Browser-Based 3D-Gallery
To Showcase Student Work

Xiaoya Zhu

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2024

Abstract

Current research on designing intuitive and accessible 3D virtual gallery platforms for displaying users' works remains limited. This project focuses on developing a browser-based 3D gallery platform that aims to enhance user interaction time and overall experience through the integration of advanced interface design and user interaction principles. One of the core areas of focus for the platform is accessibility, addressing the gap between theoretical guidelines and practical implementation while overcoming most challenges related to resource allocation and understanding accessibility standards. The platform utilizes WebGL and other cutting-edge web technologies, employing Three.js and Blender to model realistic 3D scenes, with the goal of providing a user-friendly, accessible, and engaging art display platform. Constructed using a single-page application framework, the system includes a data layer for managing 3D models, artworks, and user information; a technical support layer for rendering and performance optimization; and an application layer designed to enhance accessibility and user experience. The 3D gallery not only showcases student works in a dynamic three-dimensional environment but also features interactive functions such as artwork upload, management, and detailed descriptions of others' artworks. User testing results indicate that the project achieved a System Usability Scale score of 72.4 and a Net Promoter Score of 33.33, reflecting positive feedback while also highlighting areas for further improvement.

Ethics approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 372749
Date when approval was obtained: 2024-06-07
The participants' information sheet and a consent form are included in the appendix.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Xiaoya Zhu*)

Acknowledgements

This report represents the culmination of a long and challenging academic journey. I would like to extend my heartfelt gratitude to everyone who has supported and assisted me throughout this endeavor.

First and foremost, I am deeply and sincerely grateful to my advisor, Dr Brian Mitchell. His invaluable guidance and mentorship have been crucial throughout my research. It has been a great honor to conduct my studies under his expert supervision. I greatly appreciate his patience, empathy, and sense of humor. I consider myself fortunate to benefit from Dr Mitchell's profound knowledge and exceptional qualities.

I also wish to express my thanks to my institution for providing high-quality learning resources and a conducive academic environment. The support and assistance from the school have laid a solid foundation for my research, enabling me to engage in academic exploration effectively.

My deepest gratitude goes to my parents, who have been my unwavering support and refuge. From childhood to this very moment, they have consistently respected and supported every decision I have made. Even in times of confusion or anxiety, they have embraced my emotions fully, providing me with the courage and motivation to confront new challenges. Their steadfast support has helped me navigate through difficult times and envision a bright future beyond the fog of uncertainty.

Finally, I want to express my appreciation to my friends. I remember the confusion and helplessness I felt upon my arrival in the UK, and it was their companionship and support that gave me the strength to face all the challenges. Whether new friends or old, I hope we can all live lives filled with passion and aspiration, finding joy in the simple moments. Even when life is difficult and does not go as planned, let us believe that all the luck and blessings we deserve will eventually come our way.

I sincerely wish continued success and happiness to all those who have helped me along this journey.

Lastly, I want to acknowledge those who recognize my imperfections yet continue to show me love.

Tables of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Project motivations

A virtual gallery is an idealized exhibition space accessible to visitors unrestricted by physical walls, location, or space constraints, while encompassing all the content and information typically required for visiting a physical gallery but presented in a virtual format. Prominent museums such as the Louvre, the British Museum, the Museum of Modern Art in New York, and the Städel Museum in Frankfurt, have been offering online virtual museum services for years (Marty, 2008). These institutions provide "virtual" tour options that allow visitors to explore collections of images and simulated exhibition spaces, gaining in-depth knowledge of the exhibits. Through these virtual museum services, visitors, educators, and tourists can interact and engage more intimately and deeply with rich digital cultural content via mouse and touch panels.

Beer (1987) reports that in-person museum visitors typically spend less than a minute on each exhibit; that only 36 % of visitors view exhibits for more than 30 seconds; and that nearly half of the artwork is completely skipped. This may result in missed opportunities for deeper understanding, appreciation, and engagement with the artwork. Beer also found that most visitors showed higher engagement with interactive multimedia presentations (such as videos and interactive graphics), indicating that interactivity is a key factor in capturing visitor interest. The project's initial aim was to address this issue by extending the time visitors interact with each exhibit. However, the immediate priority was to develop a well-designed working system to encourage greater participation, focusing on the integrated application of interface design and user interaction to enhance the overall user experience.

Accessible design is another crucial project goal, Sections 2.3 and 3.2. Sadly and needlessly, too many websites still fail to meet accessibility standards. Reasons for this include insufficient allocation of resources, insufficient understanding of accessibility design guidelines, and perhaps insufficient general awareness.

1.2 Project objectives

The goal of this project is to develop a browser-based 3D gallery:

1. Optimizing the User Experience:

1. Ensure the integration of gallery space with exhibition design, avoiding excessive text pages; using 3D technology to recreate a physical gallery; enhancing the experience for virtual visitors.

2. Focusing on Accessibility:

1. The interface should be user-friendly for a wide range of users and consistent across all sections, ensuring that the system responds promptly to user inputs and operations.
2. Enable visitors of different ages, backgrounds, and levels of expertise to participate in virtual tours from any location. The virtual gallery offers broad accessibility but some users may face struggle due to a lack of appropriate technology or digital literacy.
3. The system should support multiple access points, including personal computers (PCs), tablets, and smartphones, while ensuring compatibility with both internet and intranet networks.

3. Meeting Business and Functional Requirements:

1. The system should correctly process data and execute tasks accurately to ensure normal operation.
2. It should respond quickly to user requests with minimal memory requirements, ensuring an efficient user experience.

4. Ensuring Security:

1. The system should protect user data from unauthorized access and breaches, featuring robust user permission management.

5. Maintainability and Scalability:

1. The system should be easy to modify and upgrade to meet changing requirements and be scalable to accommodate future needs.

1.3 Difficulties of the problem

Technical Complexity: Integrating multiple modern technologies involves detailed planning and execution in areas such as state management, routing configuration, and resource allocation to maintain smooth and efficient system operation. Additionally, 3D rendering may involve complex computations and resource optimization, which is critical for keeping the system stable under high-load conditions.

User Experience Design: Creating an interface that is both visually appealing and responsive across various devices requires balancing aesthetics with functionality. The design should also consider accessibility to accommodate a diverse user base, including those with different abilities.

Scalability: The system needs to handle future increases in user numbers and data processing demands. This involves flexibility in the underlying architecture and consideration of potential technology upgrades and expansion plans, allowing the system to adapt to evolving user needs and the changing technological environment.

Time Constraints: The project requires a balance between system features, performance optimization, and completion within a limited timeframe to meet the delivery schedule. Precise management and allocation of time and resources are essential to achieve the desired goals within the set deadline.

1.4 Contributions

This project developed a browser-based 3D gallery, utilizing 3D technology to create a three-dimensional virtual exhibition hall. The program uses a typical single-page application framework containing a data layer, a technical support layer and an application layer (Section 2.1.4). The data layer handles the storage and retrieval of 3D models, artwork, and user personal information. The technology support layer provides the infrastructure needed for rendering and interaction, optimizing the performance and responsiveness of the virtual gallery. The application layer prioritizes accessible content in its design, providing user interface and experience features. The Virtual Showroom, taking a cue from the design of real-world galleries, creates realistic 3D scenes using Three.js (a frontend 3D library, Section 2.2.2) and Blender (3D modelling software, Section 2.2.3). Artwork is displayed as high-resolution textured 3D models, allowing virtual visitors to observe and interact from various angles. The frontend enables artwork and a description to be uploaded to the database backend. The use of a database simplifies and strengthens the application. The database, like all the project's core components, was chosen carefully (Section 2.4) for suitability and features. The virtual gallery's functionality include browsing artwork and descriptive information, uploading, deleting, and managing personal artwork, and displaying them in the gallery. The finalised functionality can be found in Section 3.1.

1.5 Report structure

Chapter 2 provides background knowledge necessary for building a website, including key technologies and the fundamental framework. It explores relevant knowledge in 3D modeling, emphasizing that selecting the right 3D technologies is crucial for enhancing the project's implementation and user satisfaction. This chapter then offers a detailed discussion on accessibility, which is one of the key factors for project success, aiming to meet diverse user needs as effectively as possible. Finally, it analyzes the various resources required for the project, laying the groundwork for subsequent implementation. Chapter 3 shifts to the specific implementation of the project, including a detailed description of project functionalities, interface design, and critical details in the implementation process. The interface design section pays particular attention to small techniques for accessibility and interactivity, striving to enhance the user experience through detailed optimization. Chapter 4 focuses on user study, analyzing and evaluating the system from multiple angles through Alpha and Beta testing. Chapter 5 summarizes the project's outcomes and suggests future development directions. It is important to note that evaluations and conclusions are typically integrated into the various sections of the report rather than being presented as standalone chapters, ensuring a more coherent analysis and summary throughout the report.

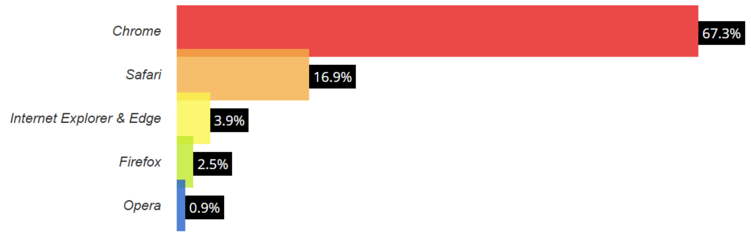


Figure 2.1 Browser market share in July 2024 (W3Counter, 2024).

Chapter 2

Background

This section provides the basic information needed to understand the project. It begins with an overview of the basic steps and considerations involved in developing a Web application, then introduces key concepts related to 3D development. This is followed by an explanation of the core concepts of accessible design. Lastly, a list of resource requirements is presented to illustrate the decision-making process.

2.1 Developing web applications

2.1.1 Web browser

As of July 2024 (W3Counter, 2024), the five most commonly used browsers are, in order: Chrome, Safari, Internet Explorer and Edge (combined), Firefox, and Opera (Figure 2.1) with Chrome leading the market with a 67.3 % share. Figure 2.2 (W3Counter, 2024) shows Chrome's usage has steadily increased over the past decade, far surpassing other browsers. Based on these statistics, the development will primarily focus on Chrome, with compatibility testing conducted on Edge, Firefox, and Opera.

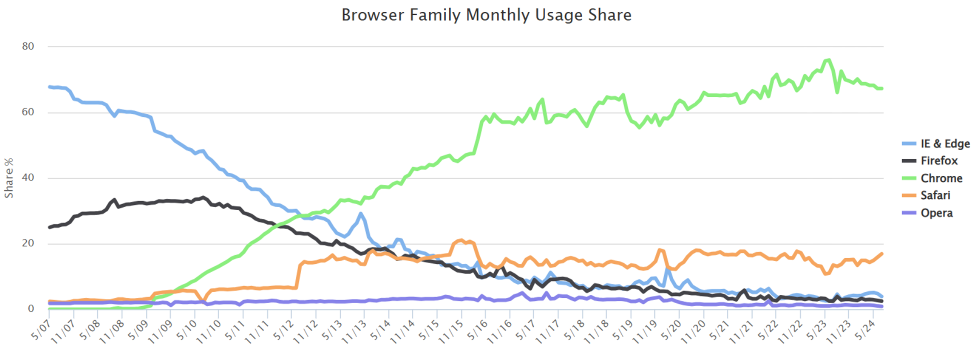


Figure 2.2 Browser market share history (W3Counter, 2024).

