



Google Summer of Code



AsyncAPI

“AsyncAPI Website UI Kit Development”

Table of Content

1. Abstract
2. Project Information
3. Timeline
4. My Understanding of AsyncAPI
5. Involvement in AsyncAPI Community
6. About me
7. Contact Information

1. Abstract

This project aims to enhance the user experience and visual consistency of the AsyncAPI website by developing a comprehensive UI Kit. Currently, the website lacks a proper design system resulting in repeated elements that are not visually consistent. This inconsistency creates a fragmented user experience and makes it difficult to maintain the website's overall design. For example, the website uses multiple variations of buttons, headings, and navigation elements. This makes it difficult for both developers and designers to keep up with the updated system design every time. This wastes time and effort in duplicating the classes every time we want to create a new element that uses the same visual style, and it will lead to further inconsistencies in the website's design as the color palette and components advance and increase further.

The solution to this problem is to develop a comprehensive UI Kit that provides a set of reusable and modular design components such as atomic-level components like pagination, backdrop, button group, slider to organism-level components like notification component, API card component etc. The project aims to assemble components in Figma, develop design components in Storybook, integrate Tailwind CSS and create an easy-to-maintain design system.

2. Project Information

2.1 Current problems with the website

- Lack of design pattern consistency, resulting in repetitive elements without a unified visual style. There are a lot of places where components need to be created for even the smallest of elements that repeat on the website which leads to copying and pasting the classes. For eg. here we can see that the current approach to defining colors and shades in the `tailwind.config.js` file lacks encapsulation and modularity. Each color and its shades are defined separately, leading to repetitive and verbose code. This approach makes it challenging to maintain and update the color palette.

Additionally, it does not fully leverage the utility-first approach that Tailwind CSS advocates for.



```
bsite / tailwind.config.js
code Blame 249 lines (241 loc) · 5.59 KB ·
74     return result
75   }
76
77   module.exports = {
78     content: [
79       './pages/**/*.jsx',
80       './components/**/*.jsx',
81       './config/tools.json',
82     ],
83     theme: {
84       fontWeight: {
85         'extra-light': 200,
86         light: 300,
87         regular: 400,
88         medium: 500,
89         semibold: 600,
90         bold: 700,
91         extrabold: 800,
92         black: 900,
93       },
94       backgroundImage: {
```

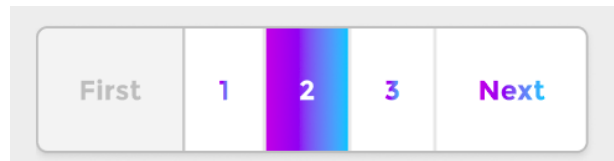
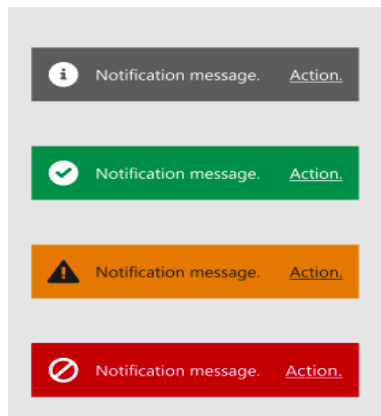
- The absence of an easily manageable and reusable design system, such as a User Interface Kit (UI Kit), has hindered collaboration between designers and developers. React's component-based architecture such as icon components, buttons, modals and Figma's design system features enable the creation of a comprehensive UI Kit that ensures consistency in design elements.
- the AsyncAPI website lacks comprehensive documentation for its design components and UI elements. This absence of detailed documentation makes it challenging for new developers and contributors to understand, utilize, and contribute to the design system effectively

2.2 Project Goals

Based on the instructions provided in this [issue](#) and my own understanding, we have the following goals to achieve:-

- Conduct an audit of all design patterns on the AsyncAPI website to determine the design elements that need to be added and ensure that all website elements adhere to the established brand guidelines, including color schemes, typography, and visual styles.
- Assemble atoms into molecules and then into organisms in Figma, facilitating the creation of modular and consistent design components that align with the project's visual guidelines and design requirements. Here is an example of a complex component - A notification component and an atomic-level component - Pagination.

Note - These are just shown as component samples, changes will be made after discussing with mentors accordingly.



- Utilize Tailwind CSS to create modular, reusable components with encapsulated styles, minimizing class duplication and promoting a more maintainable design system.
- Foster a modular design system that allows for easy maintenance and updates, Here, we will define each component using react for different UI

elements following a component-based architecture. For example buttons, navigation bars, cards etc, use props and state to pass data and manage the behavior of components. In case of a Button component, props can be used to specify the button text, color, size, and click handlers.

- Specific components for AsyncAPI(optional)
 - API documentation component - This component could be used to display API documentation in a user-friendly and interactive way while hovering over technical terms.
 - API explorer - This component could be used to allow users to explore and test APIs. We'll use React to encompass input fields for endpoint selection, request parameters, and authentication details, along with a button to trigger the request.
 - API Subscription Management - This allows users to subscribe to and manage APIs, giving them control over their usage and providing a seamless integration experience.
- Thoroughly test all components, gather feedback, and iterate on design and development aspects as necessary to ensure that the components function as intended and align with the visual and functional goals of the UI Kit.
- Improve developer productivity by providing a set of well-documented components. The UI Kit will be thoroughly documented, making it easy for developers to understand how to use the components and how to integrate them into their own projects.

