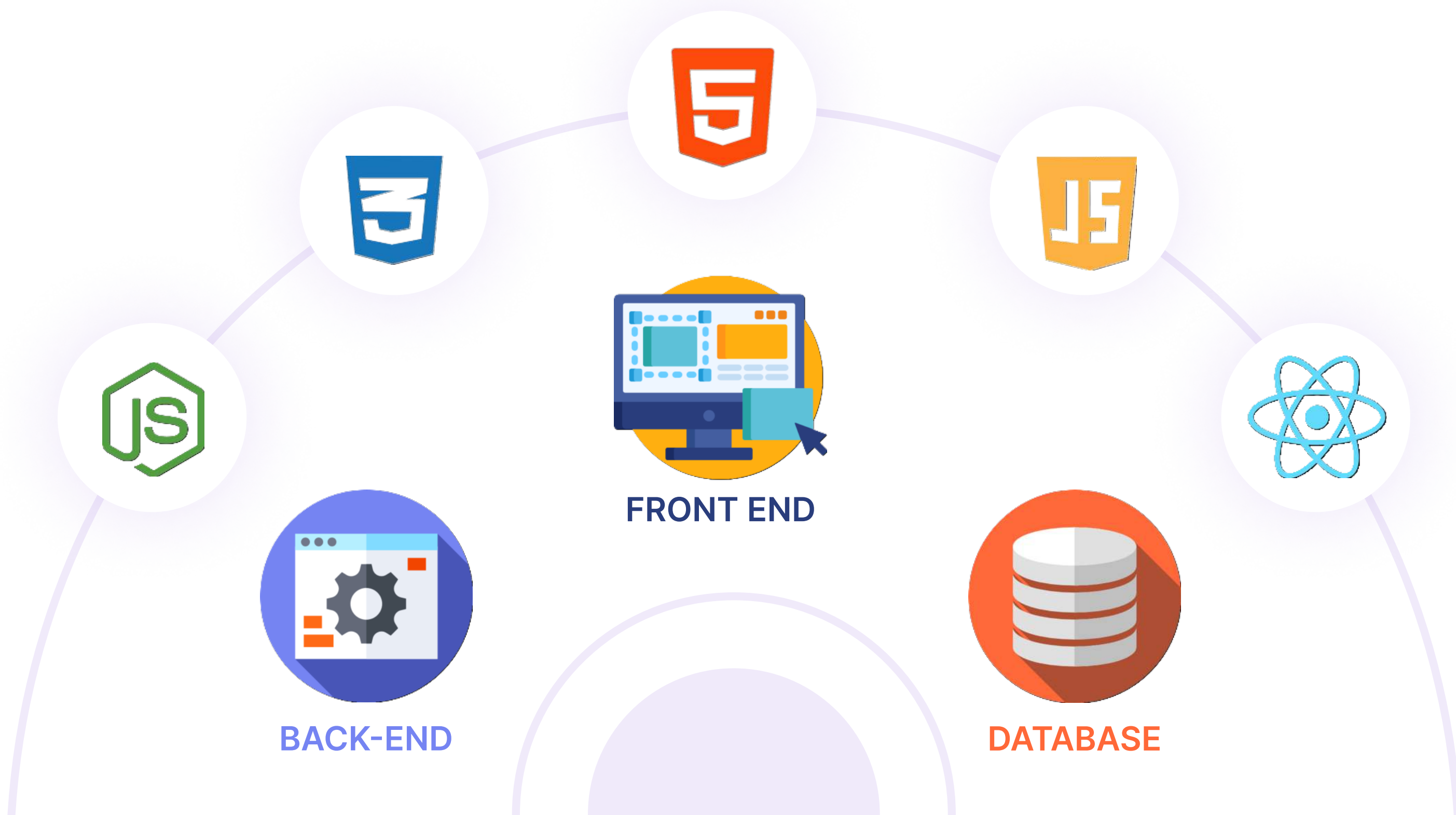


MASTER

**FULL-STACK**

**DEVELOPMENT**

IN 4 WEEKS





## **\*Disclaimer\***

**Master & Revise Full-Stack in 4 weeks.**

But make sure to go on your own pace,  
take breaks, and remember Quality is more  
important than Quantity.

