成都理工大学中英合作办学
CDUT Sino-British Collaborative Education

OXFORD BROOKES UNIVERSITY

UNDERGRADUATE PROJECT REPORT

Project Title:	**Web Design Platform Based on Low-Code Visual Components**
Surname:	**Ruitong**
First Name:	**Xiao**
Student Number:	**201918010106**
Supervisor Name:	**Javier Jia**
Module Code:	**CHC 6096**
Module Name:	**Project**
Date Submitted:	**May 5th, 2023**

Chengdu University of Technology Oxford Brookes College

Chengdu University of Technology

BSc (Single Honours) Degree Project

Programme Name: **Computer Science**

Module No.: **CHC 6096**

Surname: **Xiao**

First Name: **Ruitong**

Project Title: **Web Design Platform Based on Low-Code Visual Components**

Student No.: **201918010106**

Supervisor: **Javier Jia**

2ND Supervisor (if applicable): **Not Applicable**

Date submitted: **May 5, 2023**

A report submitted as part of the requirements for the degree of BSc (Hons) in Computer Science

At

Chengdu University of Technology Oxford Brookes College

Declaration

Student Conduct Regulations:

Please ensure you are familiar with the regulations in relation to Academic Integrity. The University takes this issue very seriously and students have been expelled or had their degrees withheld for cheating in assessment. It is important that students having difficulties with their work should seek help from their tutors rather than be tempted to use unfair means to gain marks. Students should not risk losing their degree and undermining all the work they have done towards it. You are expected to have familiarised yourself with these regulations.

https://www.brookes.ac.uk/regulations/current/appeals-complaints-and-conduct/c1-1/

Guidance on the correct use of references can be found on www.brookes.ac.uk/services/library, and also in a handout in the Library.

The full regulations may be accessed online at https://www.brookes.ac.uk/students/sirt/student-conduct/

If you do not understand what any of these terms mean, you should ask your Project Supervisor to clarify them for you.

I declare that I have read and understood Regulations C1.1.4 of the Regulations governing Academic Misconduct, and that the work I submit is fully in accordance with them.

Xiao RuiTong

Signature Date 05/05/2023

REGULATIONS GOVERNING THE DEPOSIT AND USE OF OXFORD BROOKES UNIVERSITY MODULAR PROGRAMME PROJECTS AND DISSERTATIONS

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Oxford Brookes University Library.

I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing the use of the Oxford Brookes University Library.

Xiao RuiTong

Signature Date 05/05/2023

Acknowledgment

Express my heartfelt gratitude to Dr. JoJo Walker, as my module leader, and Mr. Javier Jia, as my supervisor, for their unwavering support and expert guidance throughout the semester. Their invaluable knowledge and encouragement have been instrumental in shaping my learning journey.

Also, I'm deeply appreciative of my peers for creating a stimulating and interactive learning environment, and my family for their unceasing support. To everyone who has helped me along this journey, I am sincerely thankful.

Table of Contents

Abstract

The rapid development of applications and constant iteration has contributed to the exponential growth of low-code development (LCD). Traditional application development is often associated with excessive costs, inefficiencies, and a shortage of highly qualified developers. In contrast, low-code development platforms (LCDPs) are an attractive solution due to their ability to address these issues. Gartner [10] and Forrester [4] predict that LCDPs will be increasingly useful in the future, with several giant companies such as Google App Maker, Outsystems, and Salesforce Platform launching LCDPs recently.

This report presents an LCDP based on visual components, which enables citizen developers to design websites with minimal hand-coding. The aim is to provide an alternative solution that reduces costs, increases efficiency, and accelerates time-to-market. The report includes a summarized list of low-code platform features and uses it as a benchmark for comparing three commercial LCDPs. The critical analysis enables a detailed comparison of the platforms and helps to identify their strengths and weaknesses.

Ultimately, the project provides citizen developers with a method to design applications with visual interfaces, minimal coding, and prebuilt components, resulting in increased efficiency, reduced costs, and accelerated time-to-market. In today's market, it has emerged as a viable alternative to traditional application development, and its benefits can be applied to a wide range of organizations, regardless of their size and industry sector.

Keywords: Low-Code Development, Visualization, Drag and Drop, Digitized applications, Citizen Developers, Model Driven Engineering

Abbreviations

Abbreviations	Definition
LCDP	Low-Code Development Platform
LCD	Low-Code Development
IT	Information Technology. A general term for the various technologies used primarily to manage and process information
GUI	Graphical User Interface. Computer-operated user interface with graphical display
UI	User Interface. Overall design of the software's human-computer interaction, operation logic, and interface aesthetics
RAD	Rapid Application Development. Software development methodology that emphasizes rapid prototyping and short development cycles.
IDC	International Data Corporation. A market research and consulting firm that assists organizations in making informed technology investment decisions.
IDE	International Data Corporation. It is a software application designed to assist developers in creating, debugging, and deploying applications quickly.
SDLC	Software Development Life Cycle. A methodology for developing software that emphasizes rapid prototyping and short development cycles.
URL	Uniform Resource Locators.
SEO	Search Engine Optimization. It is the process of optimizing all aspects of a website to achieve higher visibility and traffic in the natural search engine rankings.
FCP	First Contentful Paint.
LCP	Largest Contentful Paint.
TBT	Total Blocking Time.
CLS	Cumulative Layout Shift.
SI	Speed Index.
GDPR	General Data Protection Regulation.
IEEE	Institute of Electrical and Electronics Engineers

Glossary

Low-Code Development (LCD): Low-code development is a visual approach to software development that enables faster delivery of applications through minimal hand-coding.

Traditional Application Development: The traditional method of application development involves the development of software by a software engineer from scratch.

Citizen developers: Citizen developers are business users who create or modify applications using accessible tools to bridge the gap between IT departments and the organization, allowing the organization to remain competitive.

Chapter 1 Introduction

1.1 Background

The rapid development of applications and the constant iteration of applications has become a new trend. Based on statistics, the average cost of an application development project typically falls within the range of US$434,000 and US$2,322,000 [1], however 52.7% of projects will ultimately cost 89% more than anticipated, with only 16.2% of projects completing on schedule, while even 31.1% of projects being abandoned before completion [1]. Aside from that, as a result of a 38% reduction in job growth in 2021, and a constant updating of market demand in the IT field [2], Gartner predicts that by 2021, the demand for information systems is expected to rise five times faster than the capacity of IT departments to meet it [10]. In this case, thousands of job vacancies might be created in the next few years, as recruitment of development engineers is becoming progressively challenging, companies are facing the challenge of unable to meet the increasing market requirements for new applications in time and having to establish a quicker and more cost-effective method to accommodate their application requirements [2]. To address this challenge, a concept called "Low-code" has been coined by Forrester Research in 2014 [4], which is an attractive option to reduce development costs and accelerate time to market, has become a preferred software development solution for companies.

In accordance with the concept of "Low-code", Low-Code Development Platforms (LCDP) is a development platform that provides a graphical user interface (GUI) and integrated development environment (IDE) for rapid application development (RAD) [11], which enables users to create software applications without coding [10]. With a growing trend of Low Code Development Platforms (LCDPs) which serve as an alternative solution to the shortage of highly qualified professional developers by allowing "individuals without formal programming training", i.e., citizen developers [3], to participate in the application development process through high-level abstractions and models [12]. According to a concept proposed by Forrester in 2014, LCDPs allow non-programmers and citizen developers to build applications by dragging and dropping components, visual modelling, and generating code automatically [4]. Aside from this, LCDPs contribute to enhancing application development efficiency and significantly reducing the costs of traditional application development, which address the excessive costs and inefficiencies associated with traditional application development [5]. Research [6] shows that low-code development is also one of the most effective solutions to the skills shortage of developers,

as it not only decreases development costs and time, but it also produces high-quality products as a result.

Currently, several major corporations have already released LCDPs, including Google, Microsoft, and Mendix [13]. By 2023, the market for LCDPs is predicted to grow sharply, with more than half of medium to large enterprises adopting LCDPs as one of their top strategic application platforms [7].

1.2 Aim

This paper presents a low-code development platform based on visual components, which enables users to create applications by dragging and dropping elements on a canvas through a visual interface, providing citizen developers with an opportunity to rapidly and economically design websites, which alleviates the high costs and inefficiencies associated with traditional application development, as well as the shortage of developers in the industry.

1.3 Objectives

a. Requirements Analysis and Planning Phase.
b. Incremental Development Phase 1: Project Structure Building.
c. Incremental Development Phase 2: Component Library Area Module Refinement.
d. Incremental Development Phase 3: Navigation Bar Module Refinement.
e. Incremental Development Phase 4: Editing Canvas Area Module Refinement.
f. Incremental Development Phase 5: Component Configuration Area Module Refinement.
g. Incremental Development Phase 5: Component Configuration Area Module Refinement.
h. Incremental Optimization and Website Testing Phase.

1.4 Project Overview

1.4.1 Scope

The project provides a low-cost, rapid and low-threshold application development service that significantly reduces the amount of programming required by facilitating users with a graphical user interface and drag-and-drop component design that enhances the

user interaction experience, thereby non-programmers like citizen developers are able to participate in the development of the website.

1.4.2 Audience

The platform targets a key audience of citizen developers who seek to develop applications with reduced programming efforts. While professional developers rely on traditional programming to build solutions, citizen developers are essentially business users who lack experience in application development.

As a low-code platform, it provides them with an easy-to-use, code-free development experience, regardless of their previous coding experience. With this approach, citizen developers can quickly and efficiently create applications without the need for specialized programming skills.

Chapter 2 Background Review

This section outlines an overview of LCDPs and selects three representative leading platforms for further analysis and comparison. The comparative analysis is based on a set of features that include visualization, deployment, interoperability, collaboration, reusability, security, development mechanisms, and types of supported applications. For effective compare features of each platform, Table 1 provides a description of each function, while Table 2 presents a summary of the features supported by each platform, which shows the results obtained predominantly from the official resources of each platform indicated by [16], [21], and [22], as well as from the experience obtained from the analysis of these leading applications.

Due to the high effectiveness [5] and low threshold [10] of LCDPs, more and more enterprises have turned to low-code development platforms in recent years. As evidenced by the January survey by research firm IDC of 380 enterprises, which indicated that 48.6% of respondents were purchasing low-code or no-code platforms, and Gartner's research reveals that low-code software development platforms are being adopted at a rate exceeding 20% per year, resulting in a global market for low-code development technology expected to reach $13.8 billion by 2021 [15]. As a result of the growing user demand, there has been a proliferation of low-code platforms on the market that have their own characteristics.

Furthermore, investigations have shown that LCDPs by themselves do not provide much innovation in most LCDPs, while their innovation is generally derived from customizations and integrations [5]. In this case, despite the capacity for LCDPs to optimize productivity during application development and execution, it is imperative that users carefully evaluate the features and functionalities of each platform before selecting one. In the following sections, a critical analysis of three low-code development platforms was conducted to determine their essence based on their characteristics and challenges.

Mendix [16] is one of the most prominent low-code development platforms, along with Microsoft PowerApps, OutSystems, Salesforce, Lansa, Zoho creators and Quick base [4]. It is utilized by enterprises such as Philips, Bosch, and Siemens to quickly design and deploy applications, has provided drag-and-drop development functions with a rich library of components to support visual development [4]. Furthermore, the platform provides users with the capability to integrate external data sources or services into the application using pre-built connectors, thus allowing it to access and exploit information from external

sources, and create context-aware apps such as cognitive services, machine learning and IoT [17]. As compared to its competitors, it suffers from a modest implementation partner network [8], which means that fewer enterprises or organizations are involved in its implementation. Moreover, it covers only basic file management and storage in content management [8], resulting in the absence of advanced content management features such as version control, workflow management, and collaboration tools.

According to a study comparing Mendix, Microsoft PowerApps, and OutSystems, it was found that the three platforms shared similar characteristics in terms of their development capabilities [18]. In the same way as Mendix, Microsoft Power Apps [21] also provides prebuilt connectors for creating context-aware applications like Mendix [18]. In addition to this, they both provide an Integrated Development Environment (IDE) that supports graphic modeling, rapid application development, and code generation [4] [18]. The difference is that Microsoft Power Apps has a competitive advantage due to integration with many services in the Microsoft ecosystem such as Azure, Excel, Outlook or similar connectors to legacy systems [18]. Despite that, as well as supporting cloud deployment [18], it supports a wide range of applications such as quality management, event monitoring, approval process control, process automation, workflow management, inventory management, and escalation management [17]. However, the platform is not without flaws. One major downside to Microsoft Power Apps is limited customization. Although PowerApps offers a range of pre-built components and templates, there are limited customization options when compared with other low-code platforms, which might make the development of highly customized applications more challenging [18]. Furthermore, the limited scalability of Microsoft PowerApps may prevent it from handling the complexity and data volumes required by large enterprise applications [18].

Another low-code development platform mentioned above is OutSystems [22], which offers visual application logic development, integration with existing systems, and business process automation. Its excellent performance in the low-code field makes it a strong competitor to other leading platforms, such as Mendix and Microsoft Power Apps. As a cloud-based LCDP similar to Microsoft Power Apps, it can be accessed remotely from any device with an internet connection, which makes remote collaboration and project completion a breeze [18]. A further benefit of OutSystems is that it facilitates agile development as well as DevOps deployment, thereby making it possible for developers to efficiently and rapidly build and deploy applications [19]. Not only that, OutSystems

offers an integrated development environment (IDE) to support the entire software development process from design to deployment [18]. Nevertheless, one potential drawback of OutSystems is that the scope of its developer community is still too limited, concentrating mainly on discrete application patterns and prototypes, as well as its sub-par content management, which is another weakness it has in common with its rival Mendix [8]. Additionally, based on the results of the study [20], the definition of a non-directional model transformation strategy implemented by OutSystems impedes interoperability between various development tools within the ecosystem of Out Systems.