2.1.2 Develop a website

Web-based applications are advanced websites that offer more than just text display. They use HTML (Hypertext Markup Language) for structure, which is converted into a Document Object Model (DOM) that JavaScript manipulates to modify content dynamically. HTML tags, like the 'class' attribute, add properties to elements, while JavaScript methods like 'createElement()' create nodes.

HTML alone rarely suffices for modern websites: CSS and JavaScript have become essential. CSS (Cascading Style Sheets), introduced by the World Wide Web Consortium (W3C) in 1996, styles HTML elements, simplifying the styling process compared with previous HTML-only methods. JavaScript, included in HTML via the '<script>' tag, adds interactivity and handles complex functionalities. It supports advanced features like WebGL and HTML5 Canvas. JavaScript runs locally, executed by the browser (client-side) thus reducing network dependency.

2.1.3 Web browser compatibility

A key challenge in web application development is the inconsistent support for HTML, CSS, JavaScript, and the DOM across different browsers and versions. This variation stems from the different development goals and strategies of the major browser companies. In the late 1990s, such inconsistencies were problematic due to the lack of uniform standards. The situation improved with the W3C formation, led by Tim Berners-Lee, which developed and promoted web standards. Although standards like CSS, HTML, and JavaScript have become more uniform, subtle differences remain due to ongoing browser-specific implementations. For example, browsers may parse the CSS box model differently, leading to variations in the calculated width and height of an element. Fortunately, most common web elements are widely supported across all major browsers. The Quirksmode website offers a detailed comparison of browser support.

The implementation of WebGL (Section 2.2.2) also varies from browser to browser. For example, some advanced WebGL features, such as reflections and complex shaders, perform very smoothly in Chrome and Firefox but may cause performance degradation or graphical rendering errors in Safari. Additionally, different browsers handle hardware acceleration differently on various platforms. For instance, Chrome and Firefox with hardware acceleration enabled can render WebGL 3D graphics very smoothly, while Safari with hardware acceleration disabled or low-performance versions of Internet Explorer may experience significantly reduced rendering speed and effectiveness.

JavaScript interpreter implementations differ across browsers, posing more development difficulties. The introduction of ECMAScript increased cross-compatibility. Developers must anticipate the problems these differences can cause, test thoroughly, and potentially reimplement parts of the code. Browser sniffing, which involves adapting code based on the type and version of the user's browser, is one approach. However, due to frequent browser updates, the W3C does not recommend this method, preferring instead feature detection, which checks whether a browser supports a particular feature (such as WebGL) rather than relying on a specific browser version.

2.1.4 Web framework for single page applications

One of the main goals of this project was to create a Single Page Application (SPA), an application that loads and runs in a client browser without requiring a full page refresh (Mikowski and Powell, 2013). Table 2.1 compares Multi-Page Applications (MPA) and SPA. For a complex code base, an ordered and modular structure must be implemented and maintained. The project is expected to have tens of thousands of lines of code and contain a large number of classes, making it impractical to just embed JavaScript code directly into script tags in HTML pages. A more advanced solution was required.

Fortunately, the industry recognizes the need for advanced code structure and offers a proven solution: the Model View Controller (MVC) architecture (Krasner, Pope et al., 1988). MVC splits the application into three parts: the model (data and logic), the view (user interface), and the controller (input processing). The

Requirement	**MPA**	**SPA**
Whether to reload resources after a jump	Yes	No
User experience	The page transitions are slow and choppy, especially on mobile devices	Fast page transitions and a good user experience, including on mobile devices
Passing data between pages	Implementing dependencies on URL, cookies, or localStorage can be complex	Using Vuex for data transfer between parent and child components is quite straightforward
Scope of application	Websites that need to be search engine friendly	Websites with high user experience requirements

Table 2.1 Comparison of MPA and SPA.

controller updates the model based on user input, while the view reflects changes in the model (Leff and Rayfield, 2001).

Vue.js, developed by Evan You, simplifies creating single-page applications and complex UIs (Scott Jr, 2015). While not strictly MVC, Vue.js uses the flexible MVVM (Model-View-ViewModel) pattern, facilitating efficient UI management through components and responsive systems (Macrae, 2018). This modern pattern, also known as MVVM or MV* frameworks (Vyas, 2022), supports intuitive data binding and is used by major companies like Alibaba, Xiaomi, and GitLab.

Choosing the right tool is crucial for setting up a Vue front-end project. Traditional tools are often complex and time-consuming, while Vite provides an efficient solution. Vite uses native ES modules for on-demand compilation, which means it only compiles the files actually requested. This approach greatly reduces the startup time of the development server. Vite employs Rollup in production for effective code bundling and optimization, producing smaller builds with improved performance that enhance the application's load speed and runtime efficiency.

2.2 Graphics for Web Applications

2.2.1 Why 3D graphics

Conveying the 'cultural context' of artwork online is crucial, as simply displaying the piece doesn't fully reflect its value. 3D environments excel at simulating the artwork's surroundings, enhancing depth and reproducibility (Cockburn and McKenzie, 2001). Research shows that user interaction in virtual galleries boosts exhibition effectiveness (Higuera and Pernas, 1999). While 2D galleries provide detailed textual information, they lack spatial depth. In contrast, 3D galleries offer a dynamic, immersive experience, allowing users to explore exhibits freely and increasing engagement (Muller et al., 1999).

However, there are also challenges with 3D displays. Many users are unfamiliar with 3D environments, which can lead to navigation and manipulation difficulties (Brooks, 1999). Novice users may encounter problems in positioning, movement and interaction, which may affect their experience. Therefore, optimising navigation design in 3D environments to enhance the user experience remains an important issue in the field of digital art display.

2.2.2 WebGL

3D graphics processing is significantly more expensive than 2D images due to its complex mathematical models and high rendering requirements. Especially in the early days, the requirement to update 3D images at 60 frames per second was a huge challenge for many web browsers (Shreiner et al., 2009).

Prior to the standardisation of WebGL, displaying 3D graphics relied heavily on browser plug-ins. For example, Adobe Flash Player was one of the dominant plug-ins, but it required special implementations for different operating systems, limiting compatibility and user experience (Evans et al., 2014).

Piggybacking on the HTML5 standard has gained WebGL widespread adoption as a 3D graphics API for the web. WebGL is based on OpenGL ES 2.0, an embedded system version of the OpenGL standard for resource-constrained devices such as smartphones and tablets (Parisi, 2012). WebGL is rendered via an HTML5 Canvas element for rendering and utilises a dedicated WebGL context to process 3D graphics, which is different from traditional 2D drawing methods.

While WebGL provides a low-level API that allows developers to have fine-grained control over graphics rendering, its complexity also requires developers to implement all features manually. High-level abstraction libraries can address this issue.

Three.js provides a more intuitive API for creating, rendering, and interactively manipulating 3D scenes by encapsulating the complexity of WebGL. It has a wide range of built-in geometry (such as cubes, spheres and planes) and materials (such as basic, standard and mesh materials), and supports texture mapping and shader effects to enrich the visual presentation. three.js has been used in a wide range of applications such as commercial advertisements, interactive art exhibitions, education and game development. For example, Mozilla's A-Frame framework is a virtual reality development tool based on Three.js that makes it easier for developers to create immersive virtual reality environments (Marcos et al., 2017).

2.2.3 Blender and 3D file format

The modelling process in Three.js is relatively complex and can be simplified with the help of other modelling software, such as Blender, a powerful open source 3D modelling and animation tool. Blender supports a wide range of export formats, allowing models to be used efficiently in web projects. Although there is a wide variety of 3D file formats, including .obj, .3ds, .u3d, .o3d, .x3d, .dae, .unity, .skp, .dxf, and .blend (McHenry and Bajcsy, 2008), which have their own applications in the field of 3D computer graphics, not all of them are widely used. Some formats are more popular in specific areas, while others may be relatively unknown.

Schilling, Bolling and Nagel (2016) show that the .gltf and .ply formats are increasingly used in the field of heritage preservation and presentation. These formats are preferred for modern Web 3D scenes due to their lightweight, efficiency and strong compatibility. The .gltf format, in particular, is more compatible with Three.js, which is whyit was chosen for this project. However, despite the multitude of 3D file formats, relying on a single format can lead to difficulties in synchronising updates and increase the complexity of management and maintenance (Rahaman and Champion, 2019). Therefore, choosing the right file format is critical to the long-term maintainability and success of a project.

The combination of Blender and Three.js not only simplifies the 3D modelling process, but also enhances the visual and interactive experience of web projects.

2.3 Design for accessibility

2.3.1 What is accessibility

Accessible design aims to remove barriers that limit access to the Internet and enable all users, including people with disabilities, to use, perceive, understand, navigate and interact with the Web. As defined by the World Wide Web Consortium, accessible design enables people with disabilities to perform a variety of online tasks, such as shopping and information browsing.

The social significance of accessible design is to ensure that all people have equal access to the Web, not just those who can afford the technology or use standard equipment. Good accessibility design is not only beneficial for people with disabilities, but also for all users as it adapts to a wide range of access devices (such as mobile phones and PDAs).

Common types of accessibility barriers include visual impairments (such as blindness and low vision), hearing impairments (deafness and hearing loss), motor impairments (physical disabilities affecting the use of a mouse or keyboard), cognitive and learning disabilities (like dyslexia and memory impairments), and language barriers. For detailed descriptions of these barriers and their impact on web accessibility, please refer to the Appendix A.1, page 45.

2.3.2 Key principles of accessible design

There is a large body of literature available on the principles of Web accessibility, for example Brinck, Gergle and Wood (2001) and Krug (2000) and Rutter et al. (2007). Perhaps the most influential work related to accessible web design has been the guidelines developed by the W3C Web Accessibility Initiative (WAI), which provides a comprehensive set of guidelines and checkpoints to help ensure that web sites conform to the concept of 'designing for everyone.'

The WAI promotes the following three core guidelines in accessible design:

Web Content Accessibility Guidelines (WCAG) describes how to make web content and websites accessible to all users, including those with disabilities.

Authoring Tools Accessibility Guidelines (ATAG) focuses on the accessibility of web authoring tools for creating content.

User Agent Accessibility Guidelines (UAAG) concerns the accessibility of tools for accessing content, such as browsers and media players.

Whilst ATAG and UAAG are also important, the focus will be on WCAG for two reasons: firstly, there is a rich literature on WCAG, which facilitates in-depth analyses; and secondly, many automated assessment tools rely on the rules and standards in the WCAG guidelines.

WCAG 2.0 (released in 2008 after WCAG 1.0) is built around four core design principles that provide the foundation for Web accessibility (Caldwell et al., 2008):

Perceptibility Web content must be presented in a way that users can perceive it. For example, provide alternative text descriptions to images and provide captions for multimedia content.

Operability user interface components must be operable via multiple input methods and have the ability to correct user errors. For example, ensure that all functions can be operated from a keyboard.

Understandability Web content should be easy to understand and manipulate. For example, use clear language and consistent navigation layouts.

Compatibility Web content should display and operate properly on different devices and user agents, including assistive technologies. For example, ensure that web pages render correctly on a variety of browsers and devices.

WCAG 2.0 contains 12 guidelines, each with one or more testable Success Criteria (SCs) that are categorised as Level A (lowest), Level AA (medium) and Level AAA (highest), totalling 61 Success Criteria. A problem may involve more than one prioritised success criterion.

2.4 Resources

2.4.1 Front-end framework

🏆 **Vue** is a progressive framework ideal for small to medium-sized projects and rapid development.

- **Pros:** **Easy to learn:** clear documentation, beginner-friendly. **Flexible:** can be gradually introduced or fully adopted. **Reactive:** two-way data binding keeps the view and model in sync.
- **Cons:** **Smaller ecosystem:** weaker compared with React. **Simple state management:** less comprehensive than React and Angular.