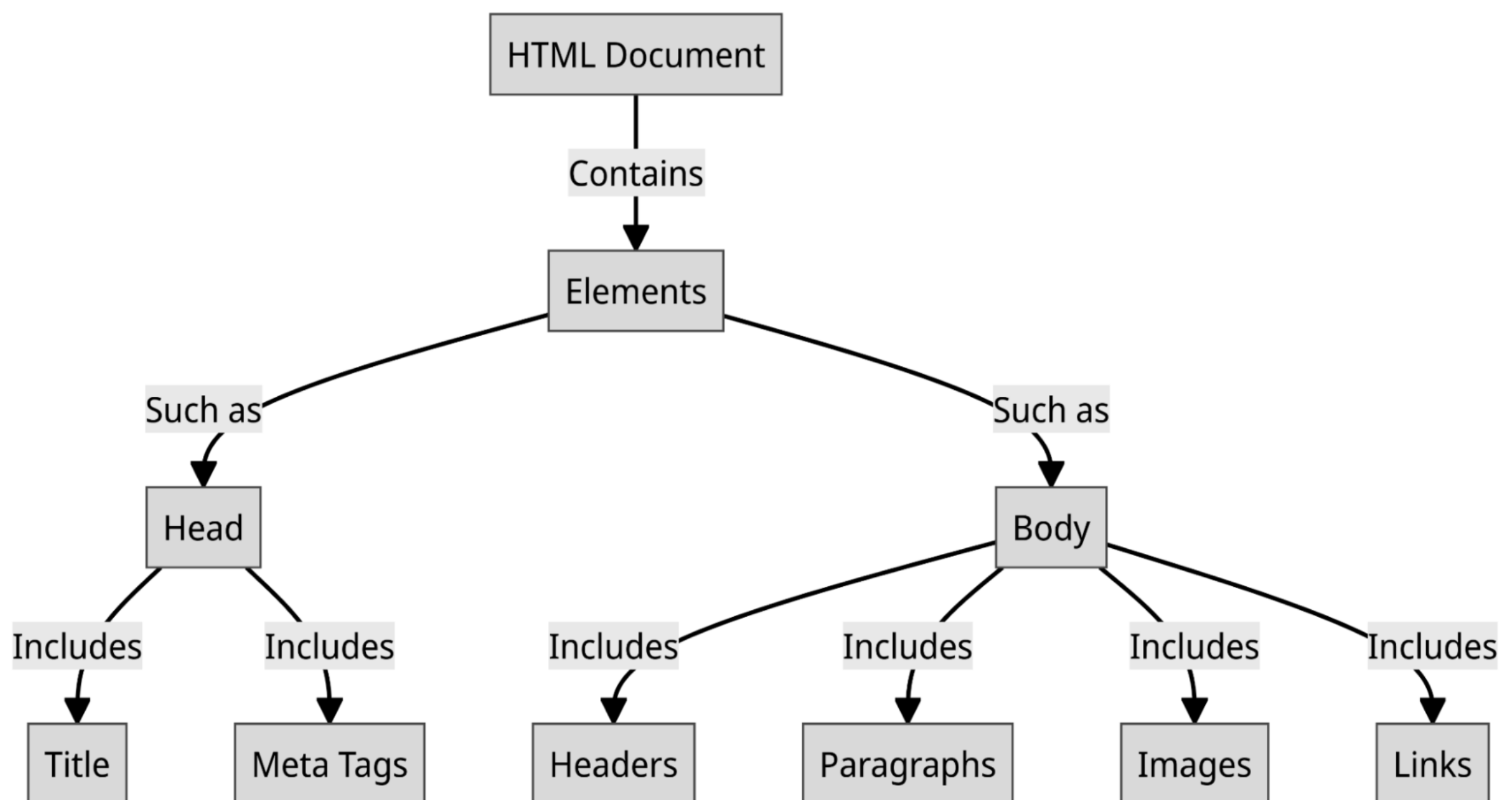
Take the help of this doc and ace your  
tech interview.



