Development of React Component Library

[1]Prof. Kanchan Umavane, [2]Emaaz Khot, [3]Naufil Shemle, [4]Vivek Varkute

[1]Asst.Professor, [2,3,4]UG Student,1,2,3,4Computer Engg. Dept. Shivajirao S. Jondhle College of Engineering & Technology, Asangaon, Maharashtra, India.

[1]kanchanumavane2020@gmail.com, [2]emaazkhot2002@gmail.com, [2]naufil.shemle@gmail.com, [4]varkutevivek491@gmail.com.

Abstract - Using contemporary JavaScript libraries such as REACT, it is crucial to divide our user interface into more small, more manageable components that are readily reusable and distributable while dealing with vast codebases. In front-end Development mostly each project require similar component created over and over. Then iterate the projects very quickly then make a lot of components to satisfy those project solutions. And there should be some way for user to reduce the amount of time that user use towards making those same components over and over. The solution suggested in this post is to put all of your reusable parts in one location that consistently has the same props. User would want to label your components so that anyone that sees where these components are being made from can easily determine what properties that component uses and what its function is. [1]

Key Words: Reusable Component Library React, Component, Software, Reuse

I. INTRODUCTION

Using contemporary JavaScript libraries such as REACT, it is crucial to divide our user interface into more small, more manageable components that are readily reusable and distributable while working with vast codebases. A pure project has a lot of components on which lots of CSS properties. This is tedious task, to give CSS property to each and every component. It wastes time and dies. So we have created a react library that is open source called 'SSJCET-UI-LIB' which has almost all components. Like button, text area, textbox, dropdown box, checkbox and etc. On this, User can customize the component the way user want it without writing code, All user have to do is to select component select style how user want your component and then copy code and paste it to your project.by using this library user can increased productivity, and increased code reuse and can focus on main project rather than UI. User just have to install the library and import the required components. [1] [2]

II. AIM AND OBJECTIVE

Aim:

The primary goal of the React component library is to streamline the development process by providing developers with a comprehensive set of pre-built, customizable UI components. Developers can economize on time by using Chakra UI instead of writing duplicate code for frequently used interface elements. This allows them to allocate more resources towards crafting engaging and innovative user experiences, ultimately enhancing the overall quality of their applications.

Objective:

A component is HTML and JavaScript combo that has code required to display subset of larger user interface. Such library is meant to hold reusable UI components all in one place so they may be utilized for many applications. A component is reusable UI element combining HTML and JavaScript for displaying parts of user interface. [3]

III. LITERATURE REVIEW

Paper 1: "Designing Code Level Reusable Software Components."

Creating interchangeable elements from other sectors and applying them to the software development industry is the fundamental concept behind developing reusable software components. Reusable components are stored and managed in an organization a reusable library or element reuse repository. The primary benefit of creating reusable software elements have the benefit of reducing the time and effort required to construct any software. By giving users access to a standard operating system, frameworks allow users to concentrate on creating desired modules rather than fine-tuning features. By making use of this resource, developers of software can devote more time to creating software requirements as opposed to setting up application development tools. A framework is a group of reusable software applications that form the basis of an application. Programmers can create applications more quickly with the aid of frameworks. Code reuse is most effectively achieved by sharing similar classes and/or sets of functions, frameworks, and processes. The construction of reusable code-level components and the design of code-level

 DOI : 10.35291/2454-9150.2024.0260

components are covered in this work. Lastly, offering standards, best practices, and coding guidelines for building reusable code-level components as well as best practices and suggestions for modifying and enhancing their usability.

Paper 2: "Paper Name: Reusable Software Components Framework"

Reusability is a notion that is widely used to reduce the cost of software development, effort, and time. Reusability has also been shown to improve software products' productivity, maintainability, portability, and dependability in previous studies involving other software products. The issues in software engineering are not with a lack of reuse per se, but rather with a lack of systematic, broad reuse. but rather with a lack of systematic, broad reuse. They do it informally, but they know how to do it. As a result, this idea needs to be given careful consideration. Build-for-reuse and build by reuse are two systematic approaches to software reuse that are the focus of this study. Next, the reusable components' evaluation requirements are accepted. Ultimately, the C-Registration System is used to implement suggested structure.

Paper 3: "Design and Implementation of Reusable Components Using PowerBuilder."

One important technique for software reuse is component technology. This article explores software reuse technologies based on PowerBuilder, with a focus on component design technology. Reusable components are constructed by first extracting reusable pieces from the application system. Next, a library of reusable components is created using the reusable components. Appropriate components are chosen the reusable library, after which it was instantiated while designing an application system. By assembling the instanced components within a reusable framework, software systems are put into practice. Reusable components can boost software development productivity.