Feature	Description
	Visual user interface
Drag-And-Drop Application	This feature allows users to quickly and easily move elements around on the screen without having to manually type in code.
Multi-Type Forms	This feature allows users to create forms with multiple types of data inputs, such as text fields, drop-down menus, radio buttons, and check boxes. Having multiple types of data inputs makes it easier for users to create forms that are more user-friendly and visually appealing.
Advanced Abstraction Model	This allows users to quickly and easily create complex forms by abstracting the underlying data into simpler, more intuitive components.
Progress Tracking	It allows users to monitor their form creation progress, allowing them to make changes whenever necessary.
	Deployment
Cloud Deployment	Cloud deployment enables users to deploy their applications quickly and easily, allowing them to access their applications securely and reliably anywhere.

Local Infrastructure Deployment	This feature provides users with greater control and flexibility over the deployment of their applications by allowing them to deploy them on their own local infrastructures.
Interoperability	
External service interoperability	This feature enhances the ability of users to integrate their applications with external services such as cloud databases or analytics services, enabling them to maintain control over their applications while taking advantage of the power of these services.
Data connection	This feature provides users with the ability to synchronize data from a variety of sources, including cloud services, databases, and APIs.
Collaborative development	
Off-line collaboration	Collaboration without an Internet connection.
On-line collaboration	Online collaboration refers to several individuals cooperating on a project remotely via the Internet.
Reusability	
Pre-built modules	Applications consist of these components that enable developers to build more complex applications.
Built-in workflows	This feature facilitates the development and execution of workflows by developers.
Security	
Application security	Application security refers to the security measures taken to ensure the integrity and privacy of computing applications.
Platform security	The term refers to the protection of the infrastructure and foundation platforms upon which low-code applications are based, as well as encryption, access control, data encryption, and vulnerability management.

Development mechanisms	
Code generation	The process of turning a low-code model into a ready-to-use application, which is handled automatically by the platform, results in code that runs efficiently on the target platform.
Supported application types	
Quality management	It refers to the process of managing the quality of software with the objective of improving application quality and meeting user requirements.
Event monitoring	It is the monitoring and management of events occurring in a system in order to identify and address problems promptly.
Approval process control	It refers to the process of ensuring the compliance and accuracy of the approval process.
Process automation	The term refers to a technology used to automate business processes in order to improve work efficiency and accuracy through automation.
Workflow management	It refers to the management of workflow to ensure smooth and efficient operations.
Inventory management	It can be utilized to track inventory levels, monitor stock movement, and generate reports to make better-informed decisions, while also helps to reduce costs associated with manual data entry and inventory shrinkage.
Escalation management	It involves setting up processes to identify and address potential issues before they become full-blown problems.

Table 1. Functional overview of LCDPs

Feature	Mendix	Microsoft Power Apps	OutSystems
Visual user interface			
Drag-and-drop application	✓	N/A	✓
Multi-type forms	N/A	✓	N/A
Advanced Abstraction Model	✓	✓	✓
Progress tracking	✓	✓	✓
Deployment			
Cloud deployment	✓	✓	✓
Local infrastructure deployment	✓	N/A	✓
Interoperability			
External service interoperability	✓	✓	✓
Data connection	✓	✓	✓
Collaborative development			
Off-line collaboration	✓	✓	✓
On-line collaboration	✓	N/A	✓
Reusability			
Pre-built modules	✓	✓	✓
Built-in workflows	N/A	N/A	N/A
Security			

Application security	✓	✓	✓
Platform security	✓	✓	✓
Development mechanisms			
Code generation	✓	✓	✓
Supported application types			
Quality management	✓	✓	✓
Event monitoring	✓	✓	✓
Approval process control	N/A	✓	N/A
Process automation	N/A	✓	✓
Workflow management	✓	✓	✓
Inventory management	✓	✓	✓
Escalation management	N/A	✓	N/A

Table 2.Comparison of Analysed LCDPS [4], [8], [16], [18], [21], [22]

Chapter 3 Methodology

3.1 Approach

3.1.1 Software Development Methodology

In software development and systems engineering, incremental model refers to a methodology for designing, developing, and delivering software products that decomposes software requirements into discrete, incremental building blocks or independent modules in the software development life cycle (SDLC), in which each component is developed, implemented, and tested independently [23].

Considering the flexibility of the incremental model in terms of project management and testing as well as its cost effectiveness, the incremental model was chosen to be the software development method for this project, decomposing the project into multiple interrelated task modules in accordance with the various functional requirements. In order to facilitate a better project management, it is necessary to develop a project plan before beginning project development, outlining the task modules to be performed and establishing a timeline for these task modules. The order of completion of the task modules depends on the priority of the functionality involved and the user requirements, while the functionality with the highest priority is delivered first. In response to this, the initial task module will include the core functionality of the project, with subsequent task modules extending that functionality and continuously adding additional functionality.

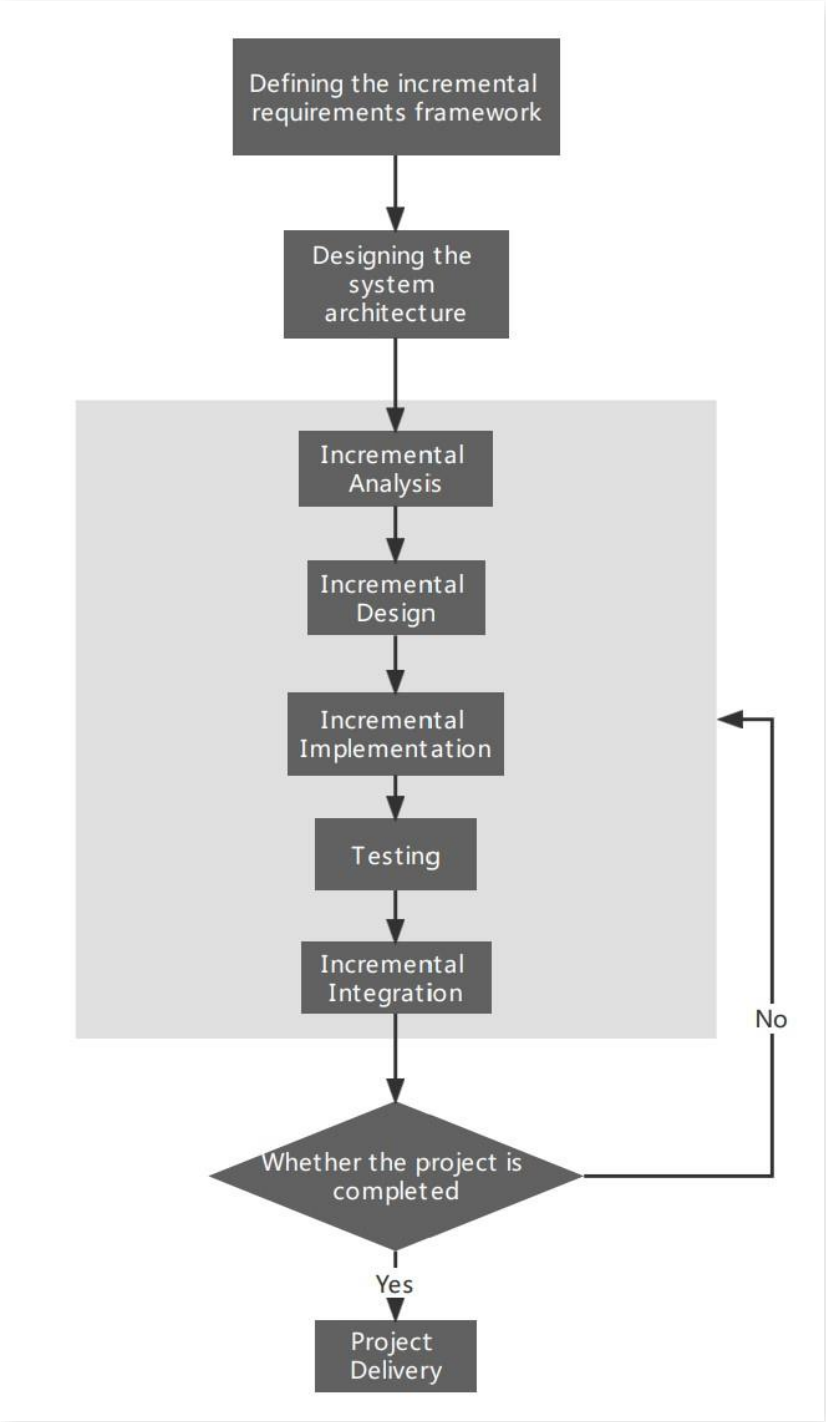


Figure 1.Incremental Development Model Flowchart

3.1.2 Requirements Gathering Methods

The requirements gathering and analysis methodology used in this project was based on the project aim, project overview, and background review outlined in Chapter 1

and Chapter 2 of this paper, in order to identify the necessary project features and functionality. For requirement gathering, a competitive feature analysis method was adopted, which involved analysing and comparing the features and functions of competing products in the same field to gain a thorough understanding of the product landscape.

This analysis served to identify the strengths and weaknesses of existing products and to obtain guidance from their product models, features and architectures. Through reflecting on the substantive similarities of competing products to integrate the effective features and functions into the product of this project, while develop an optimal functional plan. Ultimately, this process led to the development of an innovative and efficient solution that met the project requirements.

3.2 Technology

3.2.1 Project Design Tools

To organise and construct the outline structure and the core modules, this project relies on the mind mapping tool XMind. It is a powerful and easy-to-use mind-mapping software, which enables users to create visual diagrams to help them build and achieve their ideas easily. As it facilitates to break down complex ideas into smaller, more manageable chunks, it is also a great tool for project design.

The flowchart for the incremental development model was drawn with the online collaborative mapping platform ProcessOn, which is an online collaborative mapping platform that facilitates the creation and sharing of flowcharts, process diagrams, as well as mind maps through an intuitive interface. It allows multiple users to collaborate on the same project from different locations and sharing diagrams, thus making it easy to keep everyone updated.

The project's UI prototypes were created through the application of the UI design tool Figma, a cloud-based design platform that enables real-time collaboration, prototyping, and iteration on design projects. It promotes collaboration of one project between multiple designers, developers, and stakeholders, and offers an extensive library of UI elements and components for creating interactive prototypes.

3.2.2 Project Development Tools

This project is built and developed using Vue and Vite. Among them, Vue is an open-source JavaScript framework for building user interfaces and single-page applications, which is designed to be easy to use and flexible, allowing developers to

create interactive web applications with a minimal amount of code. Moreover, Vite is a fast-build tool that integrates with Vue, providing instant feedback during development so that developers can make changes quickly and easily.

In order to implement the technologies, the main development tool for this project is Visual Studio Code, which is an open-source, cross-platform code editor designed for building and debugging modern web and cloud applications. It is equipped with numerous built-in debugging tools and supports a number of plug-ins and extensions to meet a variety of development requirements, thereby facilitating the development of projects.

3.2.3 Project Testing Tools

For functional testing, this project employs Selenium IDE and PyCharm as testing tools. Test cases for corresponding functions are generated in Selenium IDE, exported as Python files, and run in PyCharm using Pytest. Among them, Selenium IDE is a web automation tool that records and plays back tests, and it supports a wide variety of browsers, such as Internet Explorer, Chrome, Firefox, and Safari. Aside from this, PyCharm is an IDE for Python that supports syntax highlighting, code completion, debugging, and unit testing. In particular, Pytest is a unit testing framework written in Python-based language. It offers simple assert statements for testing, provides a rich plug-in mechanism, and is easily integrated with continuous integration systems.

For performance testing, this project utilized Lighthouse to test the website's loading speed, page content optimization, JavaScript performance, and mobile adaptability. Specifically, Lighthouse is an automated tool developed by Google that evaluates website performance by simulating real users' behaviour during a web page's visit, and provides a series of recommendations and optimization solutions that assist developers in improving website performance, accessibility, best practices, and search engine optimization.

3.3 Project Version Management

Use GitHub to manage project versions and project-related files, and update them after each development phase.

GitHub Repository: AuroraXiao/Web-design-platform-based-on-low-code-visual-components: 201918010106 Aurora (github.com)

Chapter 4 Results

4.1 Project Implementation

4.1.1 Project Structure

The project can be divided into four main modules, as shown in Table 3. The first module is the navigation bar, which appears at the top of the website layout and provides various options, such as page editing and previewing. The second module is the component library area, located on the left side of the website layout, which provides currently integrated component libraries such as Element-UI, Vant, and Ant Design, as well as basic HTML elements. The third module is the editing canvas area, located at the bottom right of the layout, which offers logical code editing (JavaScript editing), secondary editing (Vue editing), real-time code preview, and page deletion functionality. Lastly, the fourth module is the component configuration area, which is essentially a configuration window that appears when the user selects a component. In this module, there are two modes of property editing: binding editing mode and free editing mode. Based on these modes, attributes of the selected component can be added, updated, or deleted.