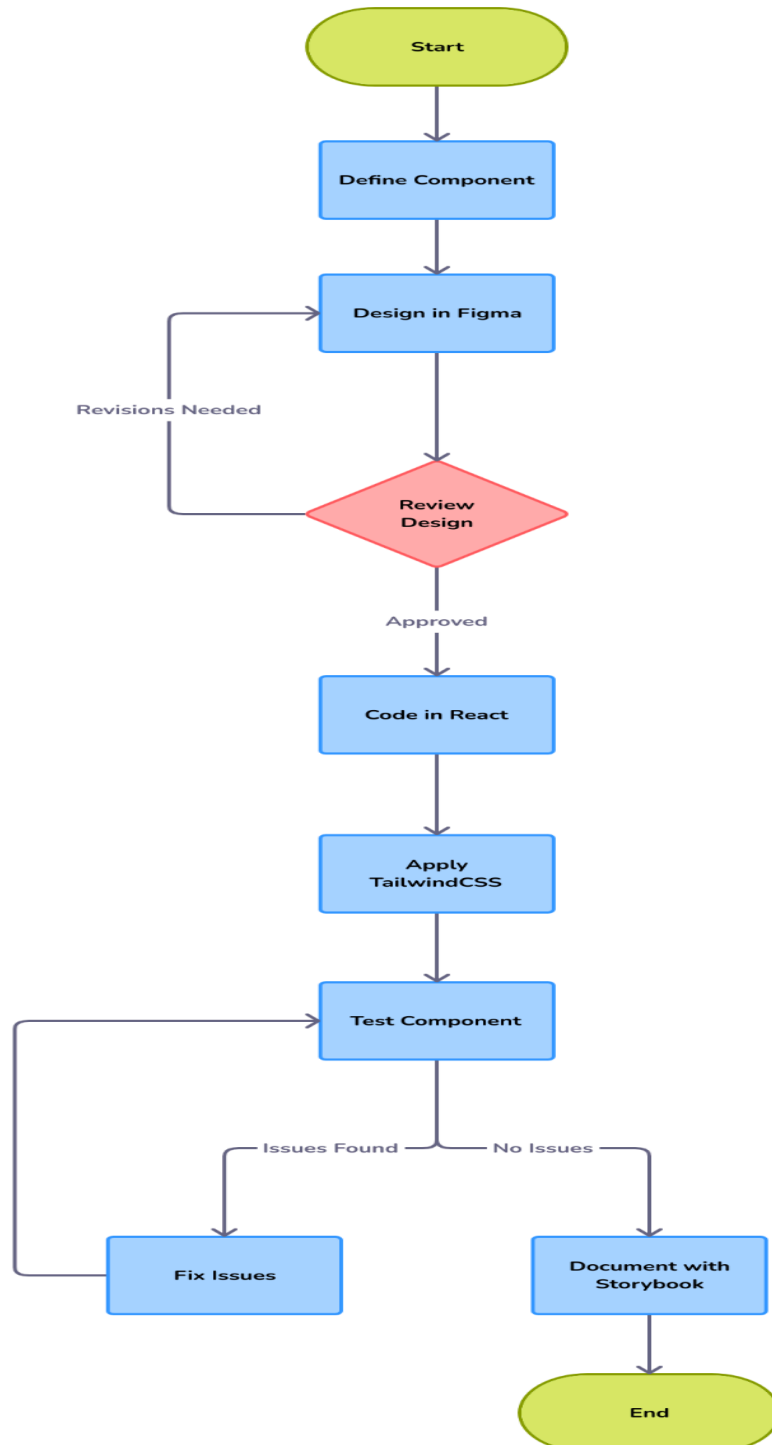
2.3 Project Implementation

1. Project Preparation and Design Patterns Audit

I will begin by conducting an in-depth review of the existing AsyncAPI website design, visual elements, the current usage of Tailwind CSS, and component classes to identify recurring components, visual styles, and areas of inconsistency or redundancy. Document the findings to inform the

development of the UI Kit. Engage with the AsyncAPI team to gather insights on design guidelines, brand principles, and visual identity requirements. This will provide a comprehensive understanding of the current design patterns and coding practices.