IV. EXISTING SYSTEM

The current method requires a lot of code to be produced, with separate CSS codes created for each component. This wastes time and manpower. Code developed using HTML5, CSS 3. This results in 1000 lines of code, which is too much for the file. When a project's component was to be utilized in another project, the CSS3 code from the previous project had to be copied, resulting in confusion and lost time. As a result, component's actual functionality needs to receive greater focus in the project. Due to constraints of current system, React projects needed to adopt a more standardized and efficient approach to UI development. "SSJCET-UI-LIB" is created in order to these problem and completely transform creation, personalization, and reuse of UI components in React apps. [3] [7]

V. COMPARATIVE STUDY

Sr. No	Author	Project Title	Publication	Technology Used	Purpose
	Anas Bassam AL-Badareen, Mohd Hasan Selamat, Marzanah A. Jabar, Jamilah Din, Sherzod Turaev	Reusable Software Components Framework	13 June 2019	Not specified	Framework for reusable software components
	B. Jalender, Dr. A. Govardhan, Dr. P. Premchand	Designing Code Level Reusable Software Components	1 January 2019	Not specified	Designing reusable software components at the code level
	Mohammad Bajammal, Davood Mazinanian, Ali Mesbah	Generating Reusable Web Components from Mockups	ASE '18, Montpellier, France, 3–7 Sep 2018	HTML, CSS, JavaScript	Generating reusable web components from mock-ups
	This Project	'SSJCET-UI-LIB'	Not Published Yet	React.js, JavaScript, CSS,	Styling libraries (e.g., styled-components)Providing a library of customizable React components for improved UI

VI. PROBLEM STATEMENT

In order to enable developers to modify their components in a matter of minutes, the project aims to create a library of components which are reusable, that are lightweight, easy to implement, responsive, highly adaptable, and modularizable. Developers can provide consistency and enable other users to reuse their work by utilizing component libraries. [7]

VII. PROPOSED SYSTEM

React-component-lib is a simple-to-use library built on React that includes customisable components that may be used to create webpages. The library consists of the following: Box, Radio, Input, Default, Layout, Container, Grid, Notification, Progress Bar, Sub Title, Table, Tag, Text, Title, Checkbox, and so on. [4] Section On storybooks, this library was constructed. One UI development tool is Storybook. This enables working on a single component at a time.. Without having to launch an intricate development stack, import specific data into your DB, or browse your program, user may create whole user interfaces. [5]

Every component in this library can be customized to a great extent. First, the developer can choose the property he wants

 DOI : 10.35291/2454-9150.2024.0260

to change by enabling or disabling the true or false options. Next, he can copy the code for that specific property and incorporate it into his project. [6] A public online repository for libraries is called Node Package Manager (NPM). [7]

VIII. ALGORITHM

Import React from 'react'

Import getStyles function from '../../styles/getStyles'

Import ProgressBarProps and defaultPropsfrom './types'

Define a ProgressBar functional component with props:

ProgressBar(props): Inside the component

Initialize progressBarClassesStr to 'progress,' If props.color is not 'default':

Append 'is-' + props.color + ',' to progressBarClassesStr

If props.size is not 'default':

Append 'is-' + props.size + ',' to progressBarClassesStr Generate CSS classes:

Split progressBarClassesStr by ',' into an array of class names

Call getStyles(array_of_class_names) and store the result in a variable classes

Return JSX: Render a <progress> element with attributes:

className={classes} value={props.value}

- ...otherProgressBarProps (spread the remaining props)

Display the progress value as text inside the <progress> element, e.g., {props.value + '%'}

Set default props for the ProgressBar component:- ProgressBar.defaultProps

Export the ProgressBar component as the standard export of the module

End

IX. MATHEMETICAL MODEL

To describe some aspects, however, various simpler mathematical notations or representations might be used. Remember that these are symbolic representations rather than formal mathematical models:

1. Component Interaction Model: - To depict the interaction between React components, use a directed graph (e.g., G = (V, E)), where V stands for components and E for the data or event flow between them.

2. Formula for Performance Metrics: Usermay use a formula like ResponseTime = ProcessingTime + NetworkLatency to define a basic performance statistic.

In this context, ResponseTime denotes the time required to reply to a request, ProcessingTime denotes the duration of data processing, and NetworkLatency denotes the duration of data transfer via the network.