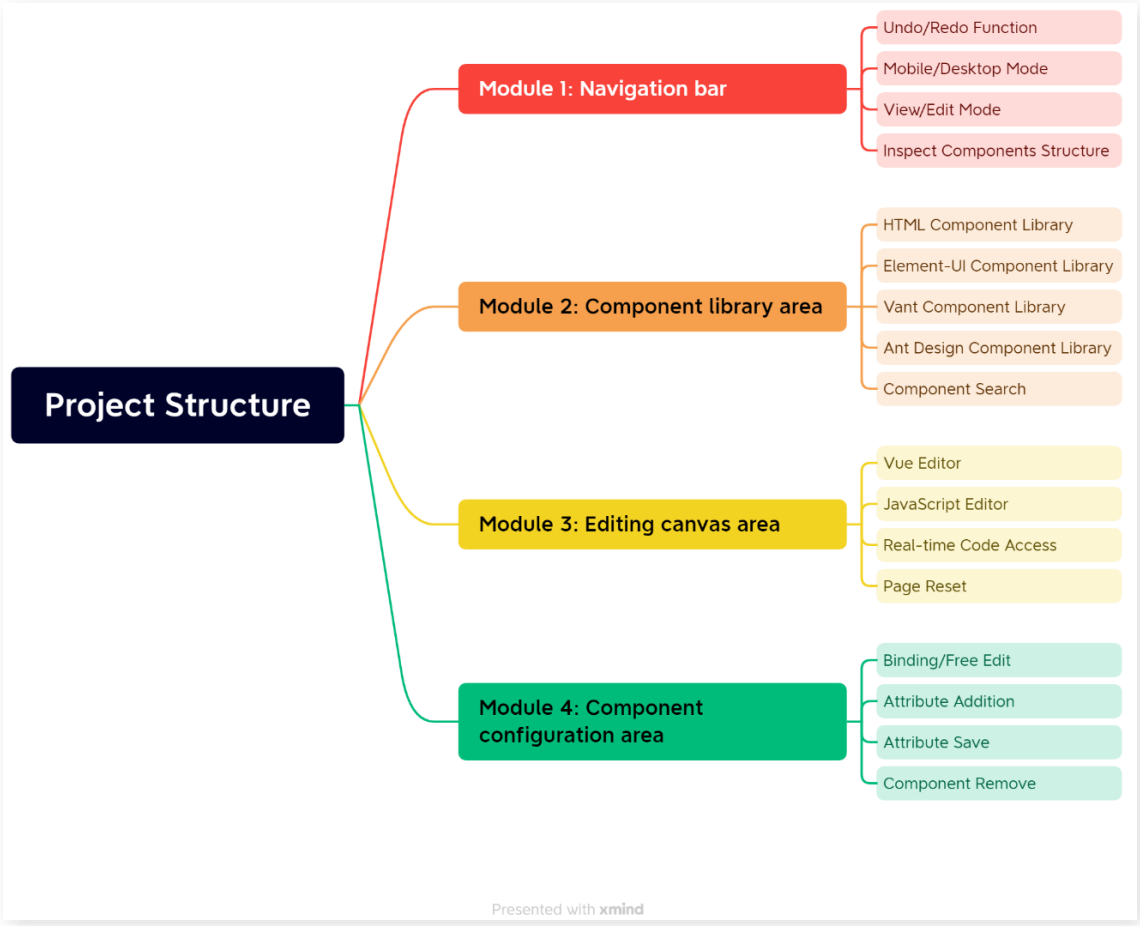
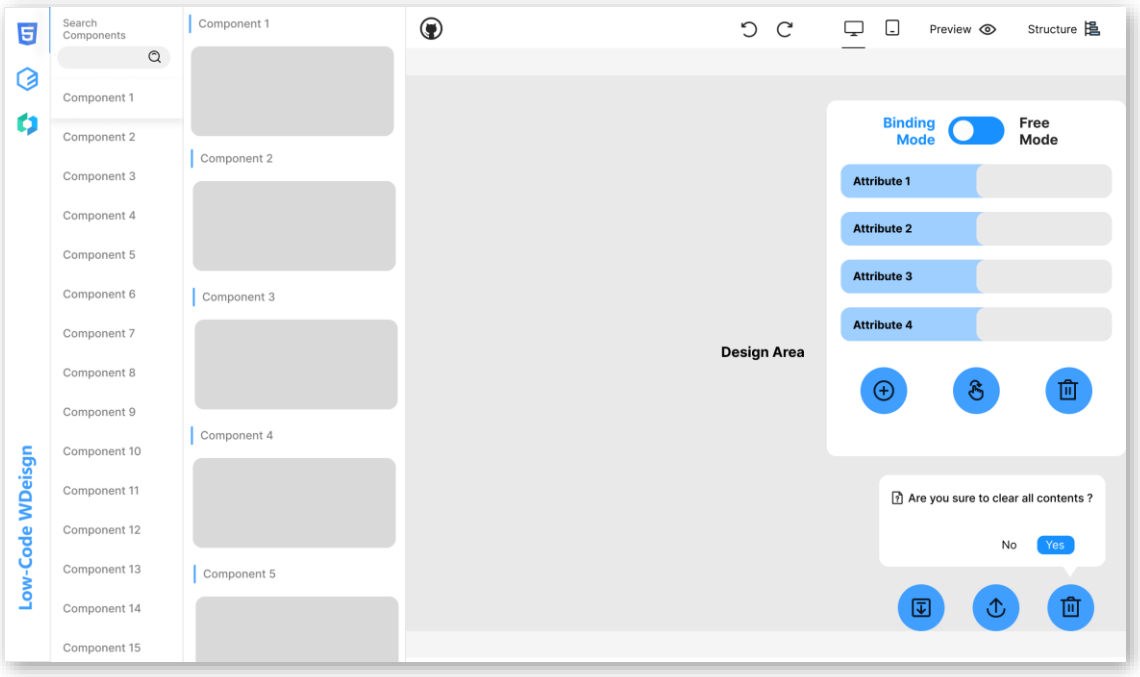


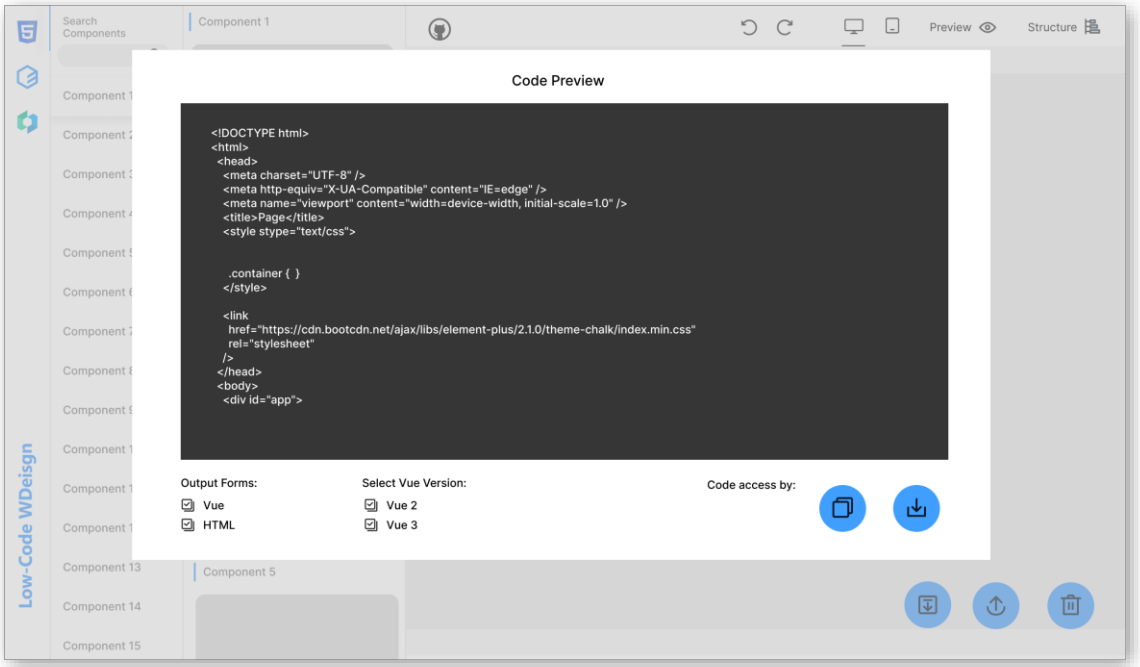
Figure 2. Project Structure

4.1.2 Project Design

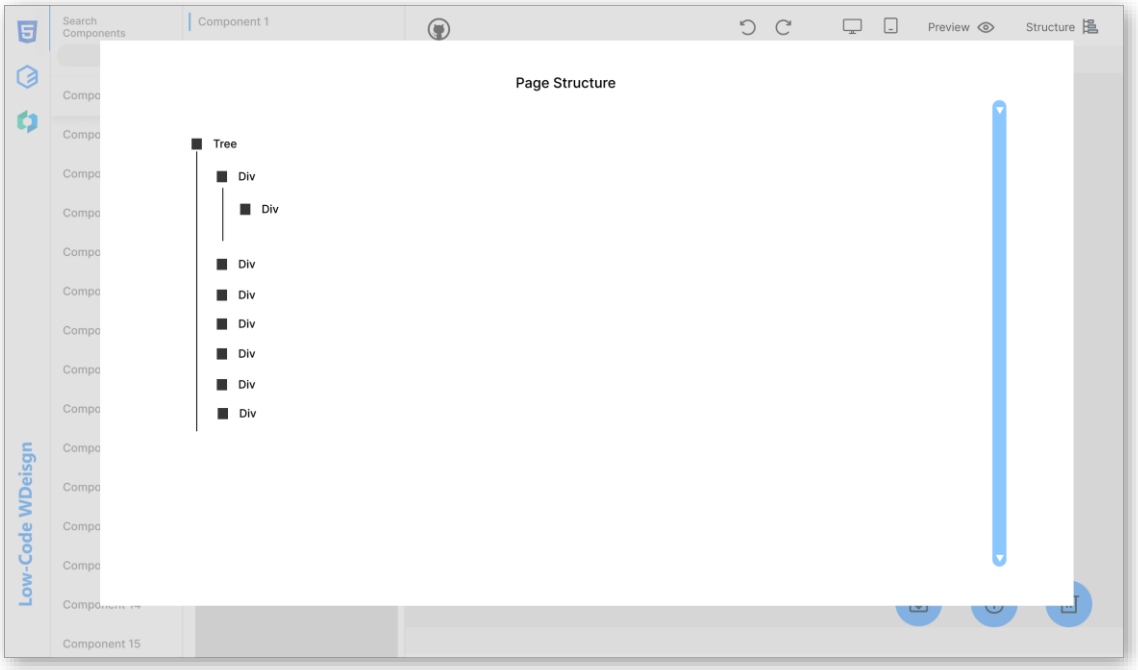
The initial design concepts for the project website are illustrated in the following UI prototype diagrams. The prototypes are simple frameworks to show the layout of the pages and the interaction of the various features in order to provide a more visual representation of the product's initial conception. In developing this project, the UI prototypes provided a comprehensive understanding of the product interaction logic, enabling proactive anticipatory and proactive resolution of potential problems.



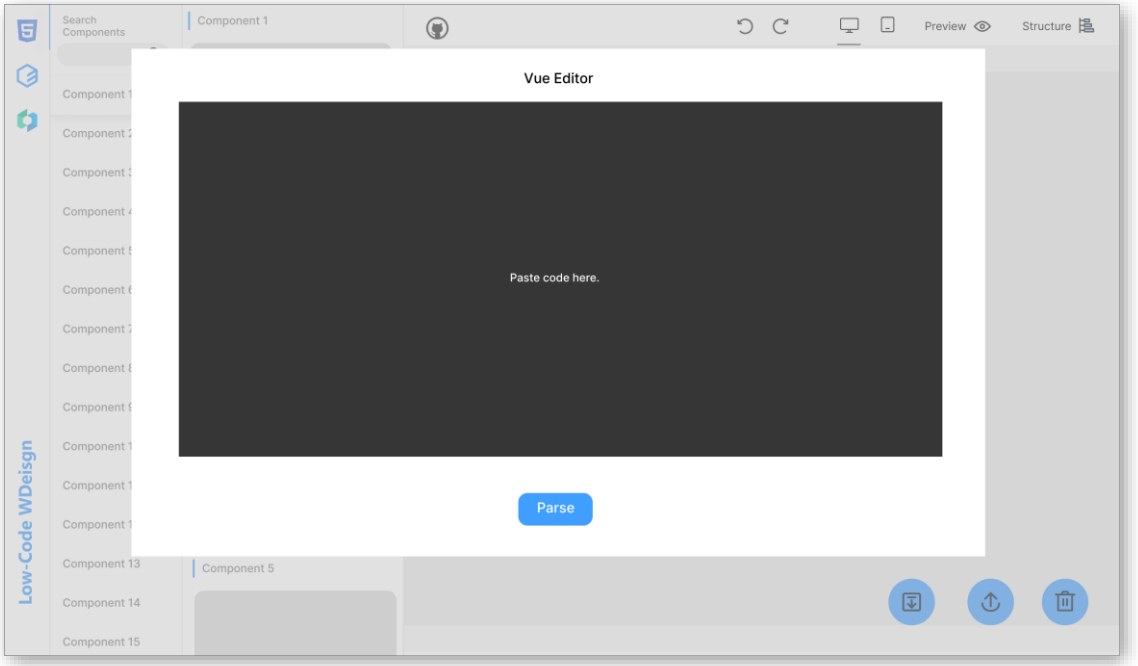
Screenshot 1. Editing Page



Screenshot 2. Code Preview



Screenshot 3. Page Structure



Screenshot 4. Vue Editor

4.1.3 Project Results

4.1.3.1 Function Overview

According to the classification of the project modules and the main functions of the modules mentioned in 4.1.1 Project Structure, this section will detail the purpose and role of the project functions mentioned above. In order to show the content of the functions more intuitively and explicitly, the description of the project functions will be represented by the Table 5 below.