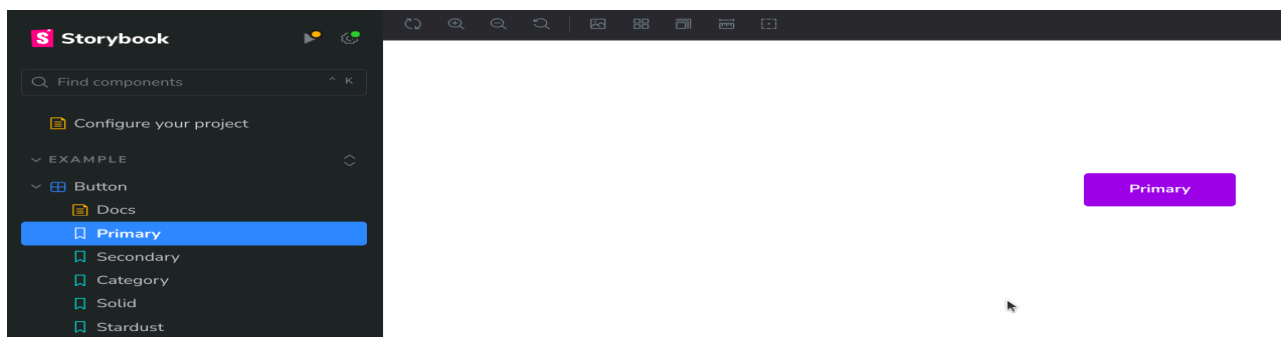
Here is the basic workflow of how the project will move forward:-



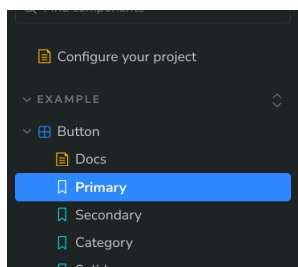
2. UI Kit Design and Development

After this, our objective is to develop a comprehensive UI Kit based on the design patterns audit with a focus on Tailwind CSS and component-based class encapsulation.

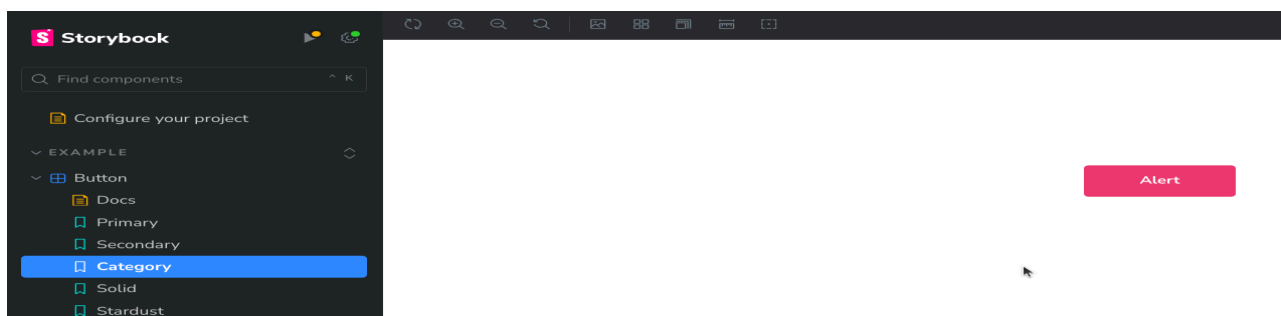
- Figma Mockup Creation - Utilize Figma to create a comprehensive UI Kit that encompasses atomic, molecular, and organism-level components. Work closely with the design team to ensure a consistent design language across all components.
 - Here is the sample button code and how it will be implemented in storybook.
 - I took reference from this [website](#) mentioned in [this issue](#) and these designs are subject to change according to requirements of projects.