3. User Interaction Model: A set of equations or a state transition matrix can be accessed to characterize user interactions. As an illustration:

UIState(t+1) = UIState(t) + UserAction(t)

where UserAction(t) denotes the user's action at time t, UIState(t+1) denotes the UI state following the user's action, and UIState(t) represents the UI state at time t.

4. Code Quality Metric: Cyclomatic Complexity (V(G)) is one mathematical formula that can be accessed to determine code quality. As an example: V(G) = E - N + 2P

X. SYSTEM ARCHITECTURE

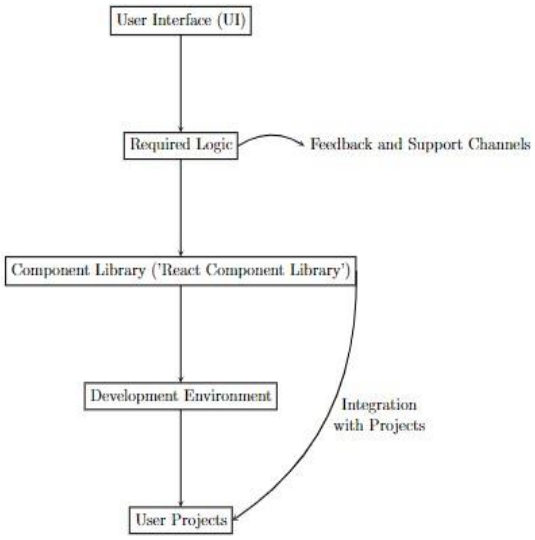

Fig1. System Architecture

User Interface: Refers to the user interface elements and design of the application, including buttons, images, forms, and layouts, providing an eye-catching and intuitive experience for users.

Required logic: The essential logic and functionalities needed to implement the features and behaviour of the application, ensuring it meet project requirement and goals.

Development Environment: The collection of instruments, libraries, and frameworks used by developers to create, test, and deploy the software, providing an efficient and productive workflow for building the application.

User project: The specific project or application being developed by the user, which utilizes the UI, logic, and development environment to achieve its goals and serve its intended purpose.

XI. ADVANTAGES

Reuse UI elements to maintain team independent thinking.

Greatly enhance UI/UX continuity.

Develop new UI applications quicker.

Lower the weight and bundle size of apps.

Lower maintaining expenditures and software development.

 DOI : 10.35291/2454-9150.2024.0260

Enhance output.

Make software that is more standardized.