React (developed by Facebook) uses a virtual DOM and is suitable for large single-page applications.

Pros: **High performance:** virtual DOM improves performance. **Component-based:** easier to maintain and reuse code.

Cons: **Steeper learning curve:** not beginner-friendly. **Requires additional libraries:** focuses only on the UI layer.

Angular (developed by Google) is an MVC framework suited for large enterprise-level applications.

Pros: **Comprehensive framework:** includes dependency injection and routing. **Strong typing support:** developed with TypeScript. **Well-equipped tools:** built-in Angular CLI.

Cons: **Steep learning curve:** harder to learn than React and Vue. Code redundancy: generates larger code files.

2.4.2 Back-end framework

🏆 **Express** is a lightweight Node.js framework ideal for rapid development and highly customizable projects.

Pros: **Easy to use:** Simple setup with minimal code. **Flexible:** Highly customizable and easy to integrate. **High Performance:** Non-blocking I/O and lightweight architecture. **Rich Ecosystem:** Extensive npm packages available.

Cons: **Simple Structure:** Requires manual architecture design. **Scalability:** Can become complex in large-scale applications.

Django is a Python framework suited for projects needing built-in features.

Pros: **Comprehensive Features:** Includes ORM and admin interface. **Security:** Built-in protections against common threats. **Consistency:** Adheres to consistent development conventions.

Cons: **Heavy:** Comes with many features, which may add complexity. **Inflexibility:** Limited extensibility compared with lighter frameworks.

Spring Boot is a Java framework designed for enterprise-level applications.

Pros: **Enterprise Support:** Provides a robust toolset and enterprise features. **Auto-Configuration:** Reduces boilerplate code.

Cons: Steep Learning Curve: Requires significant time to master. **Resource Consumption:** Can lead to higher memory and resource usage.

2.4.3 CSS framework

🏆 **Tailwind CSS** is a utility-first framework that enables custom designs directly in the markup.

- **Pros: Highly Customizable:** Build fully custom designs using utility classes. **Clean Code:** Reduces duplicate CSS. **Flexibility:** Supports component-based design and on-demand loading.
- **Cons: Learning Curve:** Requires familiarity with utility class usage. **Verbose Class Names:** Utility classes in HTML can be verbose.

Bootstrap (developed by Twitter) provides pre-designed components and a responsive grid system for rapid development.

- **Pros: Fast Development:** Predefined components and styles speed up development. **Consistency**: Ensures consistent appearance across browsers. **Mature Ecosystem:** Rich plugins and extensive community support.
- **Cons: Similar Designs:** Websites may look similar. **Limited Customization:** Design is constrained by the framework's preset styles.

2.4.4 Database

🏆 **PostgreSQL** is a relational database ideal for complex queries and high data integrity requirements.

- **Pros: Advanced Features:** Supports complex queries and transactions, ensuring data integrity. **Scalability:** Allows for customization and extension, enhancing functionality. **Standards Compliance:** Adheres to SQL standards, offering broad compatibility.
- **Cons:** Resource Intensive: Requires more system resources.

MongoDB is a NoSQL database suitable for flexible data structures and high scalability needs.

- **Pros: Flexible Schema:** Supports dynamic data structures. **High Scalability:** Scales horizontally to handle large volumes of data. **Easy Development:** JSON-like document structure simplifies data manipulation.
- **Cons: Limited Transactions:** Not as robust in transaction support as relational databases. **Data Integrity:** Requires additional mechanisms to ensure consistency.

2.4.5 Web server

🏆 **Nginx** is a high-performance, lightweight web server and reverse proxy, designed to handle high concurrency and static content efficiently.

Pros: **High Performance:** Event-driven architecture handles many concurrent connections efficiently. **Low Resource Consumption:** Lightweight design with minimal memory and CPU usage. **Reverse Proxy and Load Balancing:** Built-in powerful features. **Scalability:** Supports various protocols through extensible modules.

Cons: Limited Dynamic Content Handling: Fewer built-in features for dynamic content compared to Apache.

Apache supports multiple dynamic content and module extensions

Pros: **Feature-Rich:** Supports a wide range of dynamic content and extensive modules. **Flexible Configuration:** Directory-level configuration via .htaccess files.

Cons: **Performance Overhead:** Less efficient than Nginx under high concurrency. **Higher Resource Consumption:** Uses more memory and CPU.

2.4.6 Winners

Front-end framework: Vue

Learning curve's suitability for the project timescale.

Back-end framework: Express

Lightweight and flexibility; avoids the time needed to learn a new language.

CSS framework: Tailwind CSS

Customisability and flexibility which suit the personalised accessibility requirements of the project, as well as its ability to reduce redundant CSS and optimised page load speeds.

Database: PostgreSQL

Ability work with complex (non-text) data; data integrity and advanced query capabilities (despite its high resource requirements).

Web server: Nginx

Low resource consumption; high concurrency and static content processing; powerful reverse proxy functionality.

2.4.7 Assessment of technology stack appropriateness

It can be concluded after the project is completed that these choices are indeed suitable for meeting the project's goals and requirements, and they also provide flexibility for future scalability and enhancements.

Vue.js and SPA The combination of Vue.js with SPA architecture provides a highly responsive and efficient front-end experience. This choice enhances smooth navigation and quick interactions, aligning with the need for a user-friendly and interactive interface.

Express and PostgreSQL Express as the back-end framework offers the necessary flexibility and scalability, while PostgreSQL provides robust data management and supports complex queries. These capabilities are crucial for handling dynamic content and managing user data effectively.

Nginx Implementing Nginx as the web server ensures high performance and reliable load balancing. This choice improve the system's reliability even under heavy traffic conditions.

Tailwind CSS Tailwind CSS facilitates rapid UI development with a focus on responsiveness and customizability, ensuring a consistent and attractive design across different devices.

Three.js and Blender These tools allow for the integration of complex 3D graphics into the application, significantly enhancing visual appeal and interactivity, which is vital for applications requiring advanced graphical presentations.

Chapter 3

Implementation

The first part of this chapter covers a complete list of the final gallery's functionality (Section 3.1), followed by a breakdown of the interface components (Section 3.2). That section explains how the requirements for intuitiveness and accessibility are met, followed by sections explaining the strategy for developing the project (Section 3.3, the project structure (Section 3.4), and implementation details of important functions (Section 3.5).

3.1 Implemented features

1. Explore 3D Gallery
    - Move within the gallery using the keyboard.
    - Adjust the view using the mouse.
    - Browse in full screen mode.
2. View Details of Any Artwork in the Gallery
    - Browse the introduction of the artwork.
    - View enlarged versions of the artwork.
    - Zoom in and out using the mouse and keyboard.
3. Favorite/Unfavorite Artwork in the Gallery
4. View and Hide Gallery Tips.
5. Upload New Artwork
    - Select artwork from local storage for upload.
    - Submit the form.
6. Manage Your Own Artwork
    - Delete artwork.

- Re-edit artwork.
- Download artwork.
- View artwork in enlarged mode.

7. Login, Logout, and Register an Account.

User Types:

1. **Visitor:** Can use Functions 1, 2, and 7 only.
2. **Registered and logged in Student:** Can access all features.
3. **Administrator:** Can access all features and manage artwork for all users in Function 6, not just your own.

3.2 User interfaces

Oppermann (2002) emphasizes that a user-friendly interface is a core issue for the usability of software products. User Interface (UI) design not only directly influences the user experience but also determines the software's overall acceptance and effectiveness. This project prioritizes UI design, focusing on enhancing interactivity and accessibility to improve user engagement and overall experience.

3.2.1 Main page layout

The main page (Figure 3.1) canvas integrates essential 3D models for the gallery and provides interactive entry points for additional functions. This setup serves as the primary interface for users to access needed information and encourages exploration of additional features.

The toolbar (upper right corner) includes a fullscreen button to enhance the immersive experience. The menu bar pops up automatically when hovering over the menu button, allowing users to access various functions through different buttons. A hint box in the lower right corner provides navigation guidance. Users can navigate using the keyboard's WASD and arrow keys, which benefits those with mobility impairments by eliminating reliance on mouse operations. Additionally, the view can be rotated 360 degrees with the mouse or touchpad for a thorough exploration of the gallery.

The central area features the virtual 3D gallery, designed based on real-world art gallery layouts, particularly the New Hall of Art Museum No.1 (Figure 3.2a). The virtual gallery on artsteps.com (Figure 3.2b) also served as a design reference.

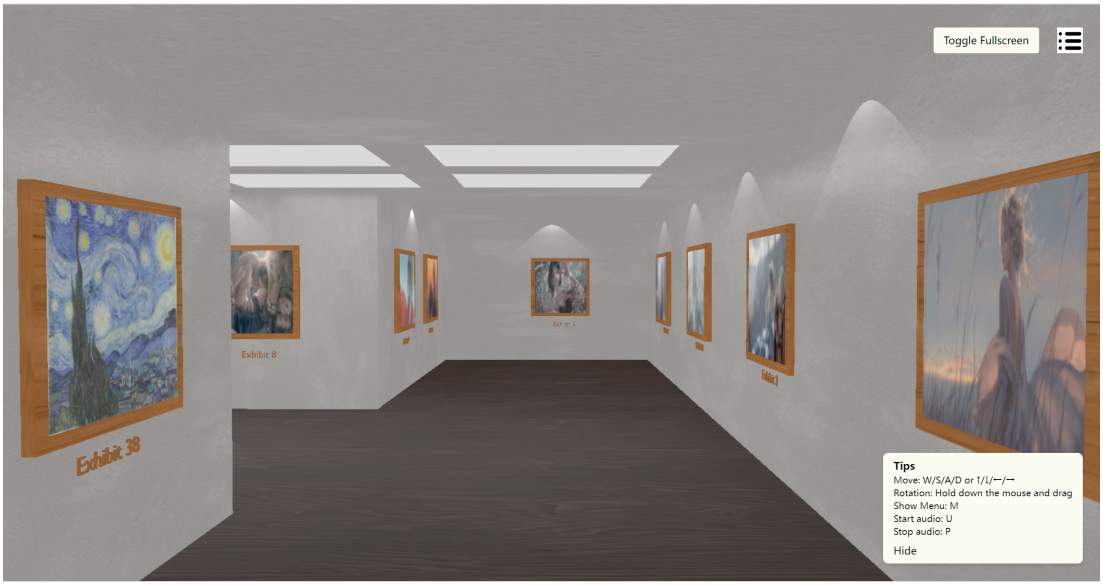
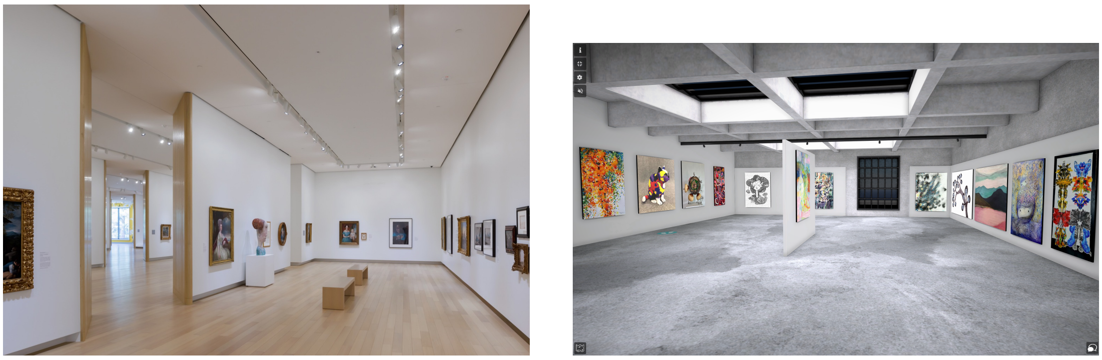


Figure 3.1 Main page layout.