Hover state:-



Alert -



Sample code for implementing tailwindcss, storybook, JS/TS, React:-

```
i-library > src > stories > Button.stories.tsx > ...
import React from 'react';
import { Meta, Story } from '@storybook/react';
import { Button, ButtonProps } from '../Button';

export default {
  title: 'Components/Button',
  component: Button,
} as Meta;

const Template: Story<ButtonProps> = (args) => <Button {...args} />;

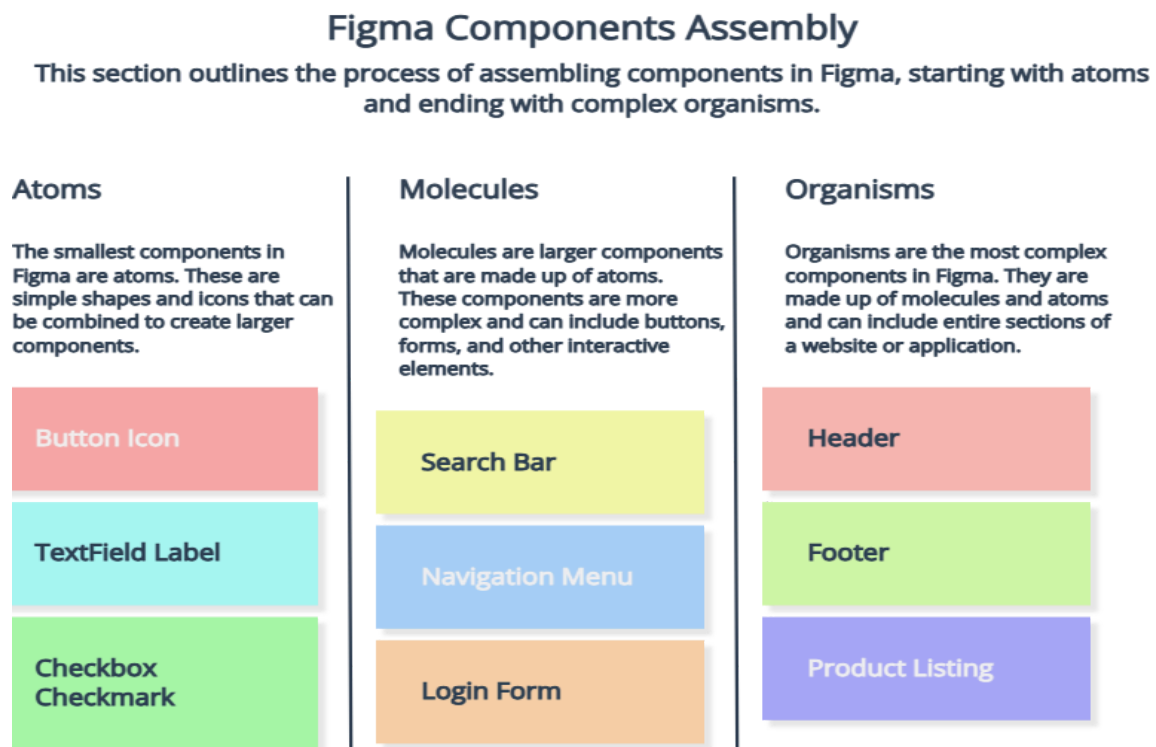
export const Primary = Template.bind({});
Primary.args = {
  text: 'Primary Button',
  type: 'primary',
};

export const Secondary = Template.bind({});
Secondary.args = {
  text: 'Secondary Button',
  type: 'secondary',
};
```

```
my-ui-library > src > stories > Button.tsx > Button > buttonClasses
1  import React from 'react';
2
3  export interface ButtonProps {
4    text: string;
5    type?: 'primary' | 'secondary';
6    onClick?: () => void;
7  }
8
9  export const Button: React.FC<ButtonProps> = ({
10    text,
11    type = 'primary',
12    onClick,
13  }) => {
14    const buttonClasses =
15      type === 'primary'
16        ? 'bg-purple-600 hover:bg-purple-700 text-white'
17        : 'bg-blue-400 hover:bg-blue-500 text-black';
18
19    return (
20      <button className={`px-4 py-2 rounded ${buttonClasses}`}
21        onClick={onClick}>
22        {text}
23      </button>
24    );
25  };
26
27
```

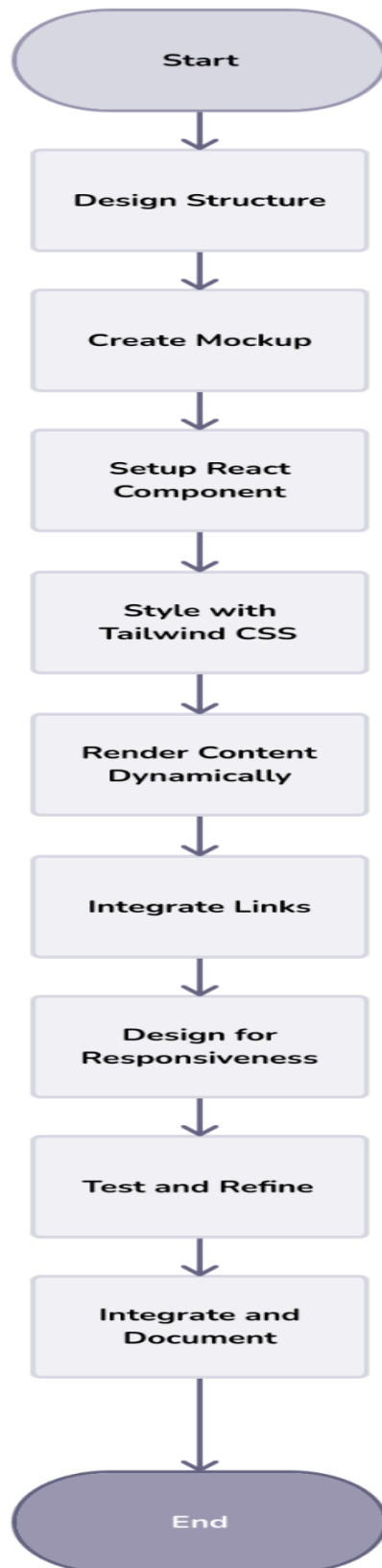

- UI Component Implementation in Storybook - Use Storybook to develop and document each component within the UI Kit. This includes representing various component states and ensuring responsive behavior across different device breakpoints.
- Tailwind CSS Integration - Integrate Tailwind CSS to construct the UI Kit, adhering to a utility-first approach. Implement class encapsulation to minimize redundancy and maintain a modular design system.

Here is the basic understanding of figma components and how they are assembled.