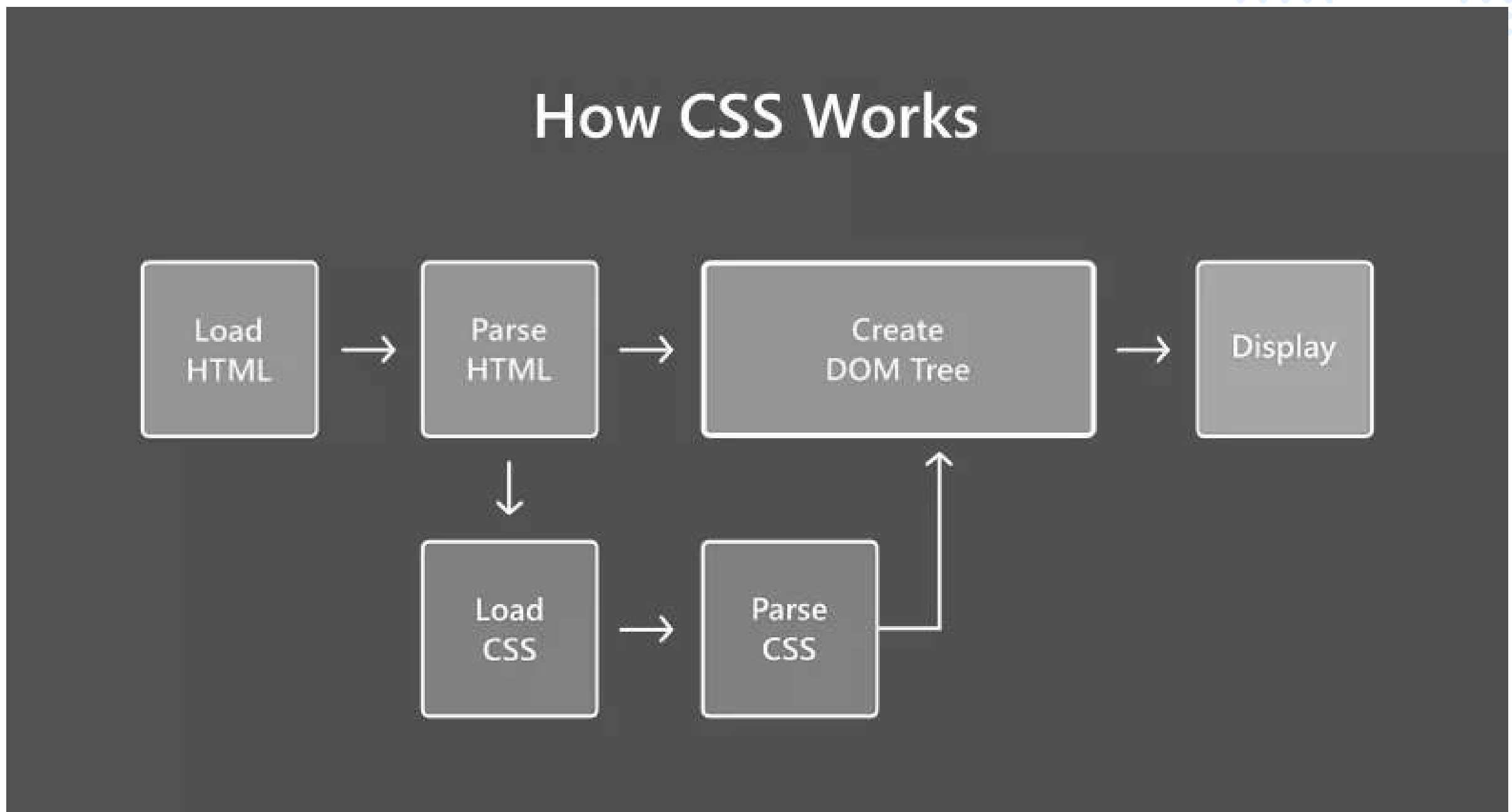


# Introduction to HTML and CSS

- Learn about HTML structure, tags, and elements.



- Understand CSS for styling web pages.



## RESOURCE:

<https://www.youtube.com/watch?v=HcOc7P5BMi4>

[https://www.youtube.com/watch?v=kLO4X\\_3VYdg](https://www.youtube.com/watch?v=kLO4X_3VYdg)



## PRACTICE:

Create a simple webpage.



**Week 1** - Front-End Development

**Day 3-4**

2 hours each



# Intermediate HTML and CSS

- Dive deeper into HTML and CSS techniques.
- Explore responsive design.