XII. DESIGN DETAILS

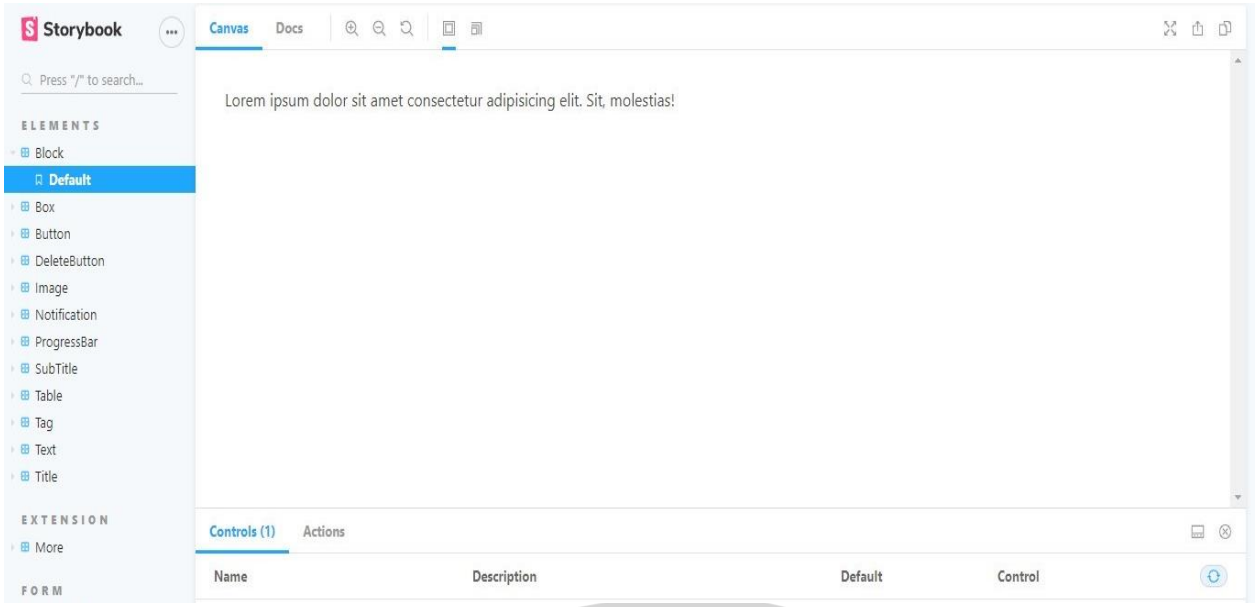

Fig1. Home Page

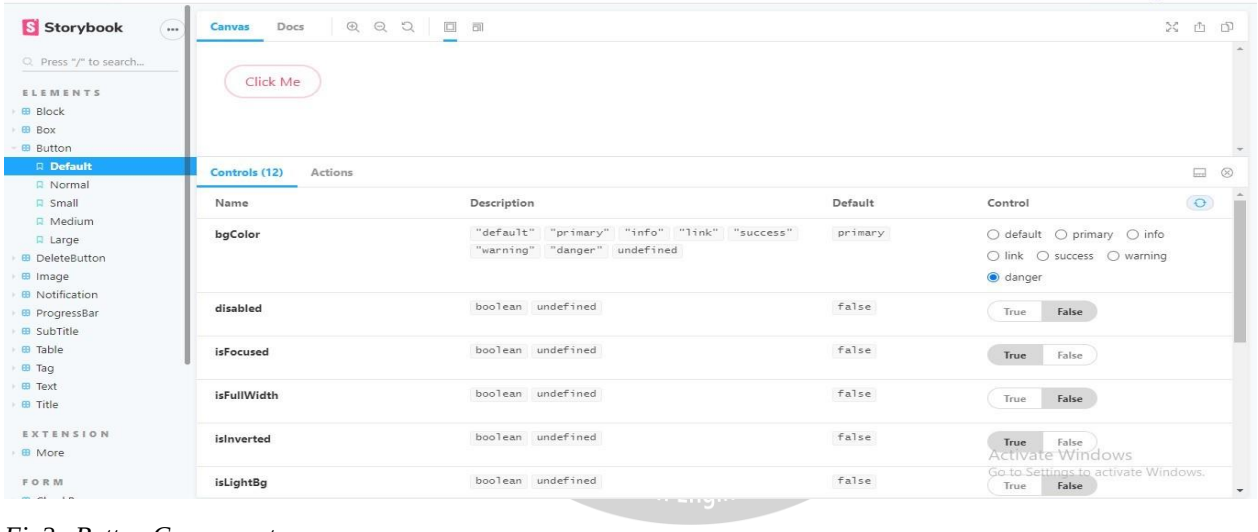

Fig2. Button Component

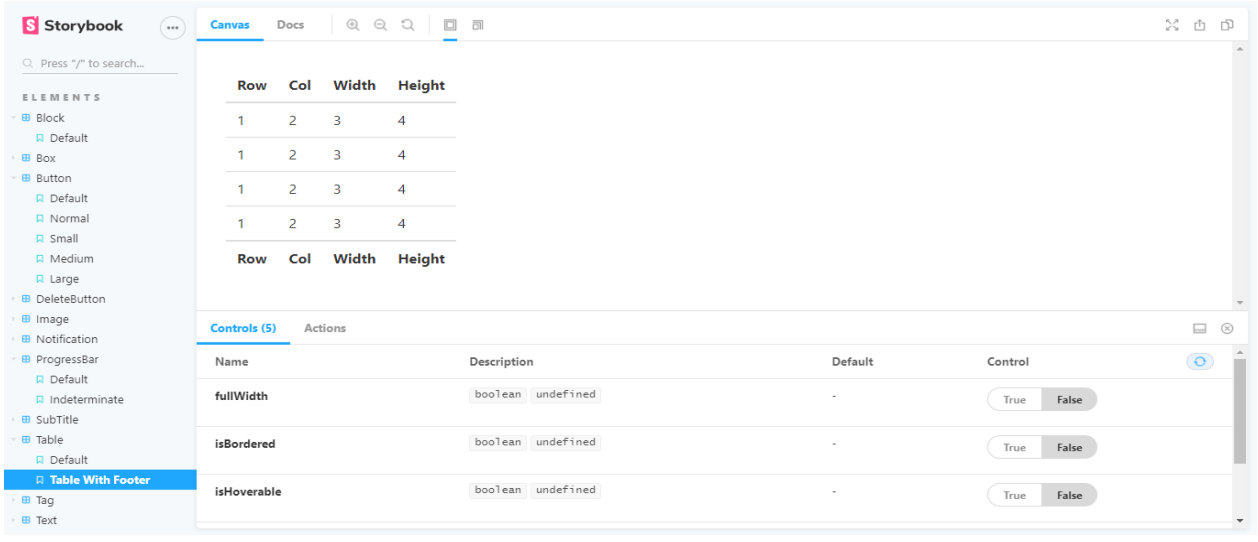

Fig 3: Table

 DOI : 10.35291/2454-9150.2024.0260

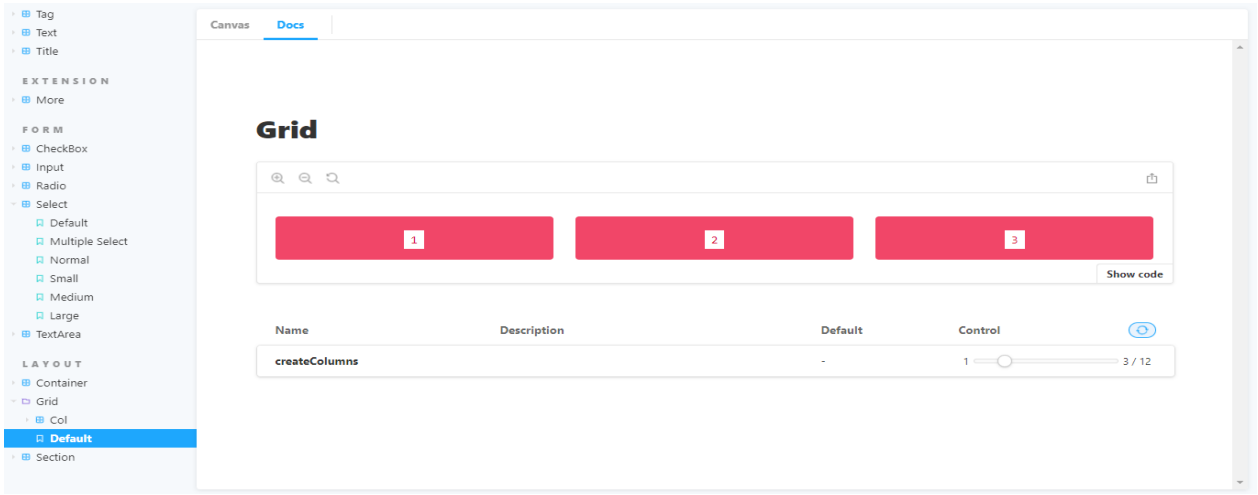

Fig 4: GRID

CONCLUSION

Thus, this paper elaborates the concept brought up by the library was to consolidate all of your reusable parts that consistently contain the same props into one location. To that, User should also document your components so that anyone who notices where they are being built can quickly ascertain what props are used in each component and what their purpose is. A developer can customize a library's single, lightweight, responsive, easily-manageable, highly adaptable, and modularizable reusable component in under a minute. Developers can achieve uniformity and allow for reuse by using Component Libraries.

XIII. REFERENCES

[1] Designing Code Level Reusable Software Components. Author Name: B.Jalender 1,Dr A.Govardhan 2, Dr P.Premchand 31 Asst Professor, Department of IT, VNRVJIET, Hyderabad, India-500090 2 Professor in CSE & Director of volution, JNTUH, Hyderabad,. 3 Professor ,CSE Department, UCEOU, Osmania University, Hyderabad. Date of Published: 1, January 2019

[2] Reusable Software Components Framework Author Name: Anas Bassam AL-Badareen, Mohd Hasan Selamat, Marzanah A. Jabar, Jamilah Din, Sherzod Turaev Date of Published: 13 June 2019.

[3] Generating Reusable Web Components from Mockups, Name: Mohammad Bajammal, Davood Mazinanian, and Ali Mesbah ASE '18, Montpellier, France, Date: September 3–7, 2018,

[4] https://bulma.io/documentation/components/\

[5] https://blog.bitsrc.io/do-we-really-use-reusable-components-959a252a0a98

[6] https://blog.usejournal.com/should-you-build-a-reusable-component-library-4e7df72413d7

[7] https://medium.com/thron-tech/why-we-made-a-new-component-library-instead-of-buying-one-2335009a0895

[8] Mohammad Bajammal and Ali Mesbah. 2018. Web Canvas Testing through Visual Inference. In Proceedings of the International Conference on Software Testing, Verification and Validation (ICST). IEEE Computer Society.

 DOI : 10.35291/2454-9150.2024.0260