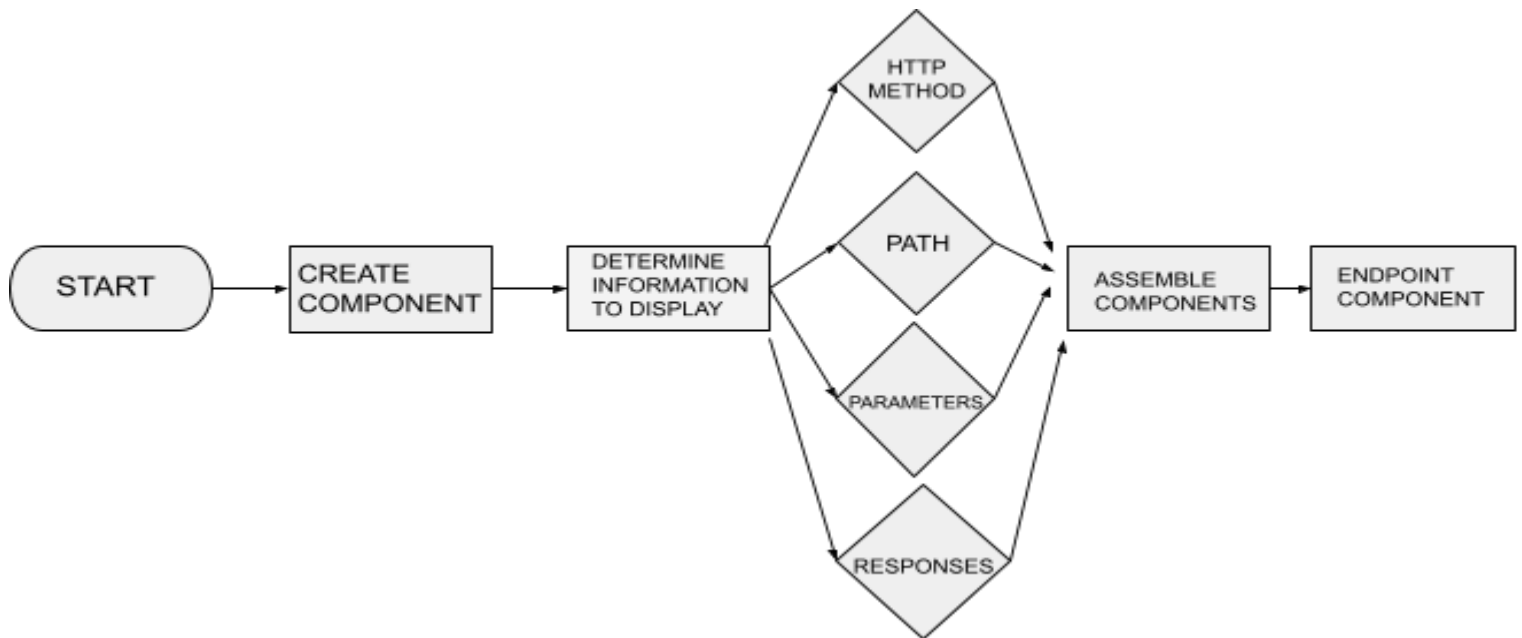
From creating a component to its implementation, we will need a proper roadmap to implement it with proper documentation and development in Storybook. At this stage, with the help of my mentors, I will first confirm the type of components that we need to implement and what expectations they have with the final component for the AsyncAPI UI Kit.

Given below is the workflow of how a component will be made and implemented.