(a) Art Museum No.1. (b) Virtual gallery.

Figure 3.2 Reference galleries.

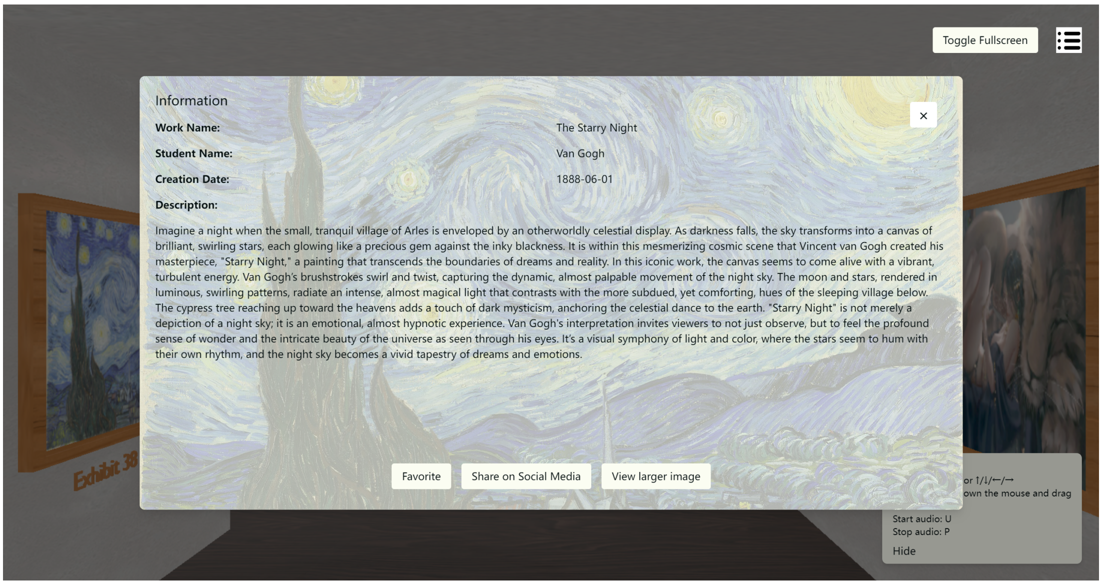


Figure 3.3 Information page layout.

The interface utilizes high-contrast visuals with off-white walls and floors and brown frames, ensuring visibility for users with visual impairments. Artwork highlights in bright red when hovered over, signaling clickability to view detailed information, thus improving responsiveness.

Detailed information (Figure 3.3) for each artwork is categorized into four sections: title, artist's name, creation year, and description. Imagery enhances item visibility (Carretti, Borella and De Beni, 2007). A narrative-themed page aids visitors in quickly understanding the artwork and enriches their overall experience.

The "Share on Social Media" button enables users to share artwork on Facebook, Instagram, or Twitter, facilitating easy social interaction. This feature supports intuitive user engagement with the artwork and expedites content sharing.

Unlike other websites that rely on users' precise operations, the page improves accessibility by shortening the path required for users to perform actions, especially for those using large screens or assistive technologies. Interacting with large screens using a cursor often requires high precision; therefore, the design carefully considers the reasonable size of operational areas and includes hover-triggered hints to facilitate interaction with page elements. While most able-bodied users can quickly reach interaction targets through fine adjustments, this process may be more complex for users with mobility impairments or cognitive disabilities. By enlarging interactive elements and target areas, and incorporating hover effects, the design significantly reduces uncertainty in the interaction process, helping users to more quickly and accurately locate and utilize target elements.

W3C guidelines state a color contrast ratio higher than 4.5:1 can significantly improve the clarity of element and control boundaries, making it easier for users to distinguish different interface elements. For users with poor vision, color blindness, or low contrast sensitivity, content with low color contrast can increase the difficulty of reading and viewing. Therefore, some users require high-contrast colors. However, excessively high contrast can also cause discomfort for people with photosensitivity or dyslexia; for instance, #FFFFFF on a #000000 background may cause eye strain. No single color scheme is universally accessible. Therefore, this project has chosen #040E0F as the text color on a #FBFDF2 background to balance readability and visual comfort. All text color contrasts in this project have passed WCAG testing to ensure compliance with accessibility standards. This design approach not only enhances the accessibility of the user interface but also provides a more comfortable experience for users with varying visual abilities.

3.2.2 Upload page

The design of the upload page (Figure 3.4) facilitates convenient submission of new work by users. The page includes a form where users can fill in the artwork information as prompted and select the file to upload. Upon completion, users click the "Submit" button to submit the form or the "Clear Form" button to reset the form content. After a successful upload, the system will display a confirmation message and place the newly uploaded artwork in Exhibition 1, while other artwork will be moved sequentially to subsequent positions. To enhance user experience, the system also supports an auto-save draft feature, allowing users to continue editing their previous content even if they exit the interface midway and return later. This helps prevent redundant input and mitigates user frustration.

The cleanliness and organization of the interface are crucial considerations in the design of form placeholders. Many websites use placeholders to describe the form content, which disappear once the user starts typing. However, this design can cause users to forget the required content or format, especially for users with dyslexia, who may have more difficulty understanding the text format. To reduce cognitive load, this project employs a design that directly displays field format placeholders by adding clear text outside the input fields to guide users on the specific content needed. This way, users can always know what information to enter, regardless of whether they are about to start typing, are in the process of typing, or have already typed.

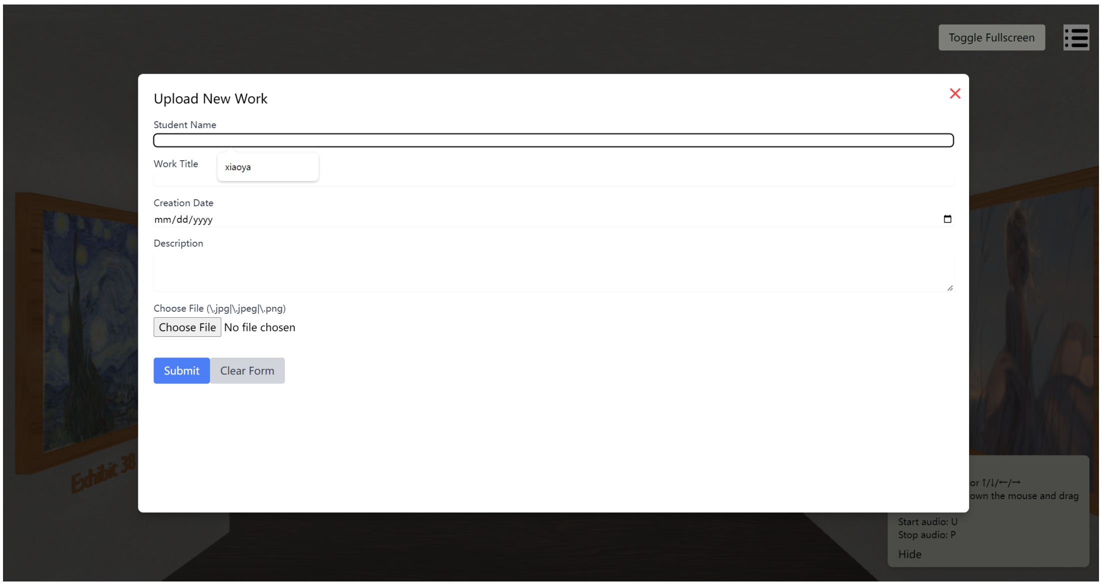


Figure 3.4 Upload page layout.

Convenience of form operation is vital to the user experience. Although many users use a mouse or touchpad, some are keyboard-only. Using the Tab key to switch fields quickly is more convenient than using a mouse. When users switch fields using the keyboard, the selected button and input field will have a bold #000000 color box to indicate the current selection clearly.

The error message design considers the possibility of users inputting errors or incomplete information. When an error occurs, the system will display a high-contrast error message (#AA0303) to prompt users to make corrections. However, it must be assumed that all users have strong color perception abilities. The problems faced by visually impaired users vary; some may be color blind ($^1/_{12}$ men, $^1/_{200}$ women), nearsighted ($^1/_{30}$), or even blind ($^1/_{188}$) (Mangione et al., 1998), and they may use screen readers to navigate the interface. Therefore, in addition to color prompts, clearer error message text and exclamation marks are added to ensure accurate and effective information transmission. To accommodate users with cognitive impairments, error messages use simple and direct language to clearly instruct users on how to correct the errors.

3.2.3 Artwork management page

The artwork management interface offers users convenient tools for managing their artwork. It dynamically adjusts the number of artwork thumbnails displayed per row based on screen size, as illustrated in Figure 3.5. This adaptive design improves the user experience across various devices, providing optimized visual effects and ease of use on both mobile and desktop screens.

Hovering over a piece of artwork reveals multiple action options around the thumbnail: edit, view in detail, delete, and download. The "View Large Image" option supports mouse scrolling, icon buttons, and keyboard shortcuts (L to zoom in, K to zoom out), accommodating a wide range of user preferences.

The system prompts a clear confirmation message after selecting the delete option to prevent accidental deletion. Instead of "Yes/No" the interface uses "Confirm Delete" and "Cancel" buttons. This explicit text significantly reduces cognitive load by making the button functions immediately clear. Such precise labeling minimizes guesswork for users, particularly those using screen readers, and improves accessibility. All button texts feature high color contrast to ensure that users can easily identify these key functions while scanning the page.

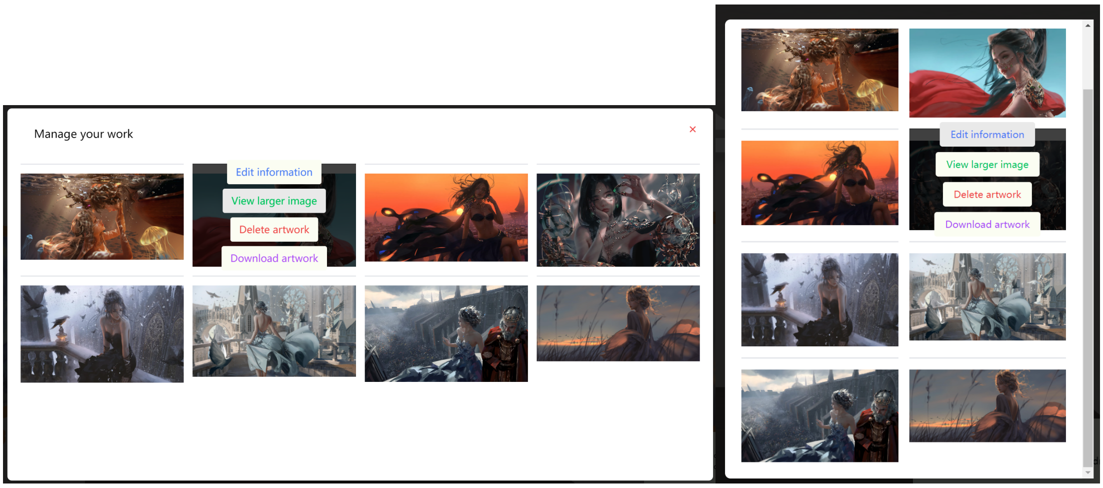


Figure 3.5 Manage page layout.