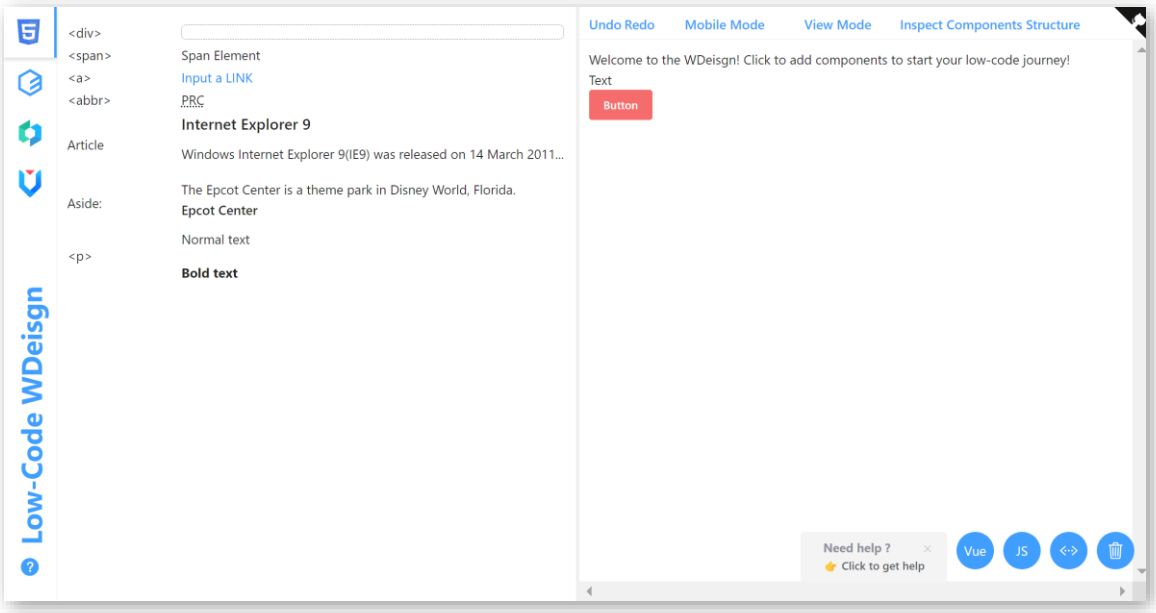
Implemented Module	Function Name	Function Description
Navigation bar	Undo/Redo Function	The function to manipulate the components in the editing canvas area, which allows users to undo and redo their operations in the editing canvas area.
	Mobile/Desktop Model	Enables visual pages to be compatible with different devices.
	View/Edit Mode	Provide visual page preview function, click Esc key to exit full screen preview.
	Inspect Components Structure	Display the hierarchical structure of the visual pages.
Component Library Area	HTML5	Provide basic drag-and-drop elements such as <p>, <a>, <h4>, etc. To use components, users can simply select the desired component and drag it to the edit canvas area.
	Element-UI	Three general component libraries are integrated and users can click on the quick navigation bar to locate the component they require or retrieve it rapidly via the component search box.
	Vant	
	Ant Design	

	Component Search	Provides the functionality to search for components, but only within the currently selected component library.
Editing Canvas Area	Vue Editor	Provides secondary editing functionality to import and parse Vue code, currently does not support parsing in languages like Pug.
	JavaScript Editor	Provide logical editing function, users can modify or import the JS code of the editing page, and view the result by attributes save function and preview function.
	Real-time Code Access	Provides real-time code modification and copying, as well as download projects in different output formats.
	Page Reset	Reset the page to its initial state.
Component Configuration Area	Binding/Free editor	Provides the functions to add, delete and modify components.
	Attributes Addition	Adding attributes to the currently selected component.
	Attributes Save	Save changes to the attributes of the currently selected component.
	Attributes Edition	Manual editing of the attributes of the selected component in modes 1 and 2
	Component Remove	Remove the currently selected component.

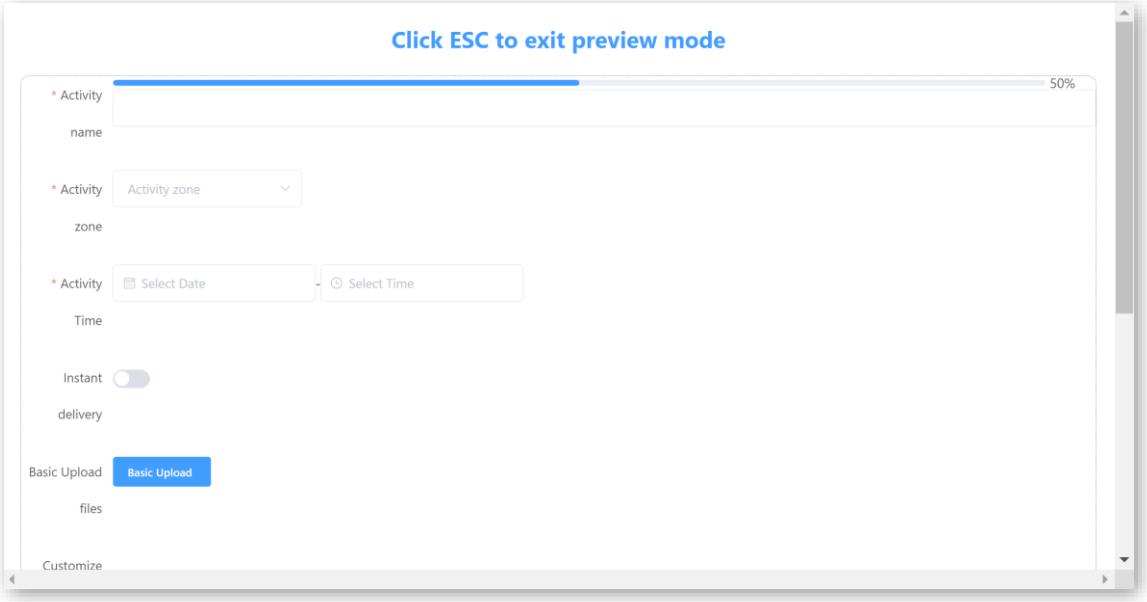
Table 3. Function Overview

4.1.3.2 Project Screenshots

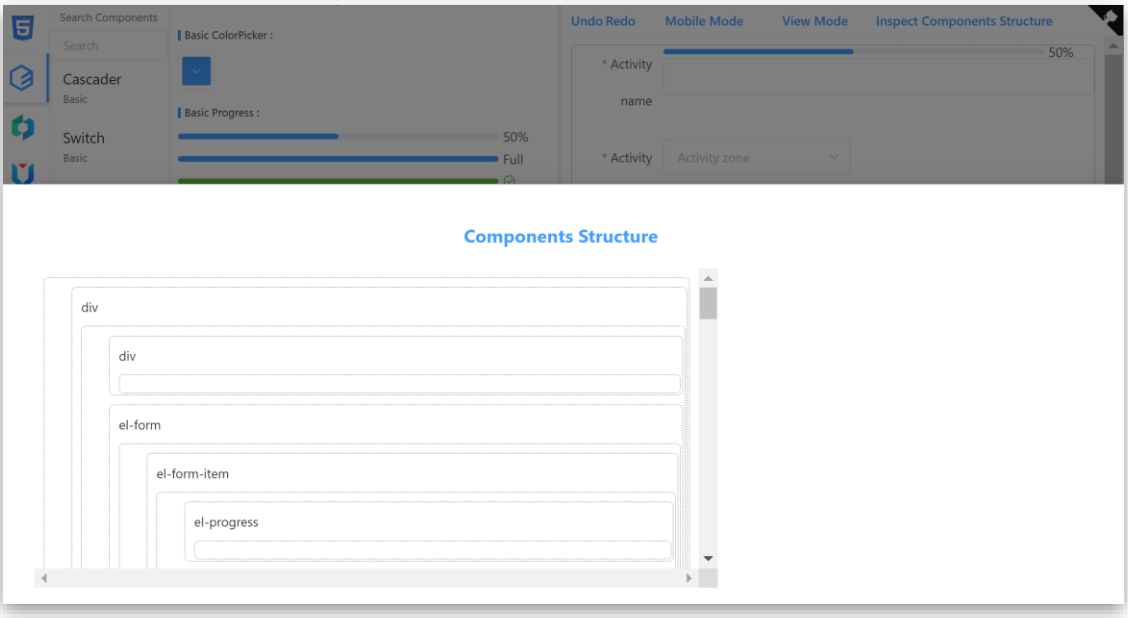
Below are the screenshots of project result:



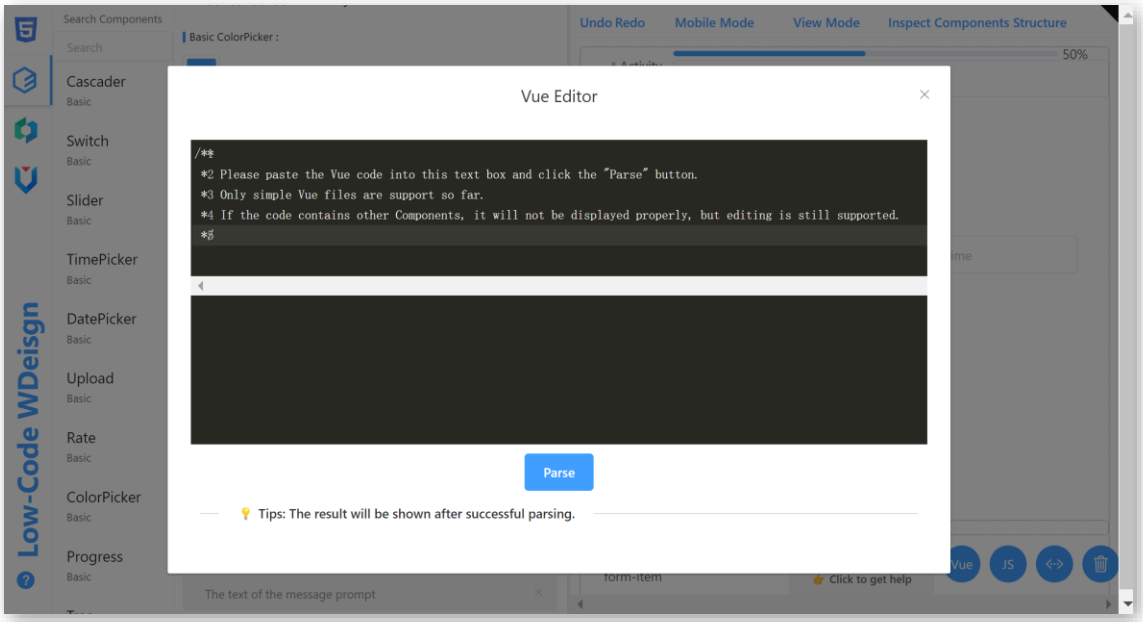
Screenshot 5. Project result



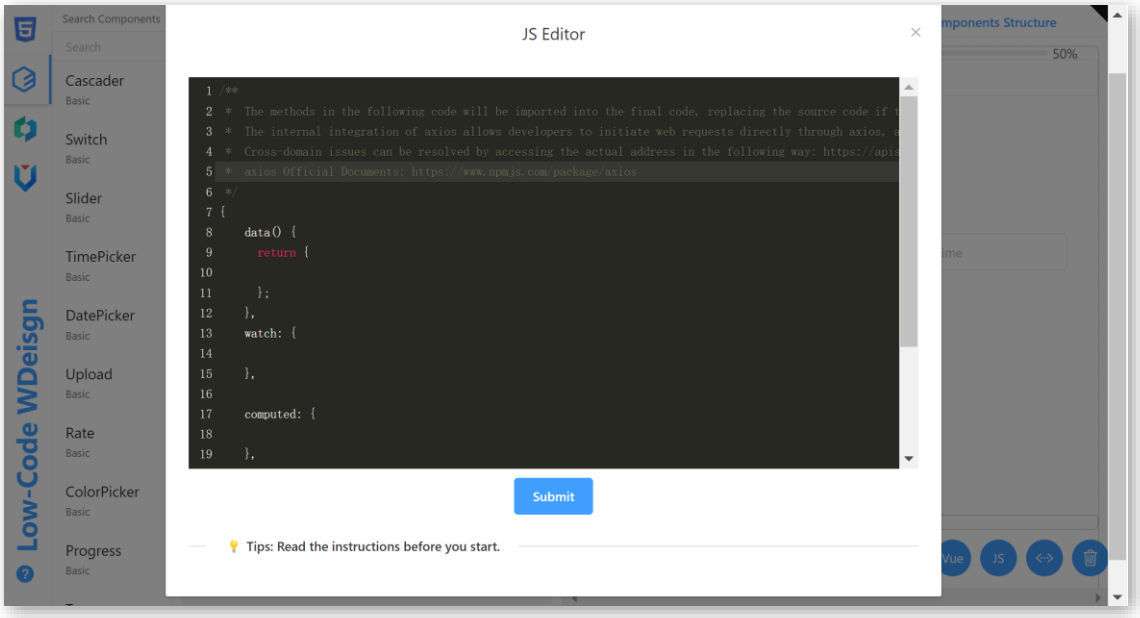
Screenshot 6. Preview Page



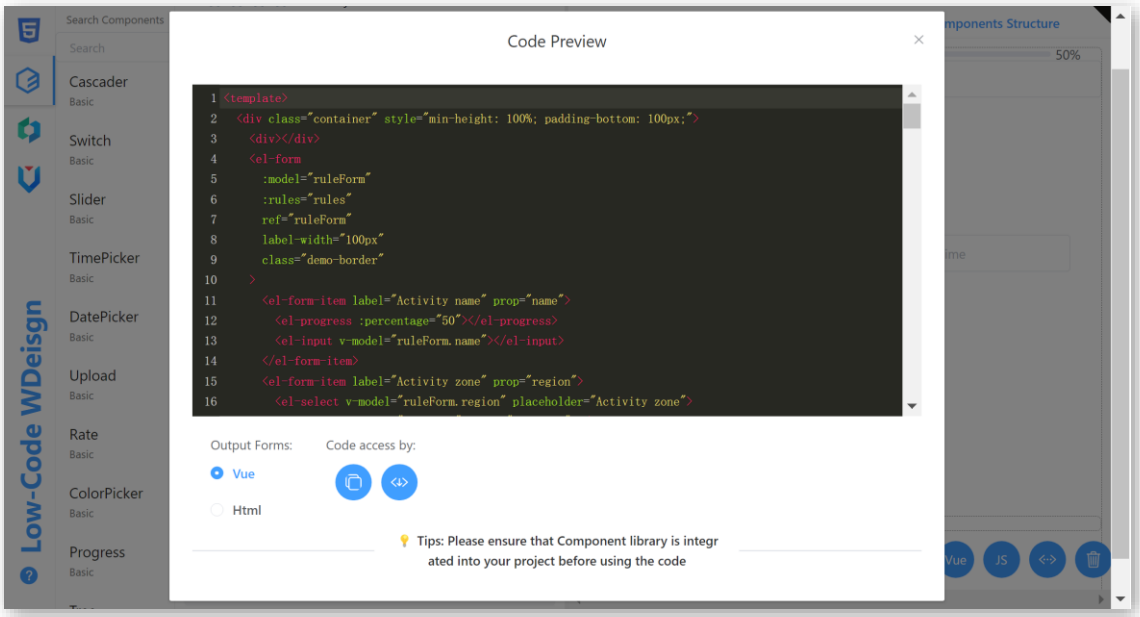
Screenshot 7. Inspect Components Structure



Screenshot 8. Vue editor



Screenshot 9. JavaScript editor



Screenshot 10. Real-time code access

4.2 Testing and Evaluation

4.2.1 Test Overview

This test plan is intended to provide guidance on the management and technical aspects of the testing process and to assist testers in the conduct of their testing activities.