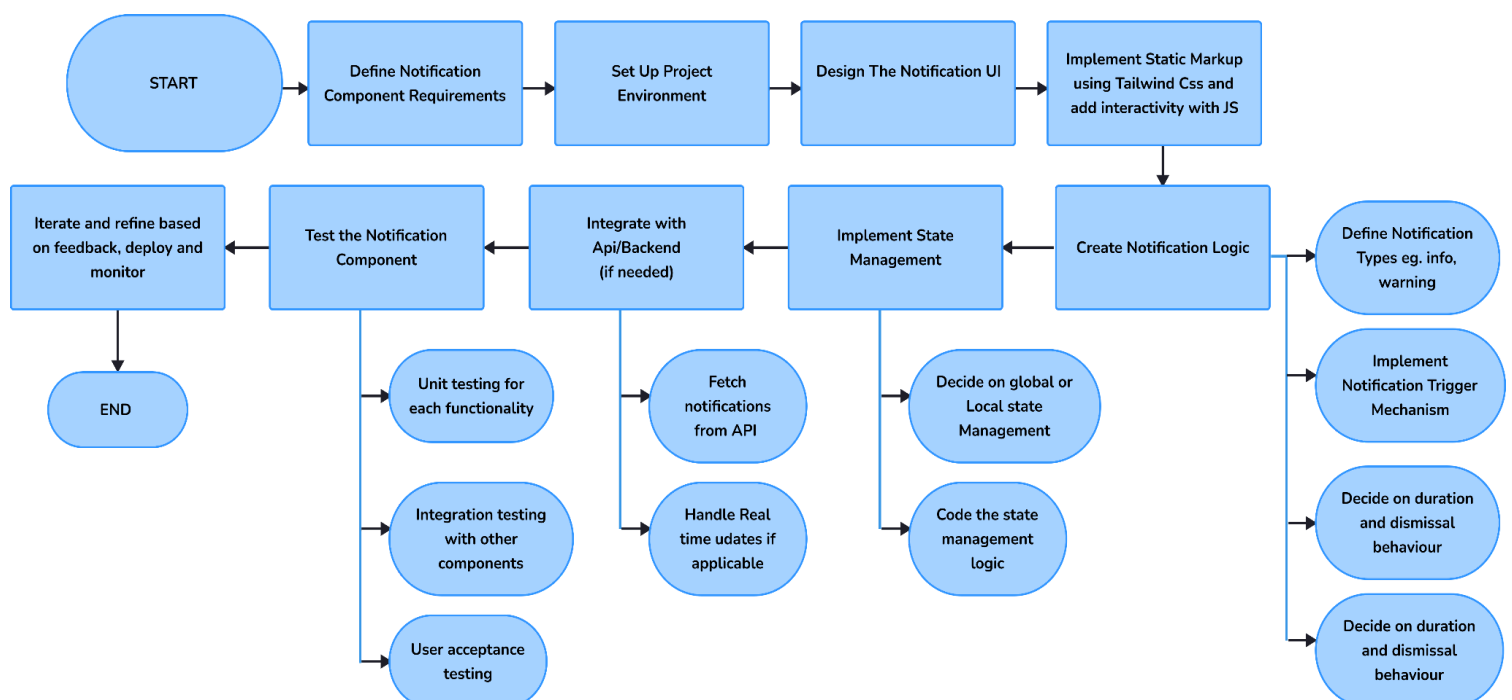


Talking about specific components for asyncAPI such as endpoint component. Let's have a look for its implementation, which comes under the category of organism-level component.

This component displays detailed information about each API, including title, description, version, and a link to the API documentation. Shows detailed information about an API endpoint, such as HTTP method, path, parameters, and responses.



For a comprehensive workflow of how an organism-level component will be made and implemented end to end let's have a look at the Notification component example.



Using AI tools to improve design(optional)

Coming to the current trends and latest technologies, integrating Figma AI into the UI Kit design process can revolutionize the creation of atomic, molecular, and organism-level components.

Figma AI can intelligently assemble molecular components from atomic components, considering their relationships and dependencies. It can generate organism components that are tailored to specific use cases and scenarios. Beyond Figma's native AI capabilities, designers can leverage the power of external AI tools to further enhance the component creation process. I searched and explored some of the AI tools which can be beneficial:-

- Content Generation - I have previously experimented Copy.ai and Jasper can generate unique and compelling text content for UI components, such as button labels, error messages, and placeholder text.
- Image Optimization - Tools like TinyPNG and ImageOptim can automatically compress and optimize images used in components, reducing file size without compromising visual quality, I have also used CanvaAI, Visme to create thumbnails for my youtube channel and designing.
- Accessibility Analysis - Accessibility AI tools like aXe and WAVE can analyze components for accessibility issues, ensuring that they meet WCAG standards and are accessible to users with disabilities.
- Code Generation - Code generators like Anima and Kite can automatically generate code for components based on their design in Figma, accelerating the development process and reducing the risk of errors.

3. Test components, gather feedback, iterate on design.

- When conducting testing for components, especially in the context of UI development, a combination of manual and automated testing approaches can be employed to ensure thorough coverage.

- To ensure that components are not only visually consistent but are also engineered to be dynamic and adaptable to diverse content and use cases. At this stage I'll be continuously discuss with the mentors, get feedback and reiterate as it is very important to make sure that components work properly.

4. Documentation and Knowledge Transfer

- At this stage, I will work on writing comprehensive documentation for designers and developers to effectively use and understand the UI Kit.
- Create detailed documentation for each component in the UI Kit, encompassing best practices for implementation, customization, and usage patterns.
- Prepare a comprehensive developer guide that explains how to effectively utilize the UI Kit and integrate it into new or existing projects. This will especially be helpful for new developers and contributors.
- Establish clear guidelines for community contributions to the UI Kit, to ensure adherence to brand and design standards.

3. Timeline

I'm able to dedicate 40-45 hours per week on GSoC this summer. I don't have any other commitments except learning and implementing the required work as I also have to increase my skills and knowledge especially related to the project while implementing it so I'll give extra time to learn and explore more about the project and techstacks. It will also help me to achieve the goals of the project. I'm ready to put in extra hours if required. I feel a good motivation to contribute to this project as I am really interested in exploring development with design especially in the era of AI where every day there is a new possibility of creativity and enhancement in art. I'll try my best to complete my work by 3 September 2024 keeping in mind that I do not need to backtrack for the already completed tasks.

After every milestone, I'll write a blog about my progress and experience with the project on hashnode. I am more than ready to work well past my committed time if needed.