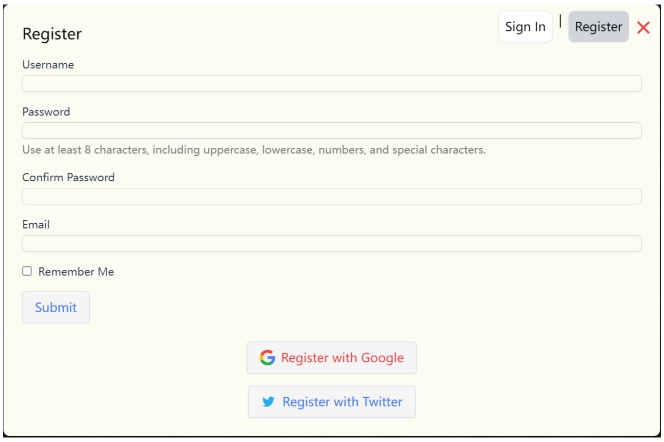


Figure 3.6 Login and register page layout.

3.2.4 Login and register Page

The system simplifies the login process by requiring only an email address for registration rather than an additional username (see Figure 3.6). This approach helps reduce the number of credentials users need to remember and lessens cognitive load during login. Login automatically prompts users about the Caps Lock key when entering a password, helping to prevent orthographical mistakes.

Providing detailed password guidelines near the input field addresses the need for strong password requirements, as users might not always be familiar with them. This feature improves security and reduces frustration from incorrect password entries by including both password input and confirmation steps. Encrypting passwords during transmission and storage ensures data protection. The “Remember

Me" option enhances convenience by saving the email address for automatic entry on future logins, with users having the choice to disable this feature if they prefer.

Offering a registration option through social media accounts allows users to quickly access the application without creating a new account or re-entering information. These design choices aim to streamline user operations and improve the ease of logging in and registering.

3.3 Development strategy

The development strategy for this project primarily employs a modular programming approach, initially addressing problems with simple methods to facilitate testing and identification of any major issues. The initial phase focuses on creating prototype solutions, and as a prototype reaches a milestone, the code is refactored and optimized based on the insights gained. Throughout the project, a to-do list is maintained, and each task is marked off upon completion.

StackOverflow is a valuable resource for problem resolving, with a large number of tagged questions providing solutions, such as fixing mouse perspective control in 3D scenes. GitHub is used for version control, with a branching strategy that maintains stable code in the main branch while allowing feature development in separate branches, effectively managing and tracking code changes.

The project was initially developed in a Ubuntu virtual machine on my own computer but severe performance degradation necessitated abandoning the VM.

Development primarily used Firefox (as required by the supervisor) and Chrome (chosen for its popularity; see Section 2.1.1), while Safari and Edge were utilized for browser compatibility testing. Those browsers use different rendering engines thus testing the project's implementation more stringently.

Switching between browsers reveals notable differences in behavior. For example, the layout of date forms differs significantly between Firefox and Chrome. Mouse navigation in Firefox occasionally fails due to discrepancies in mouse click registration compared to Chrome. These inconsistencies highlight the critical need for cross-browser testing during development.

3.4 Application general architecture

The frontend of this project utilizes the Vue.js framework (Section 2.4.1), written in JavaScript and HTML, and incorporates Vite (Section 2.1.4) for efficient development and build processes. TailwindCSS (Section 2.4.3) is used for style management and responsive design. Vue.js is responsible for building and managing components, with each HTML element stored in separate Vue template files. The main view renders these template files to the web browser. The organization of '.vue' files in the project is shown in Figure 3.7. Each Vue component is an independent unit that encapsulates an HTML template, JavaScript logic, and CSS styles. Components facilitate the decomposition of complex user interfaces into smaller, more manageable blocks. This modular approach aids in organizing code, enhancing maintainability, and improving reusability.

The project employs the Express.js (Section 2.4.2) framework as middleware to parse and handle user requests, manage cross-origin resource sharing, perform user authentication, and other operations. It primarily manages the data for artwork and user interactions with the frontend. On the database side, the project uses PostgreSQL (Section 2.4.4), which integrates with Express.js to store and manage project data. Continuous integration and automated deployment are managed through XNight (Section 2.4.5). This setup ensures that the project undergoes automatic building and testing with each code submission, thereby preventing new features and fixes from disrupting existing functionality. It also enables deployment across different environments for validation purposes.

Figure 3.8 shows the system architecture diagram, focusing primarily on the API. The system adopts a modular and component-based design approach, where each controller (module) contains a set of foundational components with similar functionalities. Although these components vary in complexity, they all belong to the same business module. For instance, the "Artwork Management" controller includes functionality components for uploading, favoriting, deleting, and editing. While there are business dependencies between controllers, most of these dependencies facilitate transitions within the business process. These controllers operate at the same structural level from a system architecture perspective.

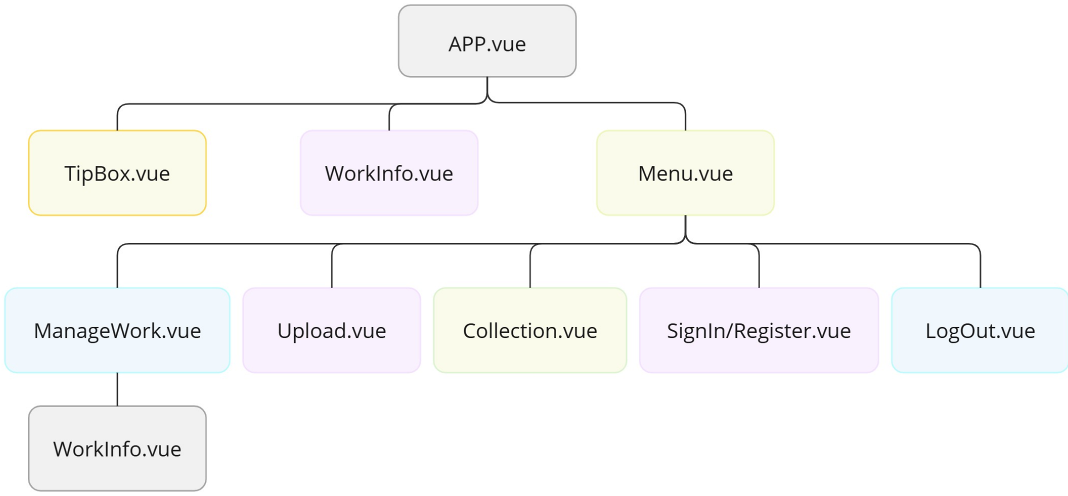


Figure 3.7 Organization of Vue components.

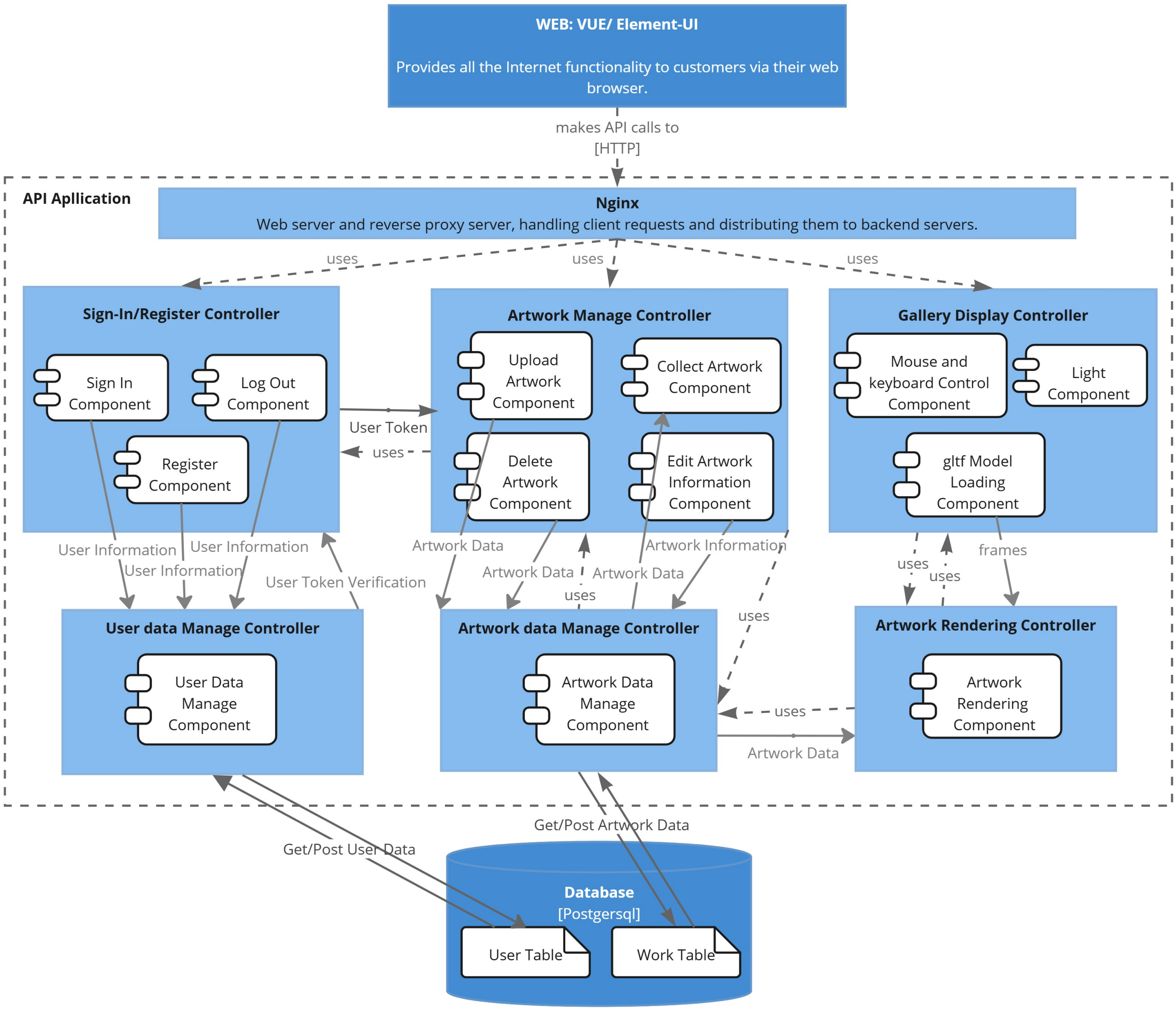


Figure 3.8 System architecture.

3.5 Implementation details

3.5.1 3D gallery modeling

The basic framework of the gallery, including walls, floors, stairs, and picture frames, was manually modeled in Blender. Additionally, a set of 3D exhibit numbers was created in Blender and placed under each frame to indicate the sequence of the artwork, facilitating users in locating the artwork in order. Since the purpose of the modeling process was to achieve visually impressive models, the geometric accuracy of the models was not verified. Textures were obtained from free websites such as Texture Haven and CC0 Textures and were extensively used in the scene. Tileable (repeating) textures were applied to maintain small texture map sizes. Other bitmaps were created to add material properties such as reflectivity, specular reflection, and surface normal details. Semantic information for each picture frame was retained by naming them work1 to work39, to facilitate later interaction with the frontend. Export the model from Blender in the .glft file format (See Section 2.2.3).

The webpage initializes by using the GLTFLoader from the Three.js library to load the .glft format model into the scene. To ensure the model displays correctly in the scene, adjust its position, size, and other parameters. Simultaneously, send an API request to the backend to retrieve the latest 39 artwork from the database. Use Three.js's TextureLoader to load the artwork images one by one; these artwork will be used as textures and sequentially applied to the frames in the scene.

The program enhances the visual quality and ambiance of the scene by using carefully designed lighting in Three.js. Rectangular lights are the main lighting source on the ceiling. This type of light provides even light distribution, filling the entire scene with soft and warm light. Spotlights were installed on the walls to focus on artwork or key areas, highlighting the details and textures of the models.