## RESOURCE:

<https://www.youtube.com/watch?v=G3e-cpL7ofc>



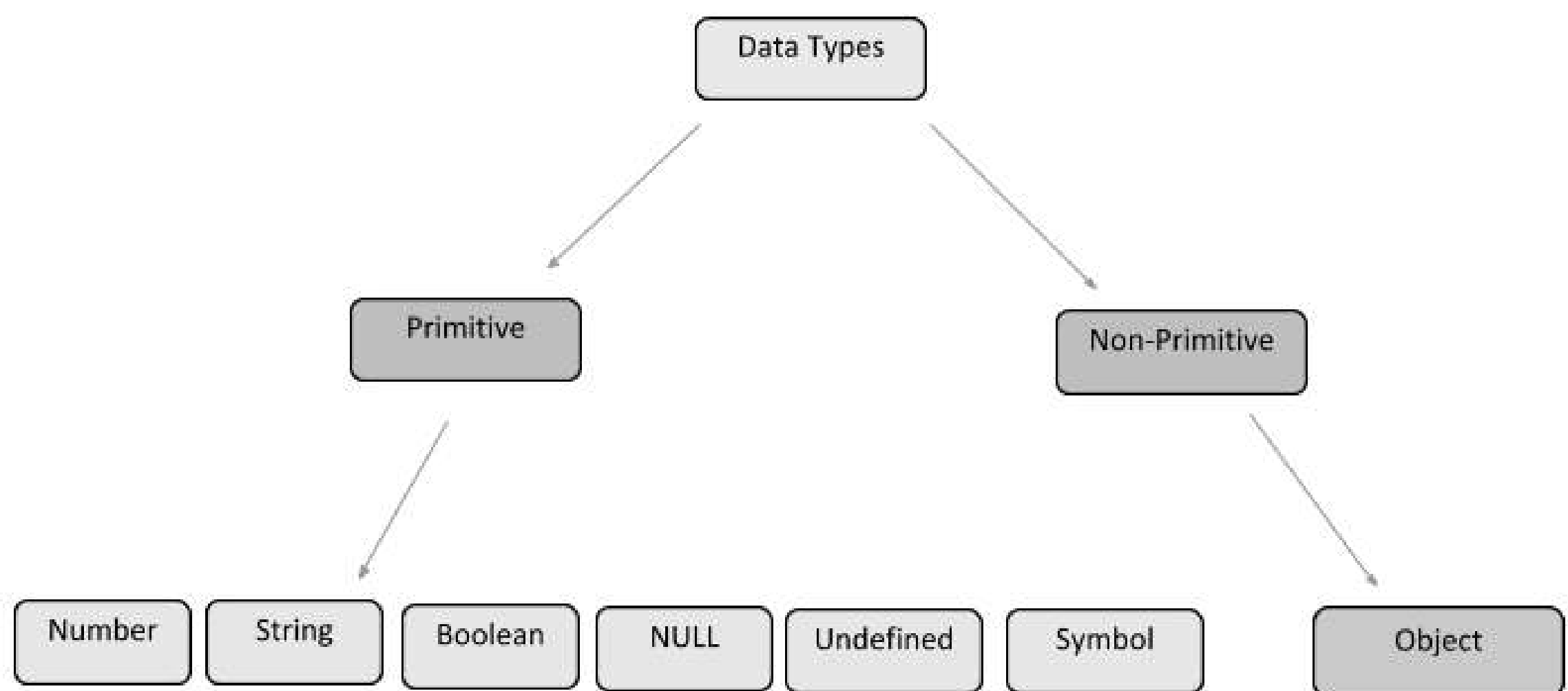
## PRACTICE:

Create a responsive webpage.



# Introduction to JavaScript

Learn the basics of JavaScript: variables, data types, and operators.





## RESOURCE:

<https://www.youtube.com/watch?v=hKB-YGF14SY>



## PRACTICE:

Write simple JavaScript code.





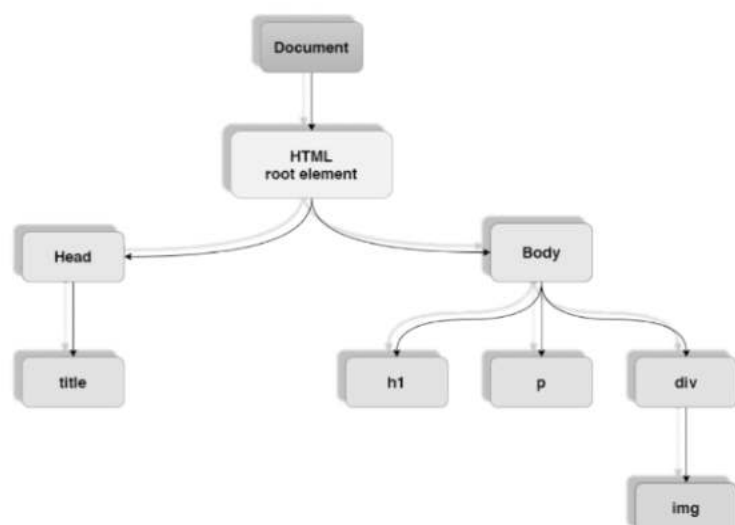
# DOM Manipulation with JavaScript

- Understand Document Object Model (DOM).
- Learn how to interact with HTML using JavaScript.