The testing process ensures that the various functional, performance, and security aspects of the low-code development platform meet the requirements and reduce vulnerabilities for future maintenance and optimization during the development process.

In addition, the test results provide valuable feedback and suggestions to further improve the quality and reliability of the low-code development platform and enhance user experience and satisfaction. During the testing process, this project will utilize Lighthouse，Selenium IDE and PyCharm, in order to develop appropriate optimization plans and recommendations for improving the score and quality of the low-code development platform based on the results of performance and functional tests and analysis of test results.

4.2.2 Functional Testing

4.2.2.1 Testing Purpose

The purpose of this functional test is to verify the proper availability of the basic functions provided by the project, including component drag and drop, undo and redo, page preview, component search, page reset, code export and code copy. The tests will cover different scenarios and use cases, and the results will be validated against expected results. The purpose of testing is to ensure that the core functionality of the project meets user requirements and provides a reliable and easy-to-use low-code development platform.

4.2.2.2 Testing methods

1. Generate test cases using Selenium IDE: Use Selenium IDE to record test cases including test steps, actions and expected results.
2. Exporting test cases to Python Pytest: Export test cases generated by Selenium IDE to Python Pytest code to run tests in PyCharm.
3. Write test scripts in PyCharm: Write test scripts using Python language and introduce required libraries and modules in the code, including Selenium WebDriver and Pytest, etc.
4. Run the test script: Run the test script in PyCharm and view the test results.

4.2.2.3 Testing Case

Testing Type	Functionality testing	**Testing Case 1**	Component Drag and Drop Function

Testing Tools	● Software environment: Selenium IDE. ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019			
Testing Case Description	This test case is used to test whether the user is able to successfully drag and drop the component to the desired location.			
Pre-condition	The user has opened the editing page of the website in the browser.			
Test ID	**Test Items**	**Steps**	**Data**	**Expected results**
FT-1	Component Drag and Drop Successful	1. Open the website. 2. Select a component from the component libraries that needs to be dragged and dropped. 3. Drag and drop the component to the desired location. 4. Verify that the component is successfully displayed at the target location and that the page layout is adjusted as expected.	N/A	1. Users are able to smoothly select the desired component from the component libraries. 2. The user can successfully drag and drop the selected component to the target location. 3. The page layout can be adjusted to the user's expectations. 4. Components are displayed correctly in the target position and can be previewed and edited correctly in the website.

Table 4. Component drag and drop function testing

Testing Type	Functionality testing	**Testing Case 2**	Undo/redo Function
Testing Tools	● Software environment: Selenium IDE. ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019		
Testing Case Description	Test the functionality to undo and redo pages to verify that the functionality works properly when editing pages.		
Pre-condition	The editing canvas area contains saved page with editable components.		

Test ID	Test Items	Steps	Data	Expected results
FT-2	Undo/redo Successful	1. Open the project website. 2. Edit the page in the editor and make some changes. 3. Click the "Undo" button to undo the most recent change. 4. Verify that the page reverts to the previous state. 5. Click the “Redo” button to update the most recent change. 6. Verify that the page is updated to its latest state and that the page layout is adjusted as expected.	N/A	1. The undo operation successfully undoes the most recent modification and the page reverts to the previous state; 2. The redo operation successfully restores the most recent modification and the page reverts to the latest state.

Table 5.Undo/redo Function testing

Testing Type	Functionality testing	**Testing Case 3**	Page Preview Function
Testing Tools	● Software environment: Selenium IDE. ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019		
Testing Case Description	This test case is designed to test the preview page function of the website to ensure that users can preview and ensure the correctness of the page layout and content before exporting the page code.		
Pre-condition	The website has been opened and a page has been created.		

Test ID	Test Items	Steps	Data	Expected results

FT-3	Page Preview Displayed Successful	1. Select the Preview button on a website page to open a page preview window. 2. Check whether the layout and content of the page in the preview window are consistent with the actual page. 3. Click the links and buttons in the preview window to make sure they work properly. 4. Scroll through the preview window to make sure that all elements on the page appear correctly. 5. Close the preview window and return to the edit page of the website.	N/A	1. The preview window can be opened correctly, and the displayed page is consistent with the actual page. 2. All links and buttons work fine. 3. All elements on the page can be displayed normally.

Table 6. Page Preview Function testing

Testing Type	Functionality testing	**Testing Case 4**	Component Search Function
Testing Tools	● Software environment: Selenium IDE. ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019		
Testing Case Description	The test verifies the search functionality of the component's search box, and the correctness and completeness of the search results.		
Pre-condition	The website is opened and a certain number of components exist in the website component libraries for searching.		

Test ID	Test Items	Steps	Data	Expected results
FT-4	Component Search Successful	1. In the component management bar, locate the component search box. 2. Enter keywords in the search box. 3. Press the Enter key. 4. Verify that the search results match the keywords entered. 5. Click the component name of one of the search results to check whether the corresponding component is located in the component management bar.	1. Input: container	2. After entering a keyword in the component search box, the search results should be able to accurately match all components containing the keyword. 3. Clicking on the component name for a search result should successfully navigate to the component's detail page.

Table 7. Component Search Function testing

Testing Type	Functionality testing	Testing Case 5	Page Reset Function
Testing Tools	● Software environment: Selenium IDE. ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019		
Testing Case Description	This test will verify the correctness of the page reset function, after which all content of the page should be cleared.		
Pre-condition	A page has been created and some components have been added to it.		

Test ID	Test Items	Steps	Data	Expected results

FT-5	Page Reset Successful	1. On the edit page, click the "Reset" button. 2. To confirm the reset operation, click "OK" after a prompt box pops up. 3. Verify that all components and content on the page have been cleared except the default text.	N/A	1. All components and content on the page should be cleared and restored to their initial state, leaving only the default text.

Table 8. Page Reset Function testing

Testing Type	Functionality testing	**Testing Case 6**	Code Export and Copy Function
Testing Tools	● Software environment: Selenium IDE. ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019		
Testing Case Description	This test case is to test whether the page code export function is working properly.		
Pre-condition	The website is open and the page contains the code to be exported.		

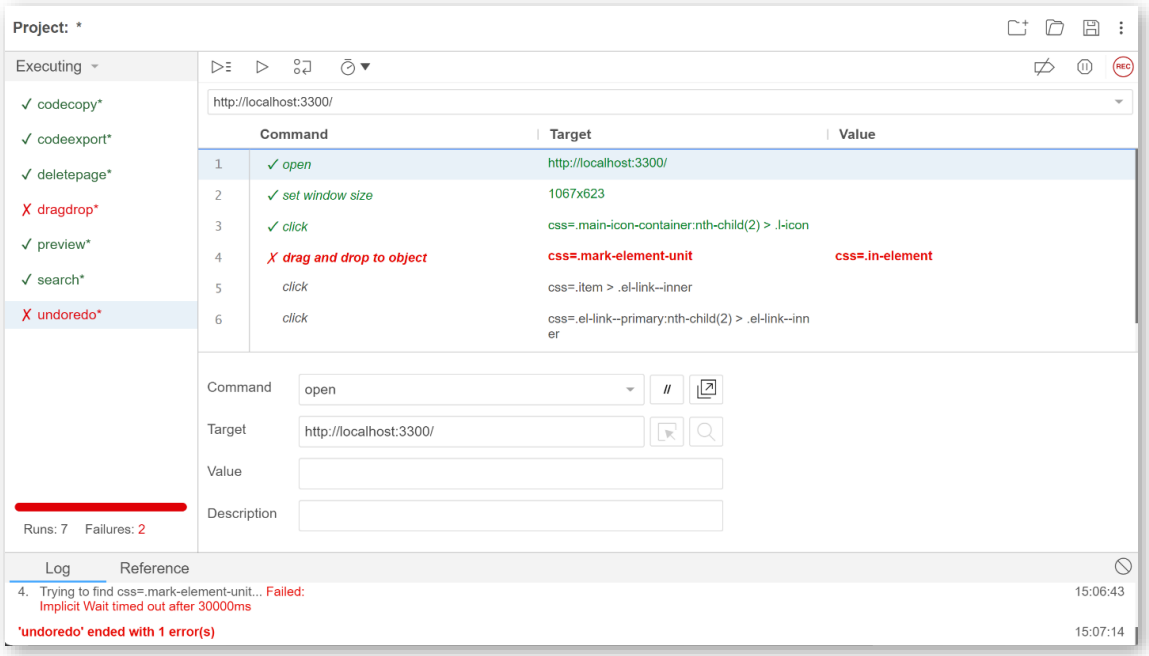
Test ID	**Test Items**	**Steps**	**Data**	**Expected results**
FT-6	Code Export Successful	1. Click the Real-time Code Preview button. 2. Set the output format, select component library, select Vue version, set filename. 3. Set file name and click set button. 4. Click the Download button.	1. Filename: test	2. Exported code files can be saved correctly. 3. Exporting does not cause page crashes or other errors.

FT-7	Code Copy Successful	1. Click the Real-time Code Preview button. 2. Set the output format, select component library, select Vue version, set filename. 3. Click the Copy button.	N/A	1. The copied codes can be pasted normally. 2. Copy operations will not cause page crashes or other errors.

Table 9. Code Export and Copy Function testing

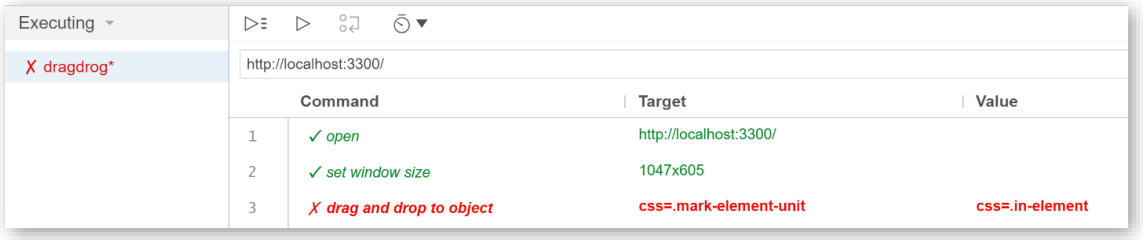
4.2.2.4 Test Actual Results

1. Generate test cases using Selenium IDE, among which the test case named "dragdrop" and the test case named "undoredo" have run errors.

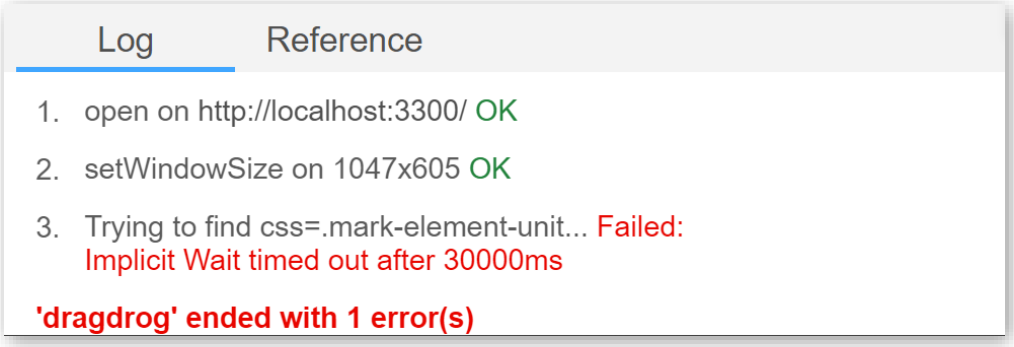


Screenshot 11. Test Actual Results

- Error message for test case "dragdrop":