The program employs a movable virtual camera to view and navigate the virtual scene, allowing users to move freely forward, backward, left, and right within the three-dimensional environment, and to rotate 360 degrees for flexible viewpoint adjustment. Since there is no need to manipulate a large scene in the user interface, visualizing a limited area is sufficient. The data complexity depends on the data to be visualized but also requires appropriate rendering and loading performance from the platform used.

3.5.2 Raycaster

Raycasting is extensively applied in this project to accurately detect the distance between the virtual camera and models. The first application is to prevent the virtual camera from passing through 3D objects, thereby creating collision effects and enhancing the user's interactive experience within the virtual scene. By continuously casting rays from the camera position in the direction of movement, the program can detect if there is an object at the next position and stop the camera's movement as it approaches an object, avoiding model penetration.

Another use is detecting mouse interactions with artwork, specifically identifying clickable hotspots on the webpage, a functionality often associated with 3D visualizations or webpage elements. During implementation, the program first casts a ray from the virtual camera to the mouse position to check if the ray intersects with any artwork. If an intersection is detected, the program will highlight the corresponding object based on predefined logic and, upon user click, execute an event handler that pops up detailed information about the artwork.

3.5.3 Asynchronous loading

Asynchronous loading enhances application performance and responsiveness by allowing tasks to run concurrently without blocking the main thread, optimizing resource utilization. Perceived latency is reduced for improved user experience.

The multiple interdependent libraries and modules involved in this project, some of which are hosted on web servers, necessitate an asynchronous loading mechanism to avoid blocking page rendering. This requirement means there is no guarantee all libraries will have been loaded when needed. For instance, sence.js must be loaded before showWork.js can be used. To address this problem, the 'async' and 'await' methods are employed for asynchronous loading, ensuring that libraries and modules are loaded as needed. Traditionally, JavaScript files or modules are loaded sequentially, with the order being crucial to ensure that one module is correctly loaded before it depends on another module. By using 'async' and 'await', the loading process of libraries and modules can be asynchronous, effectively managing dependencies, avoiding the complexities of sequential loading, and facilitating component testing and maintenance.

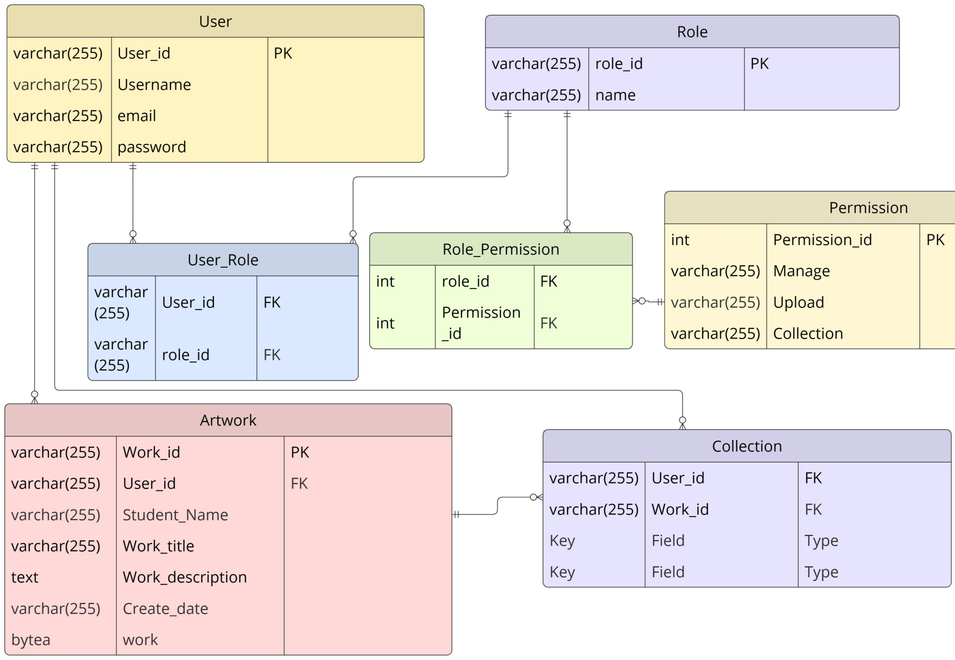


Figure 3.9 Database design, PK = Primary Key, FK = Foreign Key.

3.5.4 Database design

The database, Figure 3.9, uses Role-Based Access Control (RBAC) to simplify permissions management by assigning roles instead of individual permissions, thus enhancing maintainability and scalability. The Artwork table links to users via User_id and tracks all works, the Collection table records user interactions with artwork via Work_id — an efficient way to permissions and collections.

3.5.5 User token and encryption measures

Data encryption is crucial in software systems because it protects sensitive information from unauthorized access, enhancing personal privacy and data security. Additionally, encryption helps comply with legal regulations and prevent potential financial losses. The user login process sends an email and password to the server, which issues a JSON Web Token (JWT) upon verification. This token, stored in Local Storage, is used for future authentication requests, reducing server load.

The project incorporates encryption measures to safeguard sensitive user data, with a focus on protecting passwords and email addresses. Bcrypt is used to hash user passwords. This hashing algorithm employs a one-way encryption method, ensuring that identical plaintext will generate different ciphertexts each time it is hashed. The algorithm is designed to be resistant to reverse-engineering, making it highly secure against cracking attempts. Additionally, bcrypt includes a tunable cost factor, which increases the time required for hashing and provides increasing protection against attacks as computational power advances. Bcrypt also utilizes a salt—an additional random value added to passwords before hashing. This prevents identical passwords producing identical hashes, thereby mitigating attacks that use precomputed hash tables. By adding this complexity, bcrypt enhances the defense against attacks based on hash value matching. Email addresses are encrypted alongside passwords to protect user identity. This extra layer of encryption ensures that, even if the database is breached, email addresses remain secure from unauthorized access and misuse.

Chapter 4

User study

The user study comprised two stages: Alpha testing and Beta testing. Alpha testing employed Cognitive Walkthrough (CW) (Polson et al., 1992) to evaluate the system, aiming to test the preliminary version and identify potential problems users might encounter when first interacting with the interface. Beta testing involves recruiting interested participants for face-to-face testing, and assesses the system's usability through the System Usability Scale (SUS) (Brooke, 1996) to indicate ease-of-use and the Net Promoter Score (NPS) (Reichheld, 2003) to indicate the likelihood of recommending the system to others. It is worth noting that the comprehensive evaluation and conclusions regarding the system have been integrated into this chapter to present the assessment results more coherently.

4.1 Alpha testing

A series of tasks were developed early in the project based on the Cognitive Dimensions Framework to address usability. This framework is widely used in Human-Computer Interaction (HCI) to evaluate the effectiveness of interactive systems. Bennett and Stephens (2009) describe the Cognitive Walkthrough (CW) method, developed in the 1990s, as a task-specific evaluation technique suitable for assessing individuals or groups during the design and deployment stages of interactive systems. Lewis and Rieman (1993) emphasize that this method helps developers identify interface design problems within User-Centered Design and user roles, without requiring actual users to conduct tests, which is useful for student projects. CW is deployed through five processes: **1:** Identify and define the target user group. **2:** Define each task. **3:** Design the actions to perform each task. **4:** Record key information from the tasks. **5:** Evaluate and discuss the recorded information for further improvement.

Step 1: Who will be the users of the system. Section 3.1 presents the potential users targeted by the system in a virtual gallery setting: general visitors, registered and logged-in students, and administrators.

Step 2: Define tasks. The second step is to define the tasks to be analyzed in the cognitive context. To develop representative tasks, the tasks were decomposed into multiple questions based on the Cognitive Dimensions framework. This framework, proposed by Green and Petre (1996), is used to evaluate the design of interactive devices and programming languages in HCI. Initially introduced in 1996, the framework includes 14 dimensions: abstraction gradient, closeness of mapping, consistency, diffuseness/terseness, error-proneness, hard mental operations, hidden dependencies, juxtaposability, premature commitment, progressive evaluation, role expressiveness, secondary notation and escape from formalism, viscosity, and visibility. In this study, these dimensions were used as the basis for tasks to assess the effectiveness of user experience, and a CW was conducted. Below are the tasks of interest for different types of users:

For logged-in students:

1. Students can effectively browse the entire gallery without prior training.
2. Students can accurately and conveniently view information about artwork of interest during their visit.
3. Students can easily share their favorite artwork on social media.
4. Students can effortlessly upload new artwork to the gallery and locate the exhibition site of the uploaded pieces.
5. Students can re-edit and modify the content they have uploaded.
6. Students can safely and quickly delete their own artwork.
7. For all functions in Section 3.1, if an error occurs in operation or decision-making, users can go back and reattempt the actions.
8. Students with disabilities (hearing impairment, visual impairment, cognitive disabilities, and mobility impairment) can access all functions in Section 3.1.

For visitors:

1. The system should attract visitors to register an account.
2. Visitors should be able to register and log in quickly and easily.

For administrators:

1. Administrators should be able to easily manage (delete and modify) all artwork.

Step 3: Define the sequence of actions for each task. After defining the input topics and tasks, it is essential to detail the actions and sequence for each task. For instance, **Task 4 for students (S4)** involves:

1. Locate the entry point for uploading work by observing the 3D gallery's main interface.
2. Open the options list by clicking the menu button and then select the "Upload New Work" button.
3. If the page prompts that login is required, complete the login process.
4. Access the login and registration interface by clicking the menu button again, then finish logging in.
5. After logging in, click the "Upload New Work" button once more to enter the upload page.
6. Fill in the work information and complete the upload following the prompts.
7. Correct and retry if an error occurs due to non-compliant information by following the error message.

For visitors, the sequence of actions defined as Task 1 (V1) is:

1. Begin exploring the 3D gallery's main interface and navigate using the keyboard and mouse, guided by the hint box in the lower right corner.
2. View detailed information by moving the mouse over an interesting artwork and clicking on it.
3. Attempt to share the artwork on social media by clicking the share button; a login prompt will appear.
4. Find the login entry on the main interface and proceed with logging in.

Step 4: Record key information. Recording key information by doing defined tasks can reveal weaknesses in the interactive system and assess UI efficiency. Lewis and Rieman (1993) recommend four key questions to evaluate tasks:

1. Does the user attempt to achieve any effects produced by the action? This ensures that users can pursue logical outcomes?
2. Does the user notice that the correct action is available? This evaluates the user-friendliness of the design interface?
3. Once the user finds the correct action on the interface, do they recognize it as producing the desired effect? This ensures an intuitive understanding of each function.

4. After taking the action, does the user understand the feedback received? This guarantees that the system provides appropriate feedback after the action is completed.

Considering these questions helps distinguish between successful and unsuccessful tasks. Tasks are completed based on the answers to questions A–D defined by cognitive dimensions. To effectively extract key information from the tasks, each step was thoroughly evaluated. Table 4.1 shows the application of questions A–D for each operation. The symbol “Y” (Yes) indicates that the operation is clearly defined and can be completed without additional assistance, while “N” (No) suggests a potential usability issue that may disrupt the interaction experience. The final results (“R”) are recorded as “P” (Pass) or “F” (Fail). Table 4.2 lists the results (‘F’/‘P’) for each task defined in Section 4.1.

Step 5: Results and discussion. The results of the operational evaluation (Table 4.1) indicate that most actions met the initial objectives, except operations “a” and “d” in S4 and operation “b” in V1. Addressing problems with operations (a) and (d) in Task 4 can be achieved by adding prompts and guidance. A prominent upload button on the main interface might resolve these problems, although this approach could affect the design’s aesthetics. Operation “b” in Task 1 could be improved by using dynamic effects to signal that the artwork is clickable for more details. Adding a tooltip next to the artwork could indicate its interactivity.

