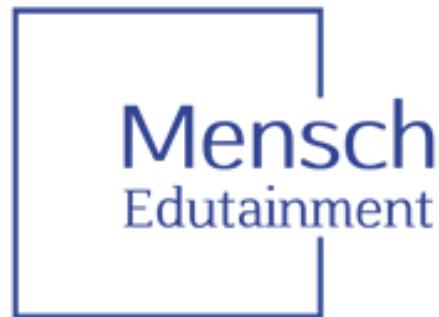




Capstone Project Phase B

Interactive Museum Guide and Content Creation System for Mensch Edutainment



24-2-D-5

Final Project by:

Shay Garbuz
May Caspi

Supervisor:

Dr. Julia Sheidin

| | |
|--|-----------|
| Abstract..... | 3 |
| 1. General Description..... | 4 |
| 1.1 Background..... | 4 |
| 1.2 Goals and Stakeholders..... | 4 |
| 1.3 Integration of AI Tools..... | 6 |
| 1. Automated Text and Audio Generation..... | 6 |
| 2. Creating Descriptions Based on a Photo and Title..... | 6 |
| 3. Improving Existing Descriptions..... | 6 |
| 2. Solution Description..... | 6 |
| 2.1 Content Management System (CMS)..... | 7 |
| 2.2 Visitor Mobile Application..... | 7 |
| 2.3 Backend Infrastructure..... | 8 |
| Overall Architecture:..... | 9 |
| 3. Research and Development Process..... | 10 |
| 3.1 Research Process..... | 11 |
| 3.1.1 Initial User Research..... | 11 |
| 3.1.2 Market Analysis..... | 11 |
| 3.1.3 Literature Review..... | 11 |
| 3.1.4 Requirement Gathering and Design..... | 12 |
| 3.2.2 Development Phases..... | 13 |
| Phase 1: Core System Development..... | 13 |
| Phase 2: AI Integration..... | 14 |
| 3.2.3 Client Collaboration..... | 15 |
| 3.2.4 Tools and Technologies..... | 15 |
| 4. Challenges and Solutions..... | 16 |
| Results and Conclusions..... | 17 |
| 5. Results..... | 18 |
| 5.1 Functional Success..... | 18 |
| 5.2 Feedback and Expectations..... | 18 |
| 5.3 Real-Time Updates..... | 19 |
| 5.4 Performance Improvements..... | 19 |
| 5.5 AI Integration..... | 19 |
| 5.6 Technology Stack Selection..... | 19 |
| 6. Lessons Learned..... | 20 |
| 7. Project Benchmarks..... | 21 |
| 8. User Guide..... | 22 |
| Overview..... | 22 |
| Welcome Screen..... | 22 |
| Sign up screens:..... | 23 |
| Update Artwork Screen..... | 47 |
| Create Artwork Screen..... | 48 |

| | |
|--|-----------|
| Edit Your Information..... | 48 |
| 9. Maintenance guide..... | 55 |
| MuseumApp Backend Programmer's Guide..... | 55 |
| MuseumApp CMS Frontend Programmer's Guide..... | 60 |
| MuseumApp Visitor Frontend Programmer's Guide..... | 65 |

Git Repository: <https://github.com/garbzshay/FinalProject>

Project Deployments:

1. CMS: <https://final-project-jwpy.vercel.app/>
2. Visitor App: <https://mensch-visitors.vercel.app/>

Abstract

The "Interactive Museum Guide and Content Creation System" bridges traditional museum experiences with the modern demands of digital transformation. The project is defined by three distinct outputs that together create a seamless and innovative solution. (1) The system simplifies the creation and management of multimedia content, enabling museum staff to produce and curate engaging materials such as text, images, and audio with ease. This eliminates the traditional complexities and costs associated with content updates and creation. (2) It features an intuitive Content Management System (CMS) designed specifically for museum owners. This CMS allows them to independently create and manage custom mobile applications for their exhibits. Visitors interact with these apps, which provide an engaging, multimedia-rich way to explore exhibits.(3) Finally, the project includes a comprehensive management tool for the client, Menachem, to oversee and coordinate the entire platform. This ensures smooth operation and alignment with the goals of the institution while providing a centralized view of activities. Together, these outputs modernize museum operations, enrich visitor experiences, and provide a cost-effective and scalable tool for digital cultural storytelling.