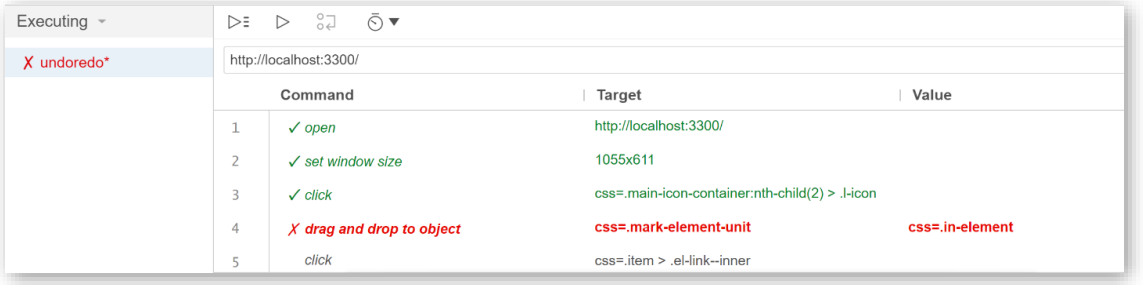


Screenshot 12. Test Actual Results

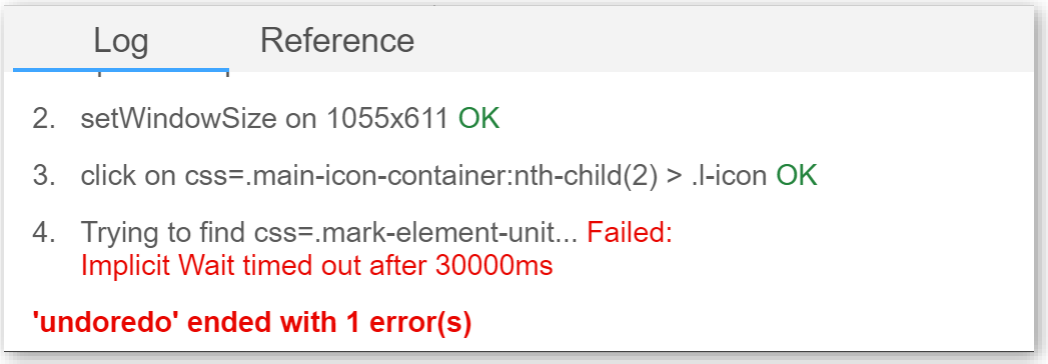


Screenshot 13. Test Actual Results

- Error message for test case "undoredo":



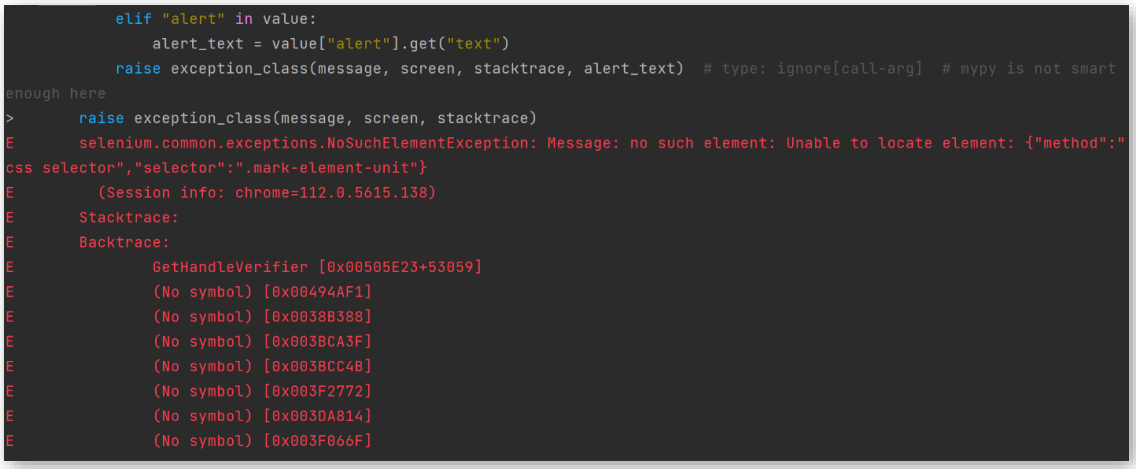
Screenshot 14. Test Actual Results



Screenshot 15. Test Actual Results

2. Export the test case as Python Pytest and run the test in PyCharm. Among them, the python files for "dragdrop", "undoredo" and "search" test cases are running with errors.

- Error message for test case "dragdrop":



Screenshot 16. Test Actual Results

- Error message for test case "undoredo":

```
                    pass
        if exception_class == UnexpectedAlertPresentException:
            alert_text = None
            if "data" in value:
                alert_text = value["data"].get("text")
            elif "alert" in value:
                alert_text = value["alert"].get("text")
            raise exception_class(message, screen, stacktrace, alert_text)  # type: ignore[call-arg]  # mypy is not smart
enough here
>       raise exception_class(message, screen, stacktrace)
E       selenium.common.exceptions.NoSuchElementException: Message: no such element: Unable to locate element: {"method":"
css selector","selector":".mark-element-unit"}

PS R:\Code\WDesign\WDesign> pytest
```

Screenshot 17. Test Actual Results

- Error message for test case "search":

```
                alert_text = value["alert"].get("text")
            raise exception_class(message, screen, stacktrace, alert_text)  # type: ignore[call-arg]  # mypy is not smart
enough here
>       raise exception_class(message, screen, stacktrace)
E       selenium.common.exceptions.NoSuchElementException: Message: no such element: Unable to locate element: {"method":"
css selector","selector":".el-input--small > .el-input__inner"}
E         (Session info: chrome=112.0.5615.138)
E       Stacktrace:
E       Backtrace:
E               GetHandleVerifier [0x00505E23+53059]
E               (No symbol) [0x00494AF1]
E               (No symbol) [0x0038B388]
E               (No symbol) [0x003BCA3F]
E               (No symbol) [0x003BCC4B]
E               (No symbol) [0x003F2772]
E               (No symbol) [0x003DA814]
E               (No symbol) [0x003F066F]
E               (No symbol) [0x003DA5C6]
```

Screenshot 18. Test Actual Results

3. Based on the error message and the error type NoSuchElementException, debug the code, import expected_conditions and By modules from Selenium library to locate the elements in the test. In addition, set the implicit wait time to 20 seconds, determine the XPath of the element in the browser, modify and utilize the full XPath to locate the elements.

```
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
```

Screenshot 19. Test Actual Results

```
18        self.driver.implicitly_wait(20)
```

Screenshot 20. Test Actual Results

```
dragged =WebDriverWait(self.driver, 10).until(
  EC.presence_of_element_located((By.XPATH, '//*[@id="app"]/div/div[1]/nav/div/div/div/div[1]/div[1]/button[1]')))
dropped = WebDriverWait(self.driver, 10).until(
  EC.presence_of_element_located((By.XPATH, '/html/body/div[1]/div/div[1]/div[2]/div[2]/div/div')))
# dragged = self.driver.find_element(By.CSS_SELECTOR, ".mark-element-unit")
# dropped = self.driver.find_element(By.CSS_SELECTOR, ".in-element")
```

Screenshot 21. Test Actual Results

4. Rerun the test scripts and the test results are all pass.

```
PS R:\Code\WDesign\WDesign> pytest
============================================= test session starts =============================================
platform win32 -- Python 3.9.12, pytest-7.1.1, pluggy-1.0.0
rootdir: R:\Code\WDesign\WDesign
plugins: anyio-3.5.0
collected 7 items

test\test_codeexport.py
DevTools listening on ws://127.0.0.1:58652/devtools/browser/7109d921-a5b4-4e18-8974-4af5f2beeffd
.                                                                                          [ 14%]
test\test_codeexportcopy.py
DevTools listening on ws://127.0.0.1:58712/devtools/browser/f4e0f3fd-a48b-47c8-bb87-7b6f4eef464c
.                                                                                          [ 28%]
test\test_deletepage.py
DevTools listening on ws://127.0.0.1:58772/devtools/browser/43e57fe1-4f8b-403f-b644-ecebbc219eaf
.                                                                                          [ 42%]
test\test_dragdrog.py
```

Screenshot 22. Test Actual Results

```
DevTools listening on ws://127.0.0.1:58772/devtools/browser/43e57fe1-4f8b-403f-b644-ecebbc219eaf
.                                                                                          [ 42%]
test\test_dragdrog.py
DevTools listening on ws://127.0.0.1:58830/devtools/browser/13d9e2d1-e4b6-4998-b351-99540c8590ad
.                                                                                          [ 57%]
test\test_preview.py
DevTools listening on ws://127.0.0.1:58890/devtools/browser/db493537-d6b9-4e5f-9a5d-6cf0317aea4c
.                                                                                          [ 71%]
test\test_search.py
DevTools listening on ws://127.0.0.1:58947/devtools/browser/bad070e0-1963-4446-89e9-5fc2a5b682b0
.                                                                                          [ 85%]
test\test_undoredo.py
DevTools listening on ws://127.0.0.1:59028/devtools/browser/735eff9b-d729-4193-bd4c-183eea3bf532
[7112:23380:0425/135641.624:ERROR:device_event_log_impl.cc(222)] [13:56:41.624] USB: usb_device_handle_win.cc:1046 Failed
to read descriptor from node connection: 连到系统上的设备没有发挥作用。 (0x1F)
.                                                                                          [100%]

============================================ 7 passed in 200.06s (0:03:20) ============================================
```

Screenshot 23. Test Actual Results

4.2.3 Performance Testing

4.2.3.1 testing Purpose

This performance test aims to test the project platform and evaluate its performance in terms of performance. The test will focus on the loading speed, response time, and resource efficiency of the platform in order to provide a reference for further optimizing the performance of the platform. Through this performance test, achieve the purpose of discovering the bottlenecks and problems of platform performance, and provide valuable suggestions and solutions to improve the platform performance.

4.2.3.2 Testing Methods

1. Prepare the testing environment: Install the Lighthouse plugin in Chrome and ensure that the test site is capable of using the Lighthouse software.
2. Set up test parameters: Set up the URL of the site to be tested, the type of test (e.g., performance, accessibility, best practices, SEO and progressive web app), and other parameters (e.g., mode and device) in the Lighthouse software.
3. Run the test: Click the "Analyse page load" button and wait for the test to be completed.
4. Analyse the test results: Carefully read the test results, which include scores, metrics and suggested improvements for each test type, and understand the meaning of the scores and metrics for each test type. For test types with low scores or poor metric performance, review the suggested improvements given by Lighthouse and try to implement them.
5. Repeat and compare: After applying improvements, run the tests again and compare the results. If test scores and metrics improve, the improvements are effective.
6. Document the test process and results: Document the parameters used during the test, the test results, and the effect of the applied improvements.
7. Summarize and reflect: Summarize the test process and results to check whether the test purpose and expected results are met. If the test results meet the expectations, the test can be considered successful, otherwise it needs to be re-analysed and improvement measures implemented.

4.2.3.3 Testing Case

Testing Type	Performance testing	**Test ID**	PT-1
Testing Tools	● Software environment: Lighthouse ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019		
Testing Selections	**Mode**	Navigation (Default)	
	Device	Desktop	
Performance (Total Score)	25		
Metrics	**Description**		**Metrics**
FCP	Measures the time it takes for any content to be rendered for the first time after the page is loaded, indicating when the user first sees the page content		40.0 s
LCP	Measures when the largest content element in the viewport is rendered to the screen		96.6 s
TBT	Measures the time that the main thread is blocked by long tasks during page load, denoting the quality of page responsiveness.		2050 ms
CLS	Measures the stability of the position of visible elements on the page, indicating whether elements on the page move or jitter during loading.		0.09
SI	Measures the time from page load to full page rendering, indicating how quickly the page loads as perceived by the user.		53.7 s

Table 10. Performance Testing 1

Testing Type	Performance testing	**Test ID**	PT-2

Testing Tools	● Software environment: Lighthouse ● Hardware tools: Lenovo Xiaoxin Pro-13IML 2019		
Testing Selections	**Mode**	Navigation (Default)	
	Device	Desktop	
Performance (Total Score)	54		
Metrics	**Description**		**Metrics**
FCP	Measures the time it takes for any content to be rendered for the first time after the page is loaded, indicating when the user first sees the page content		6.3 s
LCP	Measures when the largest content element in the viewport is rendered to the screen		15.9 s
TBT	Measures the time that the main thread is blocked by long tasks during page load, denoting the quality of page responsiveness.		30 ms
CLS	Measures the stability of the position of visible elements on the page, indicating whether elements on the page move or jitter during loading.		0.065
SI	Measures the time from page load to full page rendering, indicating how quickly the page loads as perceived by the user.		7.0 s

Table 11. Performance Testing 2

4.2.3.4 Testing Optimization

The optimization result PT-2 is shown in the Table 11 above, and the optimization method is as follows:

Ensure text remains visible during webfont load: To avoid displaying invisible text, use the font-display API and add "font-display: swap" to the "@font-face" style, then a font will be displayed as a system font before the custom font is ready. This helps ensure that

the text is always visible and legible, regardless of how long it takes the webfont to load. It also helps prevent a flash of invisible text, which can be disruptive to users.

```
1483  @font-face {
1484      font-weight: 400;
1485      font-family: vant-icon;
1486      font-style: normal;
1487      /* font-display: auto; */
1488      font-display: swap;
1489      src: url("data:font/woff2;charset=utf-8;base6
1490  }
```

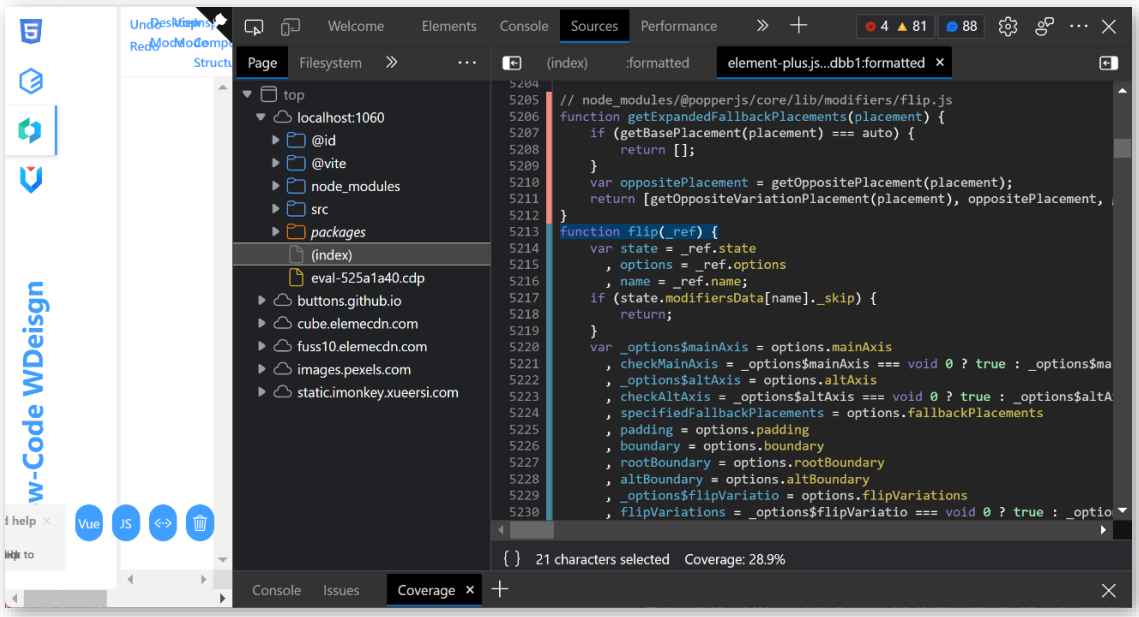
Screenshot 24. Add "Font-Display: Swap" To The "@Font-Face" Style

Pre-loading and pre-rendering: Improve page loading speed using preloading and pre-rendering techniques that allow browsers to load the resources needed for the page in advance. Get the font file in advance with <link rel="preload" as="font">, which will enable the font to render more quickly when the page is displayed.