S4	**(a)**	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**	**(g)**	**V1**	**(a)**	**(b)**	**(c)**	**(d)**
A	N	Y	Y	N	Y	Y	Y	**A**	Y	Y	Y	N
B	N	Y	Y	N	Y	Y	Y	**B**	Y	Y	Y	Y
C	Y	Y	Y	Y	Y	Y	Y	**C**	Y	N	Y	Y
D	Y	Y	Y	Y	Y	Y	Y	**D**	Y	Y	Y	Y
R	**F**	P	P	**F**	P	P	P	**R**	P	P	P	P

Table 4.1 Task execution results.

S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**V1**	**V2**	**A1**
P	P	P	P	P	P	P	**F**	**F**	P	P

Table 4.2 Analysis results of each task.

Table 4.2 shows the failure of Task S8 stems from incomplete functionality due to time constraints. Although most accessibility features have been implemented, comprehensive support for all users cannot be ensured. For instance, blind users are unable to navigate the gallery using a screen reader. The failure of Task V1 arises from the current system's lack of attractiveness. The system lacks features that could enhance user experience, such as a recommendation system and gallery categorization. Future improvements could include adding these features to increase the system's appeal and overall effectiveness for visitors.

4.2 Beta testing

The beta testing phase recruited 9 participants from the University of Edinburgh. This group comprised 6 women and 3 men, aged between 22 and 24 years. All participants had normal or corrected-to-normal vision and did not have common accessibility impairments, which means that the evaluation did not include assessments related to accessibility features. The core objective of this test was to identify potential usage problems and assess user satisfaction by observing the actual behaviour of the participants during the use of the system. After individually completing the system testing, participants were asked to fill out two surveys: SUS and NPS.

The tests reveal that testers took time to familiarize themselves with the navigation when they first started using the system. The testers spent the most time on the first piece of artwork while browsing the gallery, with a mean duration of approximately 34 seconds and a median of 32 seconds. They not only examined the detailed description of the piece but also used the zoom feature to inspect the image's details and tried out the functionality to share the artwork on social media, exploring all features available on the interface. Each tester used the navigation system and thoroughly explored every section of the gallery.

Six testers actively registered and logged into accounts during the testing period, conducting a comprehensive evaluation of all system features. This behavior indicates a strong interest from testers and demonstrates that the system has a certain appeal in engaging users to explore and use its functions.

The System Usability Scale (SUS), developed by John Brooke in 1986 (republished 1996), evaluates user perceptions of the usability of a system or product. This scale consists of 10 items (Table 4.3), with odd-numbered items featuring positive statements and even-numbered items featuring negative statements. Participants score the SUS after completing tasks. Each question is scored on a scale from 0 to 4, with 0 indicating "Strongly Disagree" and 4 indicating "Strongly Agree." Scoring follows these rules: subtract 1 from the raw score for odd-numbered items, and subtract the raw score from 5 for even-numbered items to obtain the reversed score. Sum the adjusted scores for all 10 questions to get the total score. Multiply the total score by 2.5 to convert it to a range of 0 to 100.

Studies have demonstrated SUS's effectiveness in assessing system usability. Research by Tullis and Stetson (2004) shows that SUS provides valid results quickly, even with a limited sample size (Figure 4.1). The final SUS score for the system is 72.4, which is considered acceptable (Figure 4.2).

Net Promoter Score (NPS), developed by Reichheld (2003), measures customer loyalty and satisfaction. A strong correlation was observed between the Net Promoter Score (NPS) and the average growth rate of the company across many sectors, including the aviation industry, over the three-year period from 1999 to 2002. Consequently, Reichheld (2003) concluded that NPS results could explain the growth rate across the entire industry. Although NPS has faced criticism due to misuse and may lack the rigor of some mathematical measures, it remains a useful indicator. It uses a single question to gauge the likelihood of recommending a product or service: "On a scale from 0 to 10, how likely are you to recommend our product or service to a friend or colleague?"

Respondents are classified into three categories based on their scores: Promoters (9-10), Passives (7-8), and Detractors (0-6). NPS is calculated by subtracting the percentage of Detractors from the percentage of Promoters, with scores ranging from -100 to +100. A higher NPS reflects a positive customer experience and a greater likelihood of customers recommending the product or service.

The survey results show that the system has 3 promoters, 6 passives, and 0 detractors, with an NPS score of 33.33. In comparison, Verint's 2018 survey reported NPS scores of 64 for Netflix, 63 for PayPal, 54 for Amazon, 53 for Google, and 49 for Apple. According to Retently's 2024 NPS benchmark report, the average NPS score for the Internet B2B Software & SaaS industry is 30 (Raileanu, 2024). A score between 0 and 30 is considered acceptable in the SaaS industry, so the system's score of 33.33, slightly above the industry average, indicates that it ranks in the upper-middle range for user loyalty and satisfaction.

№	Question
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

Table 4.3 System Usability Scale (SUS) Questionnaire.

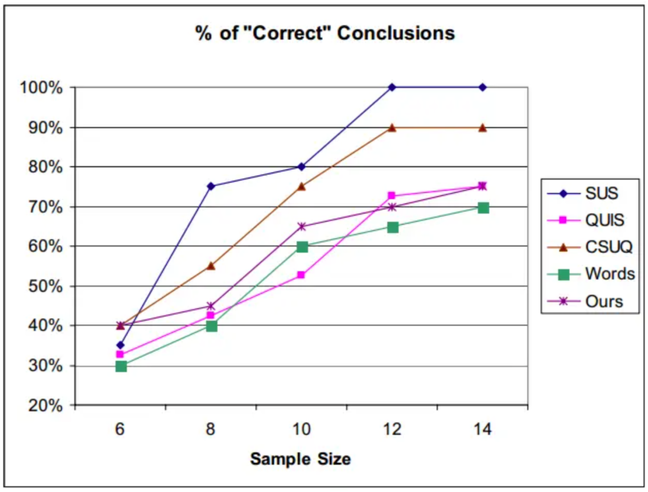


Figure 4.1 Comparison of SUS with other measurement methods.

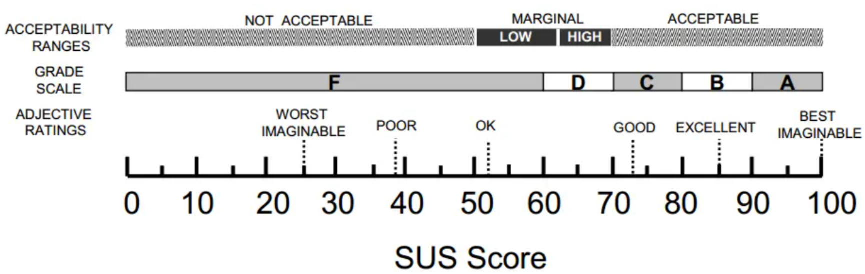


Figure 4.2 A comparison of the adjective rating, acceptability scores, and school grading scales, in relation to the mean SUS score.

4.3 Analysis and conclusions

The results of the Alpha and Beta tests indicated that the system was within acceptable limits and that users were able to perform basic operational tasks, suggesting that there were no significant problems with the usability of the system. However an in-depth analysis of user behaviour and problem feedback during continuous use revealed some noteworthy observations.

1. Users generally reported that it took a long time to get used to the navigation system. This issue did not completely prevent users from completing their tasks, but it clearly affected the smoothness of operation and overall experience. The design of the navigation system needs to be further optimized to shorten the learning curve for users.
2. The lack of interactivity and attractiveness of the system was an obvious problem. Although users were able to complete basic tasks, the lack of interactive aspects made the user experience monotonous and boring. Future designs should consider adding more rich interactive content to improve the attractiveness of the system and user engagement.
3. Users experienced difficulties in locating specific works. This suggests that the system is deficient in providing accurate navigation and positioning information, leading to user confusion when searching for specific content. Improving the location function would be an important direction to enhance user experience.
4. The lack of personalization in the interface design is also a major problem. The user interface fails to fully consider the individual needs of users, resulting in a system that is more generic but lacks personalized appeal. Future designs should pay more attention to the personalization of the interface to better meet the preferences of different users.

Chapter 5

Conclusions

5.1 Project summary

This project developed a browser-based 3D gallery that leverages advanced 3D technology to create a realistic, interactive virtual exhibition space where users can both showcase their own artwork and appreciate the works of others. Built on a single-page application framework, the system includes a data layer for handling 3D models, artwork, and user information, a technology support layer for rendering and optimizing performance, and an application layer designed to enhance accessibility and user experience. The gallery's 3D scenes are created using Three.js and Blender, offering high-resolution textured models that allow users to interact with the artwork from various angles. Additionally, a backend system was implemented for easy uploading, managing, and displaying of artwork in the virtual museum.

The project placed significant emphasis on improving accessibility, interactivity, and user experience through detailed page design. User evaluations resulted in a SUS score of 72.4 and a NPS of 33.33, indicating an overall positive reception with room for further enhancement.

5.2 Future work

User feedback highlights several key areas for future work.

Simplified navigation systems. Implementing more intuitive navigation systems is critical, with a focus on reducing the learning curve for users. Enhancements may include guided tutorials, tooltips or adaptive interfaces that adjust to user behaviour to facilitate smoother interactions from the outset.

Increased interactive features. These may include interactive 3D models with detailed annotations, AI-guided virtual tours, and gamification elements that encourage exploration and interaction with the artwork.

Advanced search and location features. This involves the implementation of an intelligent search system that allows users to filter results based on a variety of criteria, as well as an accurate indoor location system that helps users to easily find specific works in the virtual environment.

Personalisation and recommendation systems. The addition of advanced recommendation algorithms that recommend artwork based on user preferences and browsing history can create a more personalised user experience. In addition, features such as a customisable interface and user-driven content organisation will allow users to tailor the experience to their personal preferences.

5.3 Reflections

I thoroughly enjoyed developing full-stack applications and engaging in 3D modeling. Integrating the back-end system for managing artwork with the front-end display of 3D models provided a fulfilling experience through its technical challenges and creative solutions.

The biggest regret is not implementing more interactive features. Although the gallery's core functionality was completed, the user experience remained monotonous due to a lack of interactive elements. Given more time and resources, I would focus on adding dynamic user interactions, gamification elements, and personalized experiences.

For those starting similar projects, I recommend conducting thorough research and developing a detailed design plan before beginning development. Understanding current design trends, user needs, and best practices, as well as continuously incorporating user feedback into the design, will help ensure that the project better meets user expectations.

References

Beer, V., 1987. Great expectations: do museums know what visitors are doing? *Curator: the museum journal*, 30(3), pp.206–215 (cit. on p.1).

Bennett, D.J. and Stephens, P., 2009. A cognitive walkthrough of autopsy forensic browser. In: *Information management & computer security*. New York: John Wiley & Sons (cit. on p.31).

Brinck, T., Gergle, D. and Wood, S.D., 2001. *Usability for the web: designing web sites that work*. Elsevier (cit. on p.11).

Brooke, J., 1996. Sus: a quick and dirty usability scale. *Usability evaluation in industry*. Original paper on the System Usability Scale., pp.189–194 (cit. on pp.31, 35).

Brooks, F.P., 1999. What's real about virtual reality? *Ieee computer graphics and applications*, 19(6), pp.16–27 (cit. on p.9).

Caldwell, B., Cooper, M., Reid, L.G., Vanderheiden, G., Chisholm, W., Slatin, J. and White, J., 2008. Web content accessibility guidelines (wcag) 2.0. *Www consortium (w3c)*, 290(1-34), pp.5–12 (cit. on p.12).