Phase B focused on refining system architecture, implementing all features, addressing challenges from Phase A, and validating the system against defined success criteria. This document outlines the research process, technical achievements, challenges faced, and results, offering a comprehensive understanding of the project's scope and outcomes.

1. General Description

1.1 Background

Museums serve as pivotal hubs of cultural life and learning. The International Council of Museums (ICOM) states: “A museum is a nonprofit making, permanent institution in the service of society and of its development, and open to the public, which acquires, conserves, research, communicates and exhibits, for purposes of study, education and enjoyment, material evidence of people and their environment” [Davis, 2020]. Following a new orientation to the work of museums, they not only preserve and exhibit artifacts and artworks from various epochs and regions but also act as spaces where individuals can immerse themselves in a rich tapestry of human civilization, knowledge, enjoyment, reflection, and artistic expression. Their role is paramount in fostering understanding, appreciating, and critically engaging with human history and creativity [Wood, 2023]. This project addresses these challenges by providing museums with a scalable, cost-efficient Content Management System (CMS) where they can create mobile apps for their visitors by themselves. The CMS empowers museum staff to create, manage, and update exhibit content independently. The visitor app offers an interactive experience enriched with multimedia features.

1.2 Goals and Stakeholders

Goals

This project aims to transform how museums interact with their visitors by providing a scalable and user-friendly platform while fostering engagement through interactive and multimedia-rich experiences. We wanted to bring the museum owners a tool where they could be independent. Our system allows them to create and manage all of their content without relying on external resources. Below is a detailed breakdown of the specific goals and their broader implications

1. Empower Museum Staff to Independently Manage Exhibit Content

Museums traditionally rely on external developers or technical experts to update content for bespoke mobile applications, leading to delays, higher costs, and logistical inefficiencies. This project aims to eliminate these barriers by developing a **Content Management System (CMS)** that is intuitive, flexible, and accessible to non-technical staff.

Key features supporting this goal include:

- **Multimedia Content Creation:** Museum staff can create and update exhibits using text, audio, images, and video directly from their devices.
- **User Roles and Permissions:** Curators and administrators have distinct permissions, ensuring secure and streamlined content management.
- **On-the-Go Accessibility:** Staff can manage content from mobile or desktop devices, enabling real-time updates during tours or exhibit changes.

2. Create a Scalable, Cost-Effective Platform

Scalability and affordability are critical for museums of varying sizes, especially smaller institutions that often operate with limited budgets. The platform's architecture supports:

- **Modularity:** Museums can expand their use of the system by adding more exhibits or tours with minimal adjustments.
- **Flexible Licensing:** The system adopts a pay-for-a-package model where the user can choose his preferred price and terms.

3. Enhance Visitor Engagement Through Interactive and Personalized Tours

Modern audiences, particularly younger generations, demand immersive and interactive experiences. Static displays or generic audio guides no longer suffice. To address this, the project incorporates:

- **Dynamic Multimedia Content:** Visitors can access exhibits enriched with high-quality images, video demonstrations, and narrated audio descriptions.
- **Cross-Platform Accessibility:** The app functions seamlessly on iOS, Android, and computer screens, ensuring inclusivity across devices.
- **Engagement Metrics:** Tools to measure visitor interaction, providing museums with valuable insights to refine their exhibits.
- **Interrogation of AI Tools:** On the visitor side, all descriptions can be read aloud using “Text to Speech” for a more comprehensive experience.

Stakeholders

Museum Staff

This group includes administrators, curators, and museum owners who play key roles in creating and maintaining the museum's content. Each role is addressed uniquely in the system, ensuring that their specific needs and responsibilities are met effectively and efficiently.