<p>May 1 - 26</p> <p>(Community Bonding Period)</p> <p>Milestone: Familiarize with the existing AsyncAPI website design, review Tailwind CSS implementation, and discuss the project scope with mentors.</p>	<ul style="list-style-type: none">• Conduct design pattern audit and collaborate with the design team to gather insights on design guidelines and visual identity requirements.• Create a plan for the UI Kit development, including the use of Figma and Storybook for component design and implementation. <p>(At this time, I have my college end-semester exams, so I'll complete tasks accordingly)</p>
<p>May 27 - July 8</p> <p>(Coding officially begins)</p> <p>Milestone: Completion of the UI Kit design in Figma and commencement of Storybook development.</p>	<ul style="list-style-type: none">• Week 1-3 (May 27 - June 16) - Designing UI components in Figma.<ul style="list-style-type: none">○ Conduct a comprehensive exploration of the atomic, molecular, and organism-level components outlined in the project objectives.○ Create high-fidelity visual designs for individual UI components, ensuring consistency with the overall design language and brand identity.○ Establish a structured and organized Figma library that encompasses all designed components, allowing easy access and reuse during the subsequent development phase.• Week 4-5 (June 17 - July 1) - Commencing UI Kit Development in Storybook<ul style="list-style-type: none">○ Transition from the design phase to the development phase by commencing the implementation of UI components in Storybook.○ Translate the designed UI components into code within Storybook, ensuring adherence to the specified design patterns and visual guidelines. Integrate Tailwind CSS to construct the UI Kit, adhering to a utility-first approach.○ Validate the responsiveness and consistency

	of UI components across different device breakpoints within Storybook.
<p>July 12 - August 19 (Midterm Evaluation Deadline)</p> <p>Milestone: Finalization of UI Kit components, documentation, and commencing testing and iteration phase.</p>	<ul style="list-style-type: none"> • Week 5-7 (July 12 - August 2) - <ul style="list-style-type: none"> ◦ Testing of UI Kit components, gathering feedback, and making iterative improvements based on testing results and feedback. Ensure that the UI Kit is well-documented, allowing for easy integration and adoption by future contributors or developers. • Week 8-10 (August 3 - August 19) - <ul style="list-style-type: none"> ◦ Conducting thorough testing to verify the functionality and usability of the UI Kit across various platforms and devices.
<p>August 19 - September 3 (Final Week)</p> <p>Milestone: Official submission of the final work product.</p>	<ul style="list-style-type: none"> • Week 10-11 (August 19 - 26) - <ul style="list-style-type: none"> ◦ Mostly kept free to be used in case of emergency. ◦ The focus will be on refining and, if necessary, refactoring the codebase and updating the documentation for the final submission. ◦ Submission of final work product including fully developed UI Kit, comprehensive documentation, and final mentor evaluation.
<p>Optional Milestone (Extended deadline)</p>	<ul style="list-style-type: none"> • At this time, if time permits I would like to explore new possibilities of designing and creating components specific to AsyncAPI • I would also like to explore what areas could be enhanced and implemented with the help of AI designing tools, how it can benefit and improve productivity on other designers and developers.

4. My Understanding of AsyncAPI

I recently attended Global Open Source Cohort in March 2024 where I interacted with different maintainers of various open-source programs such as Outreachy, Fossasia, MLH, LFX. One selected student in one of these programs told her great experience and engagement with AsyncAPI community. From there I got interested in contributing to AsyncAPI. Also, I recently completed Build@Mercari'24 training program conducted by Mercari, Tokyo, Japan. I was among one of the 6 candidates selected from my country for this program where we learned about backend development, docker, CI/CD, workflow and pipeline, API development, data fetching, data analysis etc. I created different routes, API endpoints to fetch data from the database based on HTTP methods, Implemented error handling, API responses, etc. From there I got interested in API development. From my knowledge and exploration, AsyncAPI involves creating specifications, tools and resources to ease the design, documentation, and implementation of asynchronous communication and handling the real-time events APIs. Also, with asynchronous APIs, the client can send a request to the server and continue its operations without waiting for an immediate response. I have a good amount of experience in API development infrastructure, testing and implementation. I have used Postman for API testing in my projects and have 1-1.5 years of experience with Git/Github.

Here are some of the projects Including my API experience -

- [RC_NITT](#) - Restructured and made the website dynamic using React, javascript for college work.
- [Hacknitte](#) - A website to improve coding culture of college by creating a unified dashboard by integrating leetcode, codechef, codeforces user ids. Used various coding platforms API and successfully implemented for various use cases. Used React, mysql, spline3d, Bootstrap, MantineUI, Tailwind css
- [Youtube Clone](#) - I was experimenting with adding new features in my youtube clone out of curiosity as I am fascinated by how youtube handles data and want to explore the technical side of it. I got to know about various aspects of API implementation by using [Youtube data API](#). Used

state management systems like `useState` and `useEffect` for handling response and api fetching.

- [Student Internship Portal](#) - Made a website to make it easier for first and second year college students to get alumni-based internships. Used Firebase, Google form API and developed different routes connecting to companies website.
- [Transfinitte](#) - Made a Website for college flagship 42 hrs hackathon. MantineUI, MaterialUI, spline3d, figma etc for designing frontend.
- [Guess word game](#) - Created an app where users have to guess words based on hints provided.
- [Hurdle game](#) - Developed a game where the user has to tap on the screen to avoid hurdles.
- [Settle payment app](#) - Made a payment settling app to make group transactions easier.
- [Decentralised crowdfunding app](#) - Made workable backend of crowdfunding app using solidity.