Carretti, B., Borella, E. and De Beni, R., 2007. Does strategic memory training improve the working memory performance of younger and older adults? *Experimental psychology (formerly "zeitschrift für experimentelle psychologie")* [Online], 54(4), pp.311–320. Available from: https://doi.org/10.1027/1618-3169.54.4.311 (cit. on p.20).

Cockburn, A. and McKenzie, B., 2001. 3d or not 3d? evaluating the effect of the third dimension in a document management system. *Proceedings of the sigchi conference on human factors in computing systems*, pp.434–441 (cit. on p.9).

Evans, A., Romeo, M., Bahrehmand, A., Agenjo, J. and Blat, J., 2014. 3d graphics on the web: a survey. *Computers & graphics*, 41, pp.43–61 (cit. on p.9).

Green, T.R.G. and Petre, M., 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of visual languages & computing* [Online], 7(2), pp.131–174. Available from: https://doi.org/10.1006/jvlc.1996.0009 (cit. on p.32).

Higuera, J.F.D. and Pernas, F.J.D., 1999. Minerva: multimedia on the internet for virtual arts. *Proceedings ieee international conference on multimedia computing and systems*. Vol. 2. IEEE, pp.825–829 (cit. on p.9).

Krasner, G.E., Pope, S.T. et al., 1988. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3), pp.26–49 (cit. on p.7).

Krug, S., 2000. *Don't make me think!: a common sense approach to web usability*. Pearson Education India (cit. on p.11).

Leff, A. and Rayfield, J.T., 2001. Web-application development using the model/view/controller design pattern. *Proceedings fifth ieee international enterprise distributed object computing conference*. IEEE, pp.118–127 (cit. on p.8).

Lewis, C. and Rieman, J., 1993. *Task-centered user interface design: a practical introduction*. Boulder: Department of Computer Science, University of Colorado (cit. on pp.31, 33).

Macrae, C., 2018. *Vue. js: up and running: building accessible and performant web apps*. " O'Reilly Media, Inc." (cit. on p.8).

Mangione, C.M., Lee, P.P., Pitts, J., Gutierrez, P., Berry, S., Hays, R.D., Investigators, N.-V.F.T. et al., 1998. Psychometric properties of the national eye institute visual function questionnaire (nei-vfq). *Archives of ophthalmology*, 116(11), pp.1496–1504 (cit. on p.22).

Marcos, D., Ngo, K., Yee, C. and Carpenter, J., 2017. A-frame: a web framework for building virtual reality experiences. *Proceedings of the 23rd acm symposium on virtual reality software and technology (vrst)* [Online]. ACM, 37:1–37:2. Available from: https://doi.org/10.1145/3139131.3141212 (cit. on p.10).

Marty, P.F., 2008. Museum websites and museum visitors: digital museum resources and their use. *Museum management and curatorship*, 23(1), pp.81–99 (cit. on p.1).

McHenry, K. and Bajcsy, P., 2008. An overview of 3d data content, file formats and viewers. *National center for supercomputing applications*, 1205, p.22 (cit. on p.10).

Mikowski, M. and Powell, J., 2013. *Single page web applications: javascript end-to-end*. Simon and Schuster (cit. on p.7).

Muller, A., Leissler, M., Hemmje, M. and Neuhold, E., 1999. Towards the virtual internet gallery. *Proceedings ieee international conference on multimedia computing and systems*. Vol. 2. IEEE, pp.214–219 (cit. on p.9).

Oppermann, R., 2002. User-interface design. *Handbook on information technologies for education and training*, pp.233–248 (cit. on p.18).

Parisi, T., 2012. *Webgl: up and running*. " O'Reilly Media, Inc." (cit. on p.9).

Polson, P.G., Lewis, C., Rieman, J. and Wharton, C., 1992. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International journal of man-machine studies* [Online], 36(5), pp.741–773. Available from: https://doi.org/https://doi.org/10.1016/0020-7373(92)90039-N (cit. on p.31).

Rahaman, H. and Champion, E., 2019. The scholarly rewards and tragic irony of 3d models in virtual heritage discourse. *24th annual conference of the association for computer-aided architectural design research in asia (caadria 2019)*, pp.695–704 (cit. on p.10).

Raileanu, G., 2024. *2024 nps benchmarks by industry* [Online], March. Available from: https://www.retently.com/blog/good-net-promoter-score [Accessed 20 August 2024] (cit. on p.36).

Reichheld, F.F., 2003. The one number you need to grow. *Harvard business review*, 81(12), December. PMID: 14712543, pp.46–54 (cit. on pp.31, 36).

Rutter, R., Lauke, P.H., Waddell, C., Thatcher, J., Henry, S.L., Lawson, B., Kirkpatrick, A., Heilmann, C., Burks, M.R., Regan, B. et al., 2007. *Web accessibility: web standards and regulatory compliance*. Apress (cit. on p.11).

Schilling, A., Bolling, J. and Nagel, C., 2016. Using gltf for streaming citygml 3d city models. *Proceedings of the 21st international conference on web3d technology*, pp.109–116 (cit. on p.10).

Scott Jr, E.A., 2015. *Spa design and architecture: understanding single-page web applications*. Simon and Schuster (cit. on p.8).

Shreiner, D. et al., 2009. *Opengl programming guide: the official guide to learning opengl, versions 3.0 and 3.1*. Pearson Education (cit. on p.9).

Tullis, T.S. and Stetson, J.N., 2004. A comparison of questionnaires for assessing website usability. *Proceedings of the usability professionals' association (upa) conference*. Chicago, IL: Usability Professionals' Association (cit. on p.36).

Verint, 2018. *2018 net promoter scores by industry* [Online]. Available from: https://www.verint.com/ [Accessed 20 August 2024] (cit. on p.36).

Vyas, R., 2022. Comparative analysis on front-end frameworks for web applications. *International journal for research in applied science and engineering technology*, 10(7), pp.298–307 (cit. on p.8).

W3Counter, 2024. *July 2024 market share* [Online]. Accessed 5th September 2014. Available from: http://www.w3counter.com/globalstats.php?year=2024&month=7 (cit. on pp.5, 6).

Appendix A

Accessibility

A.1 Common types of accessibility barriers

- **Visual Impairments**
  - **Blindness:** Users who are completely blind rely on screen readers or Braille displays to access web content.
  - **Low Vision:** Users with partial vision loss may need to magnify screen content or adjust contrast to see more clearly.
  - **Color Blindness:** Users with color vision deficiency may not be able to distinguish certain color combinations, affecting the readability and understanding of content.
- **Hearing Impairments**
  - **Deafness:** Users who are completely deaf cannot hear audio content and require captions, text descriptions, or sign language translations to access information.
  - **Hearing Loss:** Users with partial hearing loss may need to adjust the volume or use hearing aids to better understand audio content.
- **Motor Impairments**
  - **Physical Disabilities:** Users with conditions such as paralysis, arthritis, or repetitive strain injuries (e.g., carpal tunnel syndrome) may be unable to use a mouse or keyboard and need assistive technologies

like voice recognition, eye-tracking, or switch devices to interact with a computer.

- **Cognitive and Learning Disabilities**
  - **Dyslexia:** Users with dyslexia may struggle to comprehend complex text and require simplified language or assistive reading tools.
  - **Attention Deficit Disorders:** These users may easily become distracted and benefit from clear, concise content and intuitive navigation.
  - **Memory Impairments:** Users with conditions like Alzheimer's disease may need clear navigation structures and consistent interface designs to help them remember how to use the website.
- **Language Barriers**
  - **Non-native Speakers:** These users may need simplified language or multilingual support to understand the content.
  - **Text Recognition Challenges:** Some users may struggle to recognize or understand specific words or symbols and need alternative symbols or images to aid comprehension.

Appendix B

Ethics information

B.1 Participants' information sheet

Participant Information Sheet

Project title:	Browser-based 3D gallery to showcase student work
Principal investigator:	Brian Mitchell
Researcher collecting data:	Xiaoya Zhu

This study was certified according to the Informatics Research Ethics Process, reference number 372749. Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers?

Xiaoya Zhu, Brian Mitchell

What is the purpose of the study?

The purpose of the study is to evaluate the usability and effectiveness of a browser-based 3D gallery designed to showcase student artwork. The research aims to assess user experience, interaction patterns, and overall satisfaction with the gallery interface. Insights gathered will inform iterative design improvements to enhance user engagement and artistic presentation.

Why have I been asked to take part?

You have been asked to take part in this study because you are a [describe research target group, e.g., student, artist, educator] and your feedback and experience are valuable for evaluating the usability and effectiveness of the browser-based 3D gallery. Your participation will contribute to improving the design and functionality of the gallery, benefiting both users like yourself and future audiences of the platform.

Do I have to take part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time, up until 01/08/2024 without giving a reason. After this point, personal data will be deleted and anonymised data will be combined such that it is impossible to remove individual information from the analysis. Your rights will not be affected. If you wish to withdraw, contact the PI. We will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part?

If you decide to take part in the study:

- We will collect data primarily through interaction with the browser-based 3D gallery. This includes your navigation patterns, interactions with different features, and any feedback you provide regarding your user experience.

- Data will be collected passively as you explore the gallery.

- The duration of your session will depend on your level of engagement with the gallery, but typically sessions last between 5 to 10 minutes.

- No participant audio/video will be recorded.

- You can access the gallery at your convenience, anytime and anywhere with an internet connection. There are no specific time or location requirements for participation.

Are there any risks associated with taking part?

There are no significant risks associated with participation in this study. Participants will primarily engage with a browser-based 3D gallery, and no sensitive personal information will be collected. Additionally, there are no audio or video recordings involved, ensuring privacy and confidentiality.

Are there any benefits associated with taking part?

There are no direct benefits associated with participation, there is an opportunity for participants to contribute to the improvement of the gallery interface, benefiting future users. Additionally, compensation such as Amazon vouchers may be provided for participants' time and feedback.

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a maximum of 4 years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researchers Xiaoya Zhu and Brian Mitchell.

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk.

What are my data protection rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

Who can I contact?

If you have any further questions about the study, please contact the lead researcher, Xiaoya Zhu, s2511927@ed.ac.uk.

If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be made available on http://web.inf.ed.ac.uk/infweb/research/study-updates.

Alternative formats.

To request this document in an alternative format, such as large print or on coloured paper, please contact Xiaoya Zhu, s2511927@ed.ac.uk.

General information.

For general information about how we use your data, go to: edin.ac/privacy-research

Participant number:______________________

B.2 Participants' consent form

Participant Consent Form

Project title:	Browser-based 3D gallery to showcase student work
Principal investigator (PI):	Brian Mitchell
Researcher:	Xiaoya Zhu
PI contact details:	brian.x.mitchell@ed.ac.uk

By participating in the study you agree that:

- I have read and understood the Participant Information Sheet for the above study, that I have had the opportunity to ask questions, and that any questions I had were answered to my satisfaction.
- My participation is voluntary, and that I can withdraw at any time without giving a reason. Withdrawing will not affect any of my rights.
- I consent to my anonymised data being used in academic publications and presentations.
- I understand that my anonymised data will be stored for the duration outlined in the Participant Information Sheet.

Please tick yes or no for each of these statements.

1. I allow my data to be used in future ethically approved research.

Yes	**No**

2. I agree to take part in this study.

Yes	**No**

Name of person giving consent	Date dd/mm/yy	Signature
Name of person taking consent	Date dd/mm/yy	Signature

THE UNIVERSITY *of* EDINBURGH informatics