## Overview

The Document Object Model (DOM) is the structural representation of the HTML elements arranged in hierarchical way to display a webpage and also making it easy to access and manipulate them.



## Accessing Dom Elements

### Get element by id

```
document.getElementById('id');
```

### Get element by class

```
document.getElementsByClassName('className');
```

### Query selector

```
document.querySelector('.classname');
```

### Query selector All

```
document.querySelectorAll('.classname');
```

## Get Children or Parent Node

### Access child node

```
const element = document.getElementById('id');
element.childNodes;
```

### Access parent node

```
const element = document.getElementById('id');
element.parentNode;
```

## Create new elements

### creating an HTML element

```
const ourElement = document.createElement('section');
```

### creating an text node

```
const textNode = document.createTextNode('Hello');
```

## Add & Delete Elements to the DOM

### Access an element first

```
const ourElement = document.getElementById('ourelement');
```

### Using innerHTML

```
ourElement.innerHTML = "Appended from innerHTML";
```

### append as a child for an element

```
ourElement.appendChild(htmlNode);
```

### remove child from an element

```
ourElement.removeChild(htmlNode);
```

### replace child from an element

```
ourElement.removeChild(newElement, oldElement);
```

### Insert before or after an element

```
ourElement.insertAdjacentHTML("beforeend", htmlNode);
```



**beforebegin**  
The HTML would be placed immediately before the element, as a sibling.

**afterbegin**  
The HTML would be placed inside the element, before its first child.

**beforeend**  
The HTML would be placed inside the element, after its last child.

**afterend**  
The HTML would be placed immediately after the element, as a sibling.





## RESOURCE:

<https://www.youtube.com/watch?v=fYR3i0mcE5c>



## PRACTICE:

Create interactive web elements.



# Introduction to React.js

- Learn the fundamentals of React.js.
- Create simple React components.

```
<!DOCTYPE html>
<html>
  <head>
    <script src="build/react.js"></script>
    <script src="build/JSXTransformer.js"></script>
  </head>
  <body>
    <div id="mount-point"></div>
    <script type="text/jsx">
      // React Code Goes Here
    </script>
  </body>
</html>
```