Secondary Audience: Visitors

Visitors represent the end users who interact with the **system's output**, which is the mobile application. This app serves as their gateway to a modern and engaging museum experience.

Role: Explore museum exhibits using the app to understand the collections better. Visitors have diverse needs, ranging from educational engagement to simple enjoyment of multimedia content.

1.3 Integration of AI Tools

AI technology is a cornerstone of the system, enhancing both efficiency and user experience for museum staff and visitors. The AI features are designed to streamline content creation and management processes while ensuring high-quality, engaging outputs. Key AI tools integrated into the system include:

1. Automated Text and Audio Generation

- **Text Generation:** AI algorithms assist in creating detailed, engaging descriptions for exhibits based on minimal input from curators.
- **Audio Narrations:** Generate high-quality audio guides directly from text, saving time and resources while ensuring consistency in tone and delivery.

2. Creating Descriptions Based on a Photo and Title

- Curators can upload an image and provide a title, and the AI tool generates a comprehensive and contextual description of the exhibit.
- This feature leverages advanced image recognition and natural language processing to deliver accurate, visually aligned descriptions.

3. Improving Existing Descriptions

- AI analyzes existing exhibit descriptions for clarity, engagement, and completeness.
- Suggests edits or enhancements to improve the overall quality, making the content more captivating for visitors.

2. Solution Description

The "Interactive Museum Guide and Content Creation System" is designed with a focus on scalability, efficiency, and engagement, catering to both museum staff and visitors. It

comprises three primary components, each tailored to address specific functional and operational needs: the Content Management System (CMS), the Visitor Mobile Application, and the Backend Infrastructure (see Figure 1). These components work in tandem to deliver a seamless experience for all users.

The system architecture follows a modular and layered design to ensure flexibility, scalability, and ease of maintenance. The core components include:

2.1 Content Management System (CMS)

The CMS is the system's backbone, designed for museum staff to manage content effectively. It features:

- **User-Friendly Interface:** Built with React.js, the CMS is responsive and intuitive, allowing users to navigate and manage content easily, even without technical expertise.
- **Multimedia Integration:** Curators can upload and manage text, images, videos, and audio files for exhibits, ensuring that visitors have access to rich multimedia content.
- **Role-Based Access:** Administrators can assign specific roles (e.g., curators, technical staff) to manage permissions and maintain security.
- **AI-Powered Features:**
 - Automated text and audio generation.
 - Descriptions are generated from photos and titles.
 - Tools for improving and editing existing content.
- **Analytics and Reporting:** Provides insights into visitor engagement, helping museum staff understand which exhibits resonate most with their audience.

2.2 Visitor Mobile Application

The mobile app is the primary interface for visitors to interact with the museum's content. Key features include:

- **Cross-Platform Compatibility:** Built with React Native, the app is compatible with both iOS and Android devices, ensuring accessibility for a wide audience.
- **Interactive Features:**
 - Personalized tour suggestions based on visitor preferences or popular exhibits.
 - Multimedia presentations, including high-resolution images and audio narrations.

2.3 Backend Infrastructure

The backend ensures the system operates efficiently, handling data processing, content storage, and real-time updates.

- **Server:** Powered by Node.js with Express.js, the backend manages API requests, user authentication, and data processing.
- **Database:** MongoDB handles unstructured data, such as multimedia content, while Firebase ensures encrypted authentication.
- **Cloud Hosting:** Deployed on Vercel for reliability, scalability, and secure data management.