5. Involvement in AsyncAPI Community

I am new to this organization but I am highly motivated to get involved in this organization in future. Whether or not my proposal gets accepted, I will still be equally interested in working on this project because I feel a good motivation to contribute to this project. Also, I am planning to attend the AsyncAPI conference virtually this year and hope to attend it offline after making some major contributions till next year.

I have specific interest in API development as it directly impacts the connectivity and functionality of software systems. From my own experience, API fetching is incredibly beneficial for new developers, as it enables them to access real-time data while building even simple websites or apps. This capability allows developers to focus directly on their primary objectives while letting the API handle other data requirements. This is the main reason why I have even more interest in exploring it.

Initially I was more into development than designing. However after exploring various software like Figma, Blender, Adobe Illustrator etc and especially after seeing so many new possibilities in design with the help of AI, I got very curious to learn skills in this area. Recently I used various AI tools in designing such as

Uizard, RunwayML, copy.ai etc. I also used some designing AI tools like Canva, Snappa, Visme for creating my youtube thumbnails and it definitely increased my productivity. Furthermore, I am fascinated by the creative possibilities unlocked by AI tools like Figma AI. These tools have the potential to enhance productivity and extend the boundaries of design. By implementing them, I am confident that I can create innovative and user-friendly UI components that meet the evolving needs of users. I believe that in the near future, AI tools will bring exciting new opportunities and innovation to the design industry, opening up a whole new world of creative possibilities. So, Async API seems like a perfect match for my interests. In future, I would love to explore AI Impact on async api design systems and possibilities.

Apart from this, I am a huge fan of open source in general and how it works by sharing knowledge and collaborating instead of competing with others, I also love to share my knowledge and experiences with the help of my youtube channel - [@vaishu_uff](#), where I share about my college experiences, technical opportunities and my projects. I aim to make more content related to tech and open source in future to increase awareness in my country about the possibilities of open source opportunities. If I contribute well, I would love to share my contributions in this community on my channel, and have podcast or video interactions with other AsyncAPI contributors if possible.

6. Contact information

Name - Vaishnavi Maheshwari

University - National Institute of Technology, Tiruchirappalli, Tamil Nadu, India

Github - [vaishnavi192](#)

LinkedIn - [Vaishnavi Maheshwari](#)

Twitter - [@vaishu_uff](#)

Resume - [Click Here](#)

Email - vaishnavi19official@gmail.com

Contact number - +918878039669

Location - Tiruchirappalli, India, IST(GMT+5:30)

7. About Me

Hi, I am Vaishnavi Maheshwari, currently a second-year undergrad student at [National Institute of Technology, Tiruchirappalli, India](#). I am a keen learner and an open-source enthusiast with a strong interest in web development.

My skills include Javascript, ReactJS, Python, Version control(Git/Github), MySql, flask and designing skills include Figma, Spline3d, MaterialUI, MantineUI, Bootstrap, Framer, Webflow.

I am also a contributor in the processing organization. I learned P5.js library and processing language to do creative coding, made various illustrations and projects based on that. From there I developed a lot of interest in exploring the creative side of coding and visual arts. Earlier I had this mindset that designing and development are made for different types of people, but after seeing the exceptional possibilities of the combination of development and designing, it made me very curious to explore this combination. I selected this project as I am eager to explore the complex component design systems and this project aligns perfectly with my technical stack as it provides an excellent opportunity to delve into the technical aspects of UI development.

In general, I am also very active in open source, I try to participate in as many open-source events as possible, college technical events and hackathons. I participated in SIH 2023, DWOC, [Delta Winter of Code](#) (open source competition like GSOC by students of NIT Trichy) and [TRI-NIT](#) Hackathon. Got 1st rank in DSA contest for girls, and 8th rank in contest among 300 students, hosted by WIN-NIT and Spider, coding Club of NIT-T. Contributed in various organisations like processing, oppia, sugarlabs. Completed Girlsript summer of code'23 and hacktoberfest'23. Participated and mentored in DWOC. I contacted over 60 organizations and maintainers to participate in DWOC. Took part in Outreachy'24, Codeheat challenge by Fossasia, Red Panda community. Attended Global Open Source Cohort in March'24.

I am a Microsoft Learn Student Ambassador and member of Google Developers Student Club, helping students of my college. These program enhances students' employability by offering training in skills not usually taught in academia.

Being a coordinator at [Technical Council \(NIT Trichy\)](#), I coordinated in conducting our college flagship hackathon, Transfinitte. Worked on various campus development initiatives like Hacknitte website, Painnitte(open source question papers uploading platform for college). My recent focus is on the TecOS initiative,

encouraging the open-source culture in our college. We conducted workshops on open source, Git/Github to improve culture in college.

Thanks!