## RESOURCE:

<https://www.youtube.com/watch?v=-mJFZp84TIY>

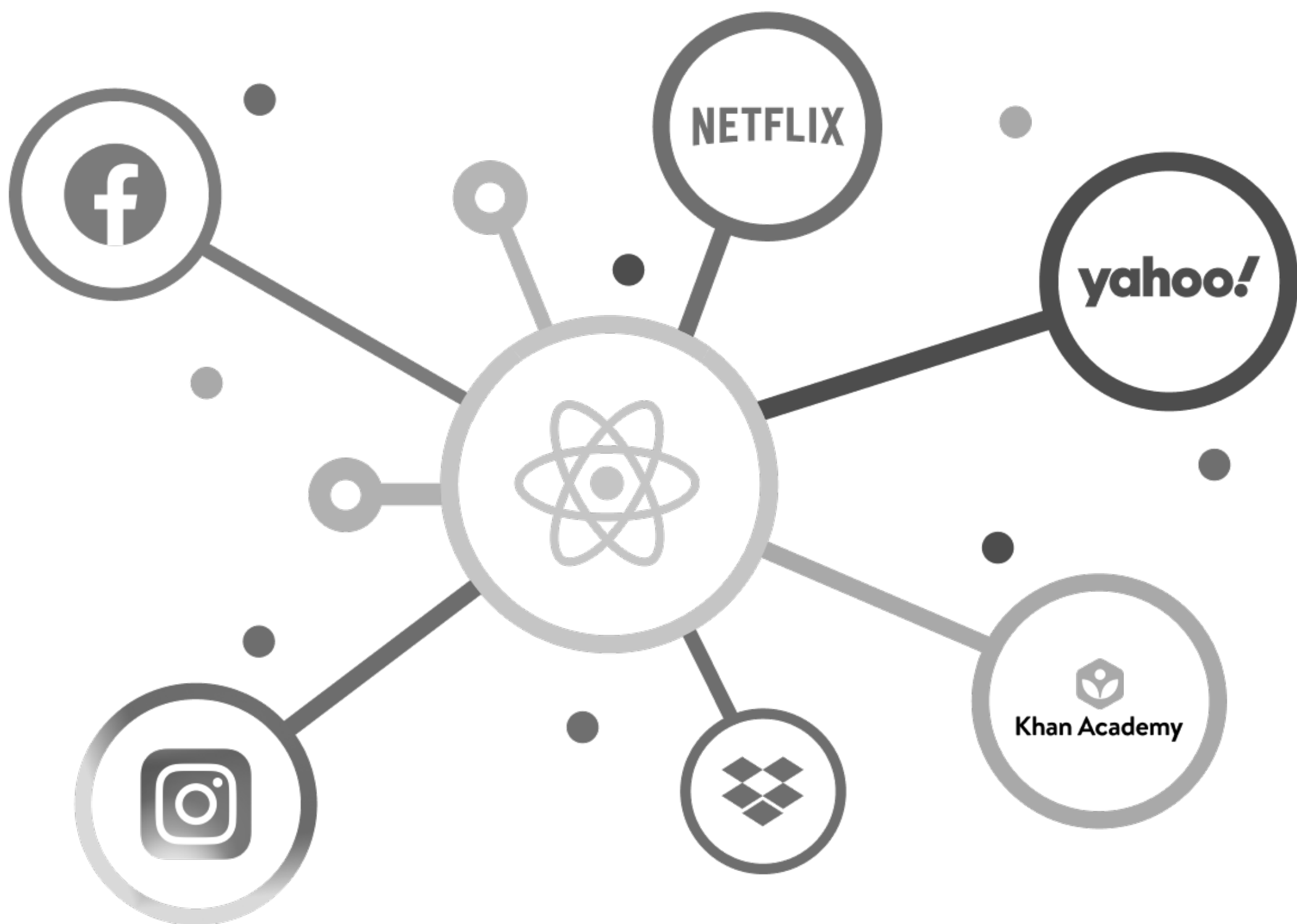


## PRACTICE:

Build a basic React app.

# React.js - Advanced Topics

- Dive deeper into React.js concepts like state and props.
- Learn about React Router for navigation.