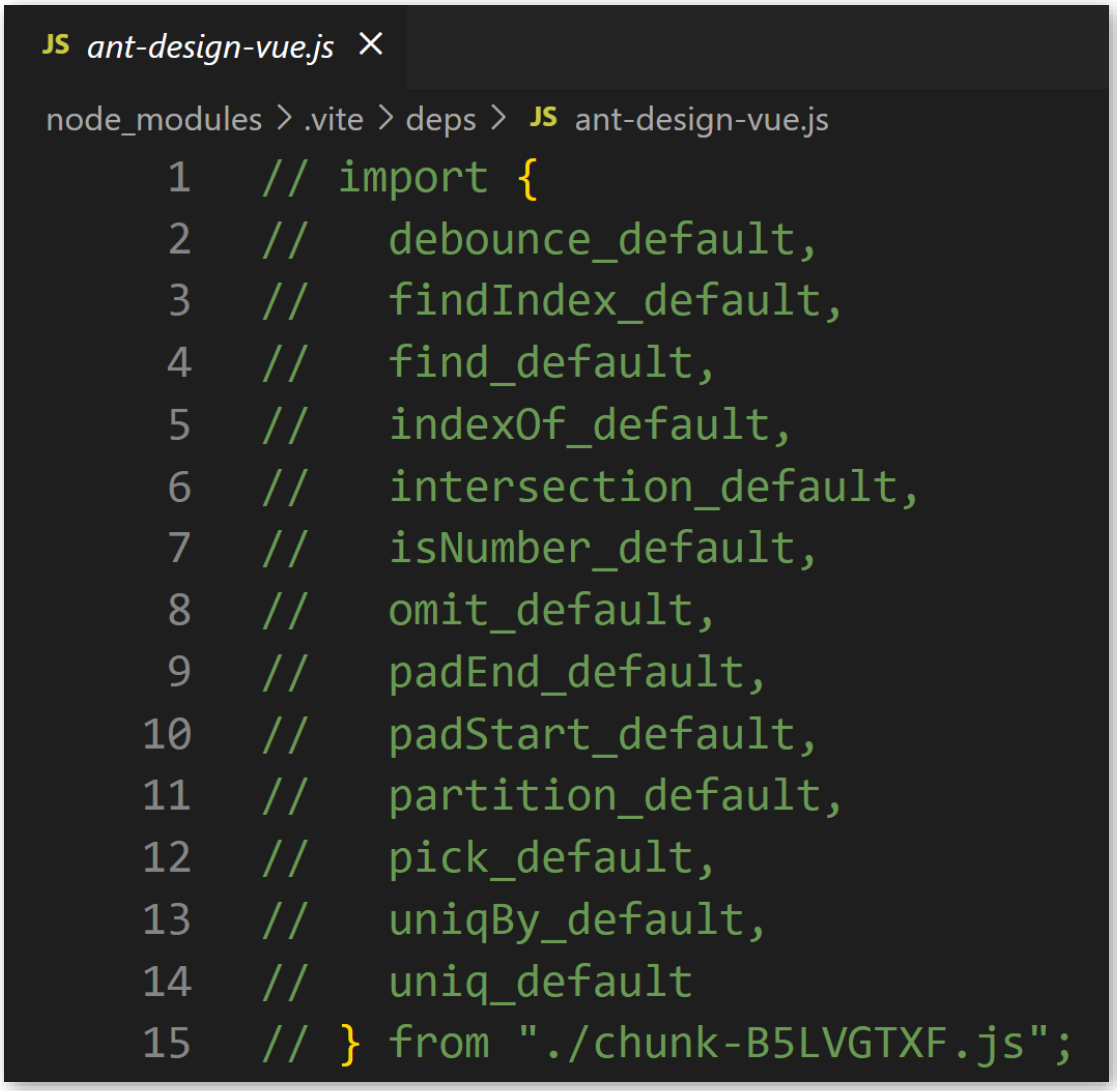
```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <link rel="icon" href="/favicon3.ico" />
6      <link rel="preload" href="/assets/Pacifico-Bold.woff2" as="font" type="f
7      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Screenshot 25. Add <link rel="preload" as="font">

Optimize server response time: reduce the file size used for the first screen and load non-first screen business logic code asynchronously. Furthermore, reduce the size of dependency resources, remove unused and duplicate dependencies, reduce unnecessary redirects as well as avoid blocking rendering JavaScript and CSS code.



Screenshot 26. Check Non-Critical Parts by Using Coverage



Screenshot 27. Remove Useless libraries

Chapter 5 Professional Issues

5.1 Project Management

5.1.1 Activities

The following are the activities required for each objective:

Phase	Objective	Time Period	Task
Requirements Analysis and Planning Phase	Project Requirements Analysis and Planning	2022/10/24-2022/11/13	1. Define project objectives and scope. 2. Gather, analyse and validate requirements. 3. Develop project plan and schedule. 4. Design project architecture and interface prototype. 5. Determine project development tools and techniques. 6. Identify features and requirements for the first round of incremental development.
Incremental Development Phase 1	Project Structure Building	2022/11/14-2022/11/27	1. Implement and deploy navigation bar. 2. Implement and deploy component library area. 3. Implement and deploy editing canvas area. 4. Implement and deploy component configuration area.
	Project Structure Debugging and Optimizing	2022/11/28-2022/12/11	1. Perform unit and integration testing. 2. Debug and optimize project. 3. Identify features and requirements for the next phase of incremental.
Incremental Development Phase 2	Component Library Area Module Refinement	2022/12/12-2022/01/19	1. Identify and integrate third-party component libraries and plugins.

			2. Implement and deploy component search function. 3. Perform unit and integration testing. 4. Debug and optimize code. 5. Identify features and requirements for the next phase of incremental development.
Incremental Development Phase 3	Navigation Bar Module Refinement	2023/01/20-2023/02/26	1. Implement and deploy undo/redo function 2. Implement and deploy mobile/desktop mode. 3. Implement and deploy view/edit mode. 4. Implement and deploy inspect component structure. 5. Perform unit and integration testing. 6. Debug and optimize code. 7. Identify features and requirements for the next phase of incremental development.
Incremental Development Phase 4	Editing Canvas Area Module Refinement	2023/02/27-2023/03/26	1. Implement and deploy Vue editor function. 2. Implement and deploy JavaScript editor function. 3. Implement and deploy real-time code access function. 4. Implement and deploy page reset function. 5. Perform unit and integration testing. 6. Debug and optimize code.

			7. Identify features and requirements for the next phase of incremental development.
Incremental Development Phase 5	Component Configuration Area Module Refinement	2023/03/27-2023/04/09	1. Implement and deploy binding/free edit function. 2. Implement and deploy attribute addition function. 3. Implement and deploy attribute save function. 4. Implement and deploy component remove function.
	Component Configuration Area Module Debugging and Optimizing	2023/04/10-2023/04/23	1. Perform unit and integration testing. 2. Debug and optimize code. 3. Identify features and requirements for the next phase of incremental development.
Incremental Optimization and Testing Phase	Project Optimization and Testing	2023/04/24-2023/05/05	1. Optimize and improve the implemented functions. 2. Perform system performance optimization and functionality testing. 3. Complete project development and deliver projects. 4. Summarize the project development process and results. 5. Analyse the problems encountered in the project development and their solutions. 6. Prospect the future development direction and application prospect of the project.

Table 12. Activities of project

5.1.2 Schedule

The following is a schedule of the project development process, which shows the entire project phase.

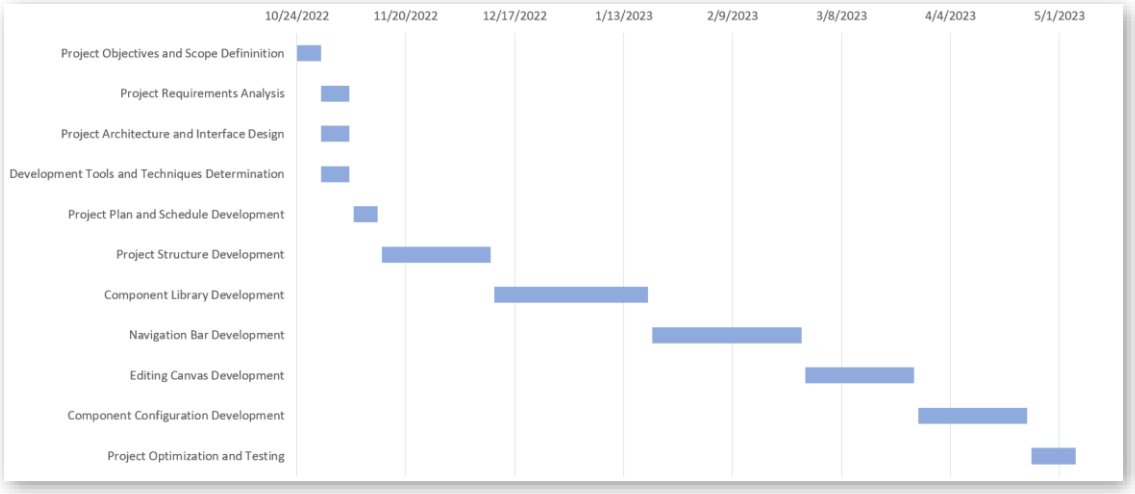


Figure 3.Gantt Diagram

5.1.3 Project Data Management

The project's reports, weekly report sheets are stored in the GitHub repository, allowing students and supervisors to check the project content and provide feedback.

Back up and upload all relevant literature to Mendeley for management and citation of the literature.

Weekly Report Sheets Link: (201918010106 Aurora) Weekly Report Sheets

Project Report Link: (201918010106 Aurora) Project Report

5.1.4 Project Deliverables

a) Project Proposal
b) Ethics Form E1
c) Progress Report
d) Final Report
e) Weekly Report Sheet
f) Poster Presentation
g) Project Code

5.2 Risk Analysis

Risk analysis as informed by the current project progress; Resolved risks and the success of the mitigation strategy; Changes to the project plan as a result of risks; Future risks.

Risk ID	Potential Risk	Cause ID	Potential Causes	Severity	Likelihood	Risk	Mitigation ID	Mitigation
R1.1	Missed deadline	C1.1.1	Illness	1	3	3	M1.1.1	Register exceptional circumstances if ill.
		C1.1.2	Cannot choose topic	1	1	1	M1.1.2	Conduct research early and meet supervisor.
		C1.1.3	Poor time management	4	2	12	M1.1.3	Make a Gantt plan early.
R1.2	Feature creep	C.1.2.1	Unsuitable or unachievable project plans	3	2	6	M1.2.1	Discuss plan with supervisor early. Create basic goals and enhancements.
		C.1.2.2	Unsuitable User Interface Design	3	2	6	M1.2.2	Discuss the UI design with supervisor, and be aware of the functionality and usability of the design.
R1.3	Software bugs	C1.3.1	Non-modular design	1	3	3	M1.3.1	Create highly modular design before implementation
		C1.3.2	Poor test plan	4	3	12	M1.3.2	Create test plan at start
R1.4	Loss of data	C1.4.1	Poor version control	4	4	16	M1.4.1	Implement version control strategy at start.

R1.6	Technical vulnerabilities	C1.6.1	Applying new technologies	4	2	8	M1.6.1	Testing of various aspects of the application of the technology.
R1.7	Data Breach	C1.7.1	Hacking of the website	5	1	5	M1.7.1	Set up firewalls and keep access logs.

Table 13. Risk Analysis

5.3 Professional Issues

5.3.1 Legal Issues

The most potential legal issue for the project is related to the processing of personal data of users. All collection and processing of users' data on this website require the prior consent of the data subjects. The GDPR [9] stipulates that the data subjects have the right to be informed about the processing of their data, whereby the utilisation of personal data by the website must be based on the data subject's consent. In this regard, the website of the project will only collect and utilise the personal data of the users with the consent of the user, as well as providing the users with the right to withdraw consent.

5.3.2 Social Issues

The availability of low-code websites and visual components increases the efficiency of website development, but also increases the potential risk to website security, since there is the possibility that developers fail to correctly identify and deal with security issues. Based on this, it could lead to a website hacking which could result in the disclosure of user data, thus causing a further risk to cybersecurity and data privacy.

