

# Study on Documentation Tool

Project Name: Kontaniben? – an E-commerce site.

Programming Language – JavaScript

For documentation we have explored 2 tools – Docusaurus & JSDoc. We read some materials, watched some YouTube tutorials and used them to decide which tool will be used for our project. For taking proper decision we compared this two tools to inform other team members for making decision. This document give a brief description of these two tools and compare them to decide which one will be appropriate for the project.

## Docusaurus:

Docusaurus is an open source, freely available static site generator. This is developed by the team behind Meta. This is built on React. It is used to create web based application. This tool is integrated with GitHub so it make sure that the project's documentation is synced with the software release. To generate a new Docusaurus site using the classic template it will be added to the project after run the command on Command Prompt, Powershell, Terminal:

```
npm init docusaurus@latest my-website classic
```

this command installs all necessary dependencies that we need to run Docusaurus. To start the site need to run the following commands.

```
cd my-website
```

```
npm run start
```

The cd command changes the directory we are working with. In order to work with our newly created Docusaurus site, we'll need to navigate the terminal there.

The npm run start command builds our website locally and serves it through a development server, ready for us to view at <http://localhost:3000/>.

Open docs/intro.md (this page) and edit some lines: the site **reloads automatically** and displays our changes.

This is the 5 min tutorial for learning Docusaurus provided by their official website.

For learning it properly the following links will be helpful.

1. <https://tutorial.docusaurus.io/>

2. <https://www.youtube.com/watch?v=IhYKNgaMmE&list=PLY91jl6VVD7wn8SHdWKRg3AAEKbHQYaNL>
3. <https://www.youtube.com/watch?v=2R53Y7eP45k>

### **Advantage:**

1. Docusaurus is a free-to-use software built by engineers at Meta.
2. Docusaurus is an open-source tool built on ReactJS that is customizable.
3. It can be self-hosted and self-managed.
4. It supports MDX – Markdown language standards and JSX – Javascript XML to write HTML in ReactJS.
5. It is user friendly and offers built in version support that allow to maintain and switch between different versions of documentation easily.

### **Disadvantage:**

1. This has heavy dependencies that means the generated documentation may include large number of dependencies which results to larger build size.
2. Sometimes more advanced customization maybe require deeper understanding of the Docusaurus configuration and theming system.
3. It has limited usecases so if the system needs more complex integration with specific tools might need additional solutions.
4. For small projects or simple documentation needs, its feature might be more than necessary, potentially leading to unnecessary complexity.

## **JSDoc:**

JSDoc is a markup language and tool used to document JavaScript code. It allows developers to embed comments in their code using a specific syntax, which can then be processed by the JSDoc tool to generate comprehensive documentation. This documentation is typically in HTML format and includes details about the code structure, functions, parameters, return types, and more.

## **How to Run:**

To use JSDoc, you need to follow these general steps:

### **1. Install JSDoc:**

You can install JSDoc globally using npm:

```
npm install -g jsdoc
```

### **2. Add Comments:**

Add JSDoc comments to your JavaScript code following the specified syntax. For example:

```
/**  
  
 * Function that adds two numbers.  
  
 * @param {number} a - The first number.  
  
 * @param {number} b - The second number.  
  
 * @returns {number} - The sum of the two numbers.  
  
 */  
  
function add(a, b) {  
  
    return a + b;  
  
}
```

### **3. Run JSDoc:**

Run the JSDoc command in the terminal, specifying the input files or directories:

```
jsdoc yourCode.js
```

## **4. View Documentation:**

Open the generated HTML documentation in a web browser.

### **Links to Learn JSDoc:**

[Official JSDoc Documentation](#)

[Introduction to JSDoc on MDN Web Docs](#)

[JSDoc Cheatsheet](#)

### **Advantages of JSDoc:**

1. **Automated Documentation:** JSDoc allows for the generation of documentation automatically, reducing the need for manual documentation efforts.
2. **Readability:** JSDoc comments enhance code readability by providing clear and structured information about functions, parameters, and return types.
3. **IDE Integration:** Many integrated development environments (IDEs) and editors support JSDoc, providing code suggestions and documentation tooltips as you write code.
4. **Consistency:** Using a standardized format for documentation ensures consistency across a codebase, making it easier for developers to understand and maintain the code.

### **Disadvantages of JSDoc:**

1. **Maintenance Overhead:** Adding and updating JSDoc comments can be seen as an additional task, and if not done consistently, it might lead to outdated documentation.
2. **Learning Curve:** New developers may need time to learn the JSDoc syntax and conventions, adding a learning curve to the onboarding process.
3. **Limited Runtime Impact:** While JSDoc helps with static analysis and documentation generation, it doesn't impact the runtime behavior of the code. Incorrect or missing JSDoc comments won't cause runtime errors.
4. **Limited Support for Complex Types:** While JSDoc supports basic types, documenting complex types and intricate patterns may require additional effort and may not be as intuitive.

In summary, JSDoc is a powerful tool for generating documentation from inline comments, offering advantages in terms of automation, readability, and IDE integration. However, it comes with the overhead of maintaining comments and a learning curve for new developers.