## RESOURCE:

<https://www.youtube.com/watch?v=-mJFZp84TIY>

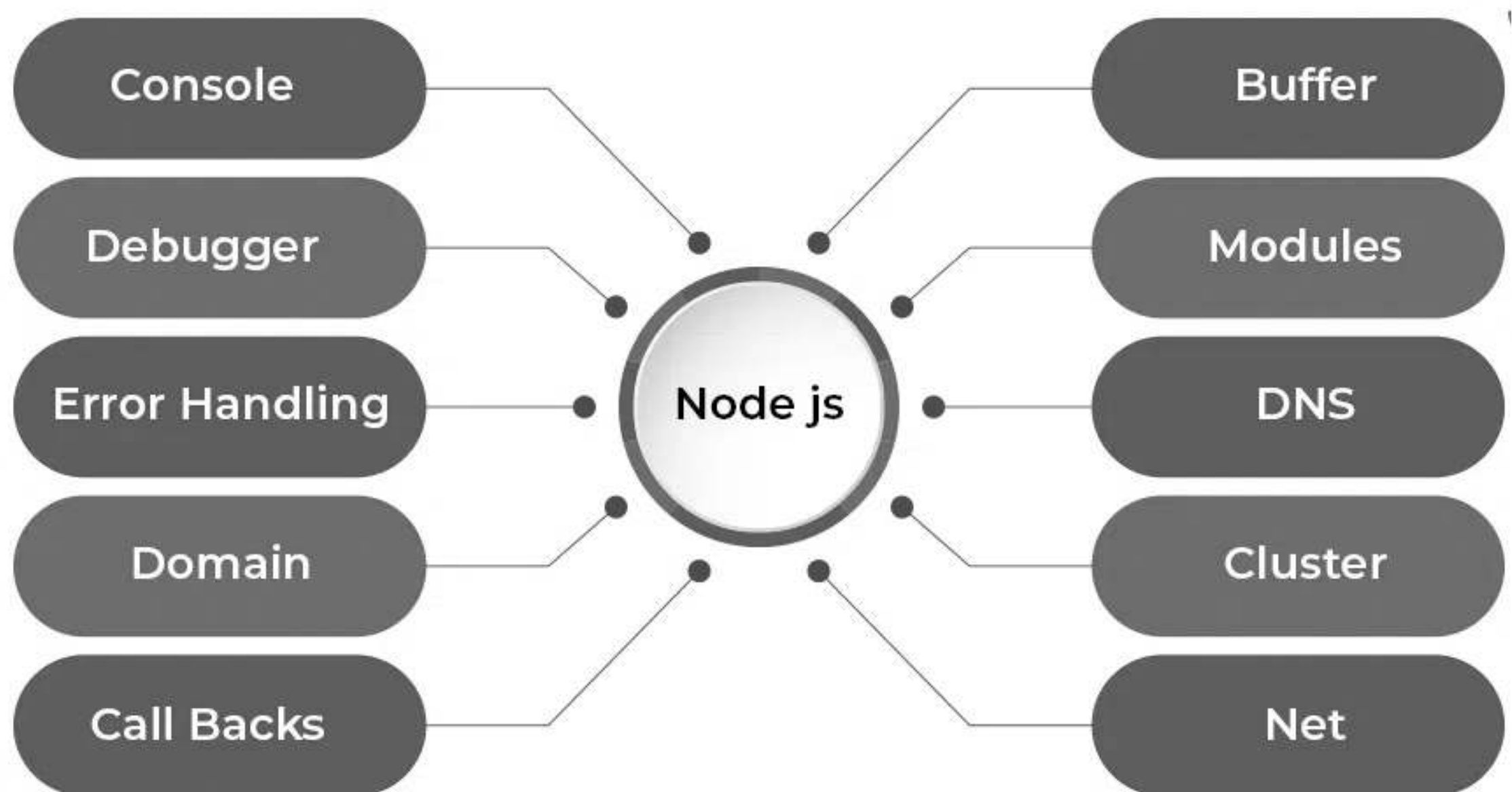


## PRACTICE:

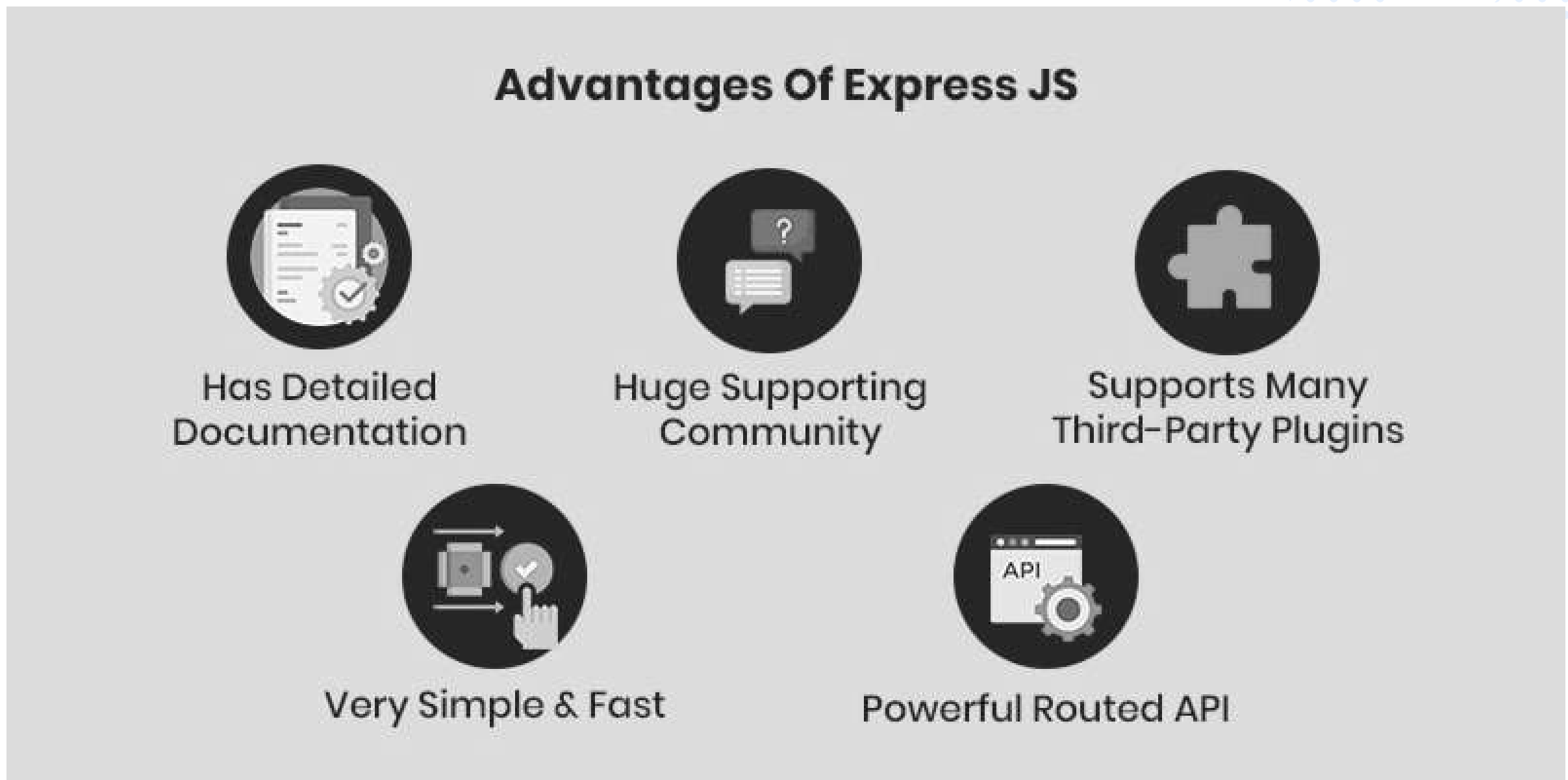
Create a multi-page React app.