5.3.3 Ethical Issues

A notable ethical implication is that the utilisation of low-code web development platforms may erode the skill level of developers, thereby squeezing traditional advanced programming skills or preventing traditional developers from earning same incomes as before, resulting in a conflict of interest between those with low development skills and those with high development skills, which is contrary to the principles of IEEE Code of Ethics [10].

5.3.4 Environmental Issues

Although low-code web development platforms can reduce the time taken to develop a website, if the volume of users on the website increases, the server may not be able to support the demand, in which case additional hardware will have to be added to expand the service. This may lead to a waste of resources and indirectly affect the environment.

Chapter 6 Conclusion

The development of an LCDP using visual components has successfully met the needs of citizen developers while reducing the high cost and low efficiency associated with traditional application development. With the adoption of modern technical architecture like Vue3 and Vite, the platform provides an array of useful features, such as drag-and-drop components, search components, preview pages, edit code, and export code. After rigorous testing, the platform has proven to be stable and provides an exceptional user experience.

During the research and development process, it became apparent that by utilizing the LCD approach, development costs could be significantly reduced compared to traditional application development. By using the LCDP, citizen developers are able to develop high-quality products in less time and low code. However, there are still some challenges and difficulties associated with the use of low-code platforms, including a steep learning curve and limited customizability and extensibility, which can require additional work to meet complex requirements.

To address these issues, this thesis offers some references to explore solutions further. Despite these challenges, it is believed that the design and implementation of this LCDP can provide an effective application development solution for citizen developers and offer a new direction and idea for related research. In the long run, the potential benefits of using LCDPs for citizen developers are significant, making it a promising area of study for the future.

References

[1] M. Fryling, ‘Low Code App Development’, J. Comput. Sci. Coll., vol. 34, no. 6, p. 119, Apr. 2019.

[2] T. Breaux and J. Moritz, ‘The 2021 software developer shortage is coming’, Commun ACM, vol. 64, no. 7, pp. 39–41, Jul. 2021, doi: 10.1145/3440753.

[3] M. Oltrogge et al., ‘The Rise of the Citizen Developer: Assessing the Security Impact of Online App Generators’, in Proceedings - IEEE Symposium on Security and Privacy, Jul. 2018, vol. 2018-May, pp. 634–647. doi: 10.1109/SP.2018.00005.

[4] V. Phalake, S. Joshi, K. Rade, and V. Phalke, ‘Modernized Application Development Using Optimized Low Code Platform’, in 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Aug. 2022, pp. 1–4. doi: 10.1109/ASIANCON55314.2022.9908726.

[5] A. C. Bock and U. Frank, ‘In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms’, in Companion Proceedings - 24th International Conference on Model-Driven Engineering Languages and Systems, MODELS-C 2021, 2021, pp. 57–66. doi: 10.1109/MODELS-C53483.2021.00016.

[6] S. Rafi, M. A. Akbar, M. Sánchez-Gordón, and R. Colomo-Palacios, ‘DevOps Practitioners’ Perceptions of the Low-code Trend’, Sep. 2022, pp. 301–306. doi: 10.1145/3544902.3546635.

[7] P. Vincent et al., ‘Magic Quadrant for Enterprise Low-Code Application Platforms’, 2020. [Online]. Available: https://www.gartner.com/doc/reprints?id=1-24BBDEDZ&ct=201005&st=sb

[8] J. Bratincevic and R. Koplowitz, ‘The Forrester WaveTM: Low-Code Development Platforms for Professional Developers, Q2 2021 The 14 Providers That Matter Most And How They Stack Up Why Read This Report’, 2021.

[9] The European Parliament And The Council Of The European Union, “on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance),” 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[10] R. Waszkowski, “Low-code platform for automating business processes in manufacturing,” IFAC-PapersOnLine, vol. 52, no. 10, pp. 376–381, 2019, doi: https://doi.org/10.1016/j.ifacol.2019.10.060.

[11] Sanchis, García-Perales, Fraile, and Poler, “Low-Code as Enabler of Digital Transformation in Manufacturing Industry,” Applied Sciences, vol. 10, no. 1, p. 12, Dec. 2019, doi: https://doi.org/10.3390/app10010012.

[12] “Mendix - the Definitive Guide to Low-Code Development,” Mendix, May 15, 2012. https://www.mendix.com/low-code-guide/

[13] Y. Luo, P. Liang, C. Wang, M. Shahin, and J. Zhan, “Characteristics and Challenges of Low-Code Development,” Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Oct. 2021, doi: https://doi.org/10.1145/3475716.3475782.

[14] L. Mearian, “Low-code tools can fill a void caused by the Great Resignation,” Computerworld, Apr. 28, 2022. https://www.computerworld.com/article/3658908/how-low-code-tools-are-filling-a-void-caused-by-the-great-resignation.html

[15] K. Shuler, “(50+) Hyper Relevant Low-Code Statistics And Facts for 2022,” quandarycg.com, Jan. 25, 2023. https://quandarycg.com/low-code-statistics/

[16] “Mendix Low-Code Platform Features - Low-Code App Development Tools,” Mendix. https://www.mendix.com/platform/

[17] A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, “Supporting the understanding and comparison of low-code development platforms,” 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Aug. 2020, doi: https://doi.org/10.1109/seaa51224.2020.00036.

[18] F. Gurcan and G. Taentzer, “Using Microsoft PowerApps, Mendix and OutSystems in Two Development Scenarios: An Experience Report,” 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Oct. 2021, doi: https://doi.org/10.1109/models-c53483.2021.00017.

[19] R. Martins, F. Caldeira, F. Sá, M. Abbasi, and P. Martins, “An overview on how to develop a low-code application using OutSystems,” IEEE Xplore, Oct. 01, 2020. https://ieeexplore.ieee.org/abstract/document/9277404/

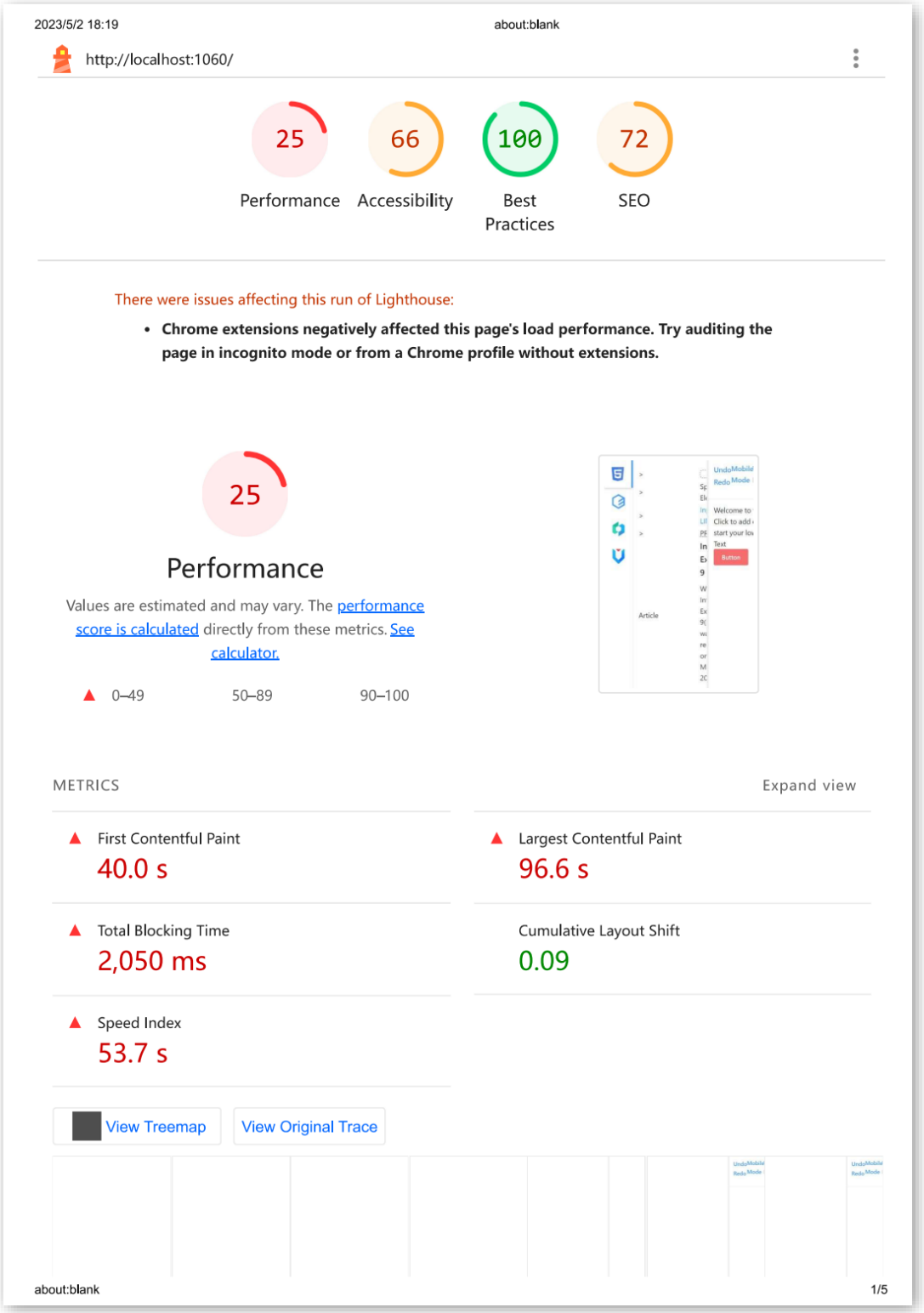
[20] P. Ramalho, H. Lourenço, and João Costa Seco, “From Builders to Editors: Bidirectional Transformations of Low-code Models,” 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Oct. 2021, doi: https://doi.org/10.1109/models-c53483.2021.00022.

[21] “Business Apps | Microsoft Power Apps,” powerapps.microsoft.com. https://powerapps.microsoft.com/ (accessed Apr. 22, 2023).

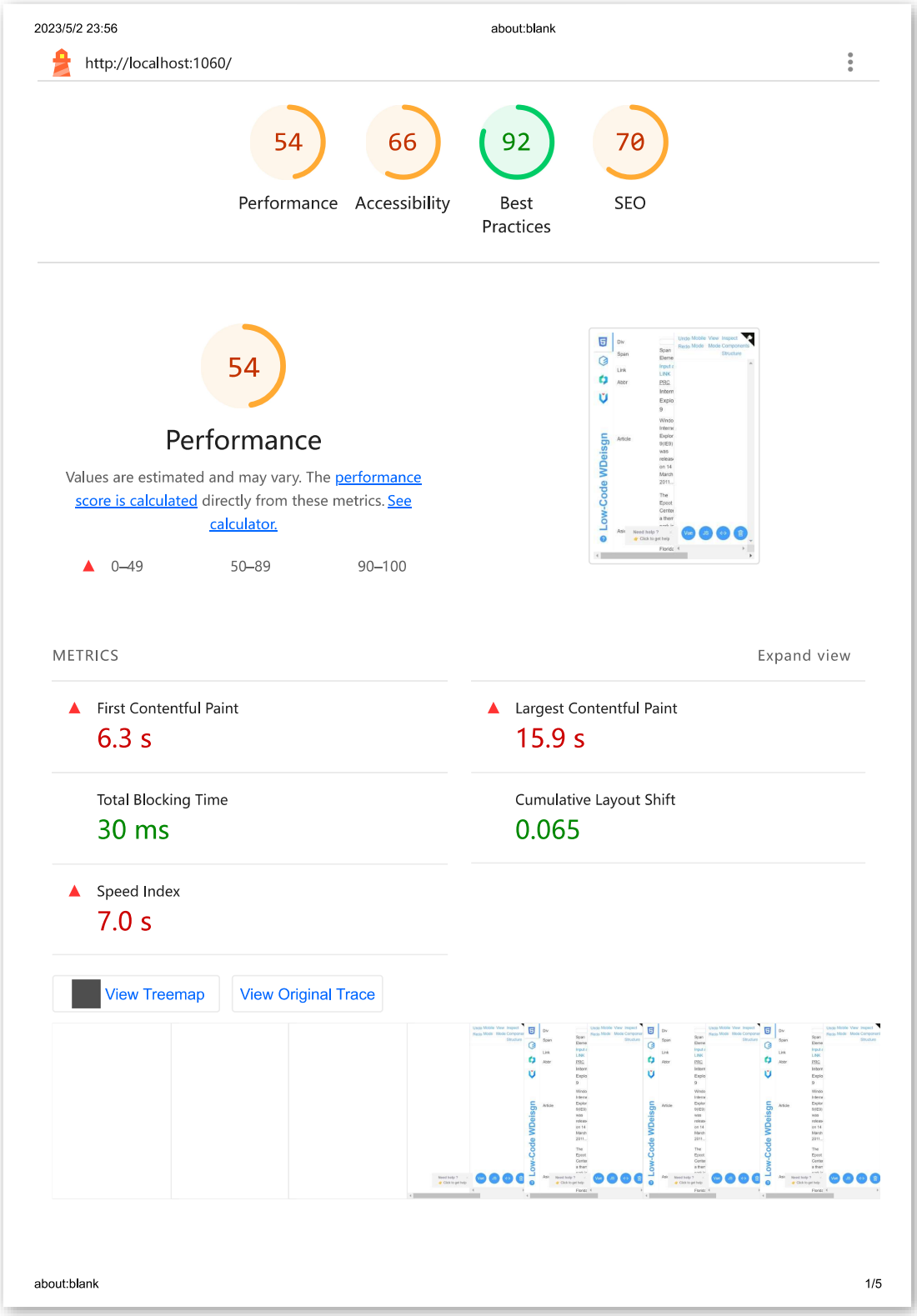
[22] “The Modern Application Development Platform | OutSystems,” outsystems.com. Accessed: Nov. 28, 2021. [Online]. Available: https://www.outsystems.com/platform/

[23] “Incremental Model in Software Engineering,” InterviewBit, Feb. 24, 2022. https://www.interviewbit.com/blog/incremental-model/

Appendices



Screenshot 28. Testing Result of PT-1



Screenshot 29. Testing Result of PT-2