# Introduction to Node.js and Express.js

- Understand server-side JavaScript with Node.js.



- Create a basic server using Express.js.



## RESOURCE:

<https://www.youtube.com/watch?v=Oe421EPjeBE>



## PRACTICE:

Build a RESTful API.







# Working with Databases

- Learn about NoSQL databases like MongoDB.
- Connect your Express.js app to MongoDB.



## **PRACTICE:**

Create, Read, Update, and Delete (CRUD) operations.



# Integrating Front-End with Back-End

- Combine your React.js front-end with Express.js back-end.
- Implement API requests and responses.



## RESOURCE:

<https://www.youtube.com/watch?v=fWYZjNGIXhc>

[https://www.youtube.com/watch?v=hYYd\\_3Tz6Zo](https://www.youtube.com/watch?v=hYYd_3Tz6Zo)



## PRACTICE:

Develop a full-stack application.

**Week 4** - Full-Stack Development

**Day 18-19**

2 hours each



# Authentication and Authorization

- Implement user authentication using Passport.js.
- Add authorization to protect routes.



## RESOURCE:

<https://www.youtube.com/watch?v=zfTR2tfu8l0>



## PRACTICE:

Secure your full-stack app.





# Deployment to Heroku

- Learn how to deploy your full-stack app to Heroku.
- Set up environment variables.



## RESOURCE:

<https://www.youtube.com/watch?v=DQk3zJIY-eE>



## PRACTICE:

Deploy your app online.



# Testing and Debugging

- Explore testing frameworks like Jest and Supertest.
- Write unit and integration tests.



## RESOURCE:

[https://www.youtube.com/watch?v=FKnzS\\_icp20](https://www.youtube.com/watch?v=FKnzS_icp20)



## PRACTICE:

Test your full-stack app.

## Week 6 - Finalizing Skills

Day 24-28

2 hours each



# Project Building

- Choose a personal project to work on.
- Apply all your skills learned in previous weeks.



### RESOURCE:

<https://www.youtube.com/watch?v=LXJOvkVYQqA&list=PLE--uVVDmBfcY-dDyaNPITPRBVbZkxE2K>



### PRACTICE:

Create your own full-stack project.



## Week 7 - Review and Practice

Day 29-35

2 hours each

# Review and Practice

- Go back to any areas you found challenging.
- Enhance your project.



### RESOURCE:

<https://www.youtube.com/watch?v=LXJOvkVYQqA&list=PLE--uVVDmBfcY-dDyaNPITPRBVbZkxE2K>



### PRACTICE:

Solve coding challenges and practice interview questions related to full-stack development.







# CONGRATULATIONS


By following this roadmap, you'll have a strong foundation in full-stack development after 70 hours of dedicated learning.

Good luck on your journey to becoming a full-stack developer!



## WHY BOSSCODER?

 **1000+** Alumni placed at Top Product-based companies.

 More than **136% hike** for every **2 out of 3** working professional.

 Average package of **24LPA.**

The syllabus is most **up-to-date** and the list of problems provided covers all important topics.

Lavanya  
 Meta



Course is very well **structured and streamlined** to crack any MAANG company

Rahul  




[\*\*EXPLORE MORE\*\*](#)