Overall Architecture:

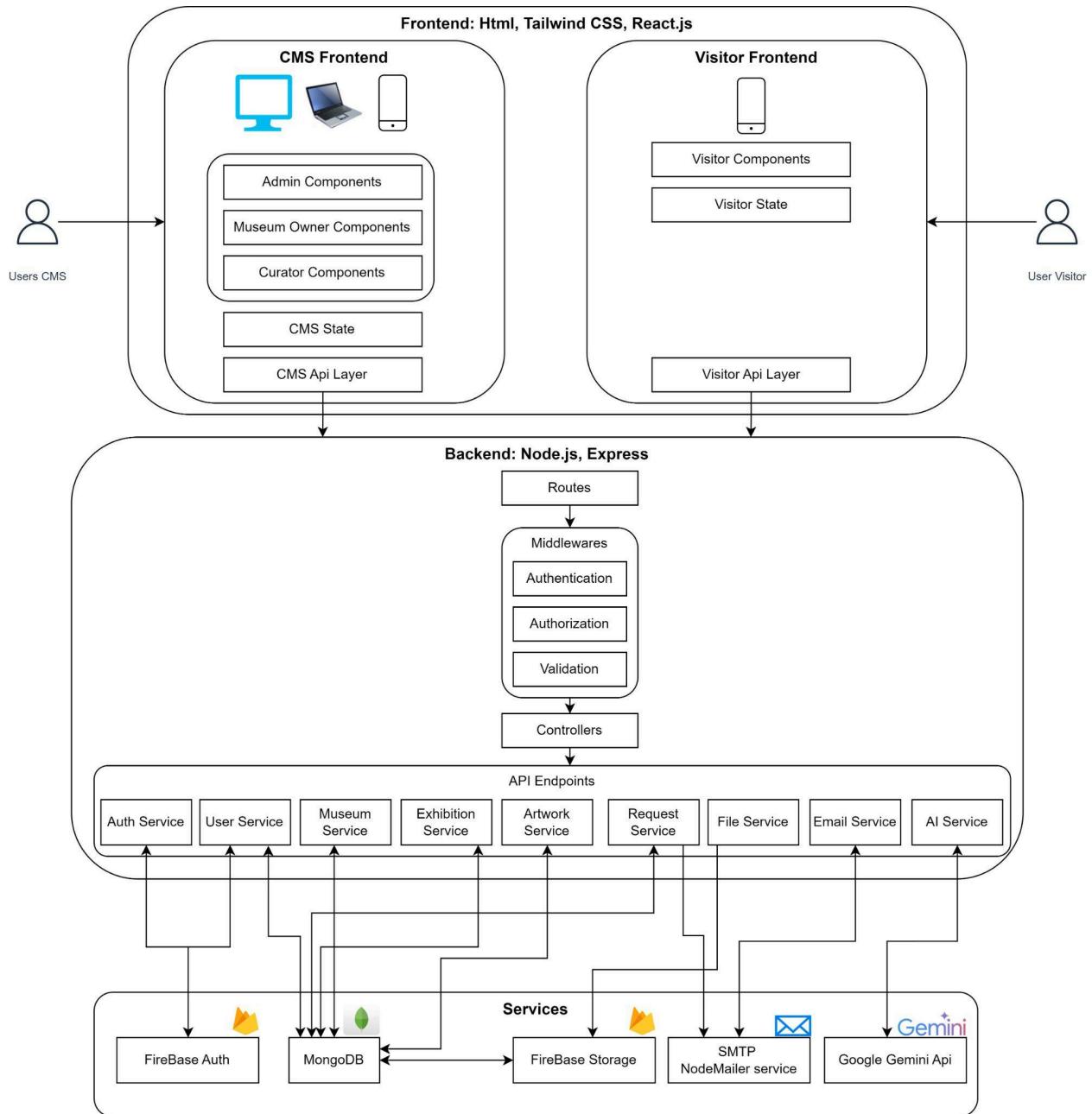


Figure 1: overall system architecture with all the technologies, tools and services.

3. Research and Development Process

The research and development process for the "Interactive Museum Guide and Content Creation System" was designed to address both the immediate requirements of museum operations and the evolving expectations of visitors. This phase included an in-depth exploration of user needs, other options available in the market, technological opportunities, and iterative development practices (see Figure 2).

Flow chart of Research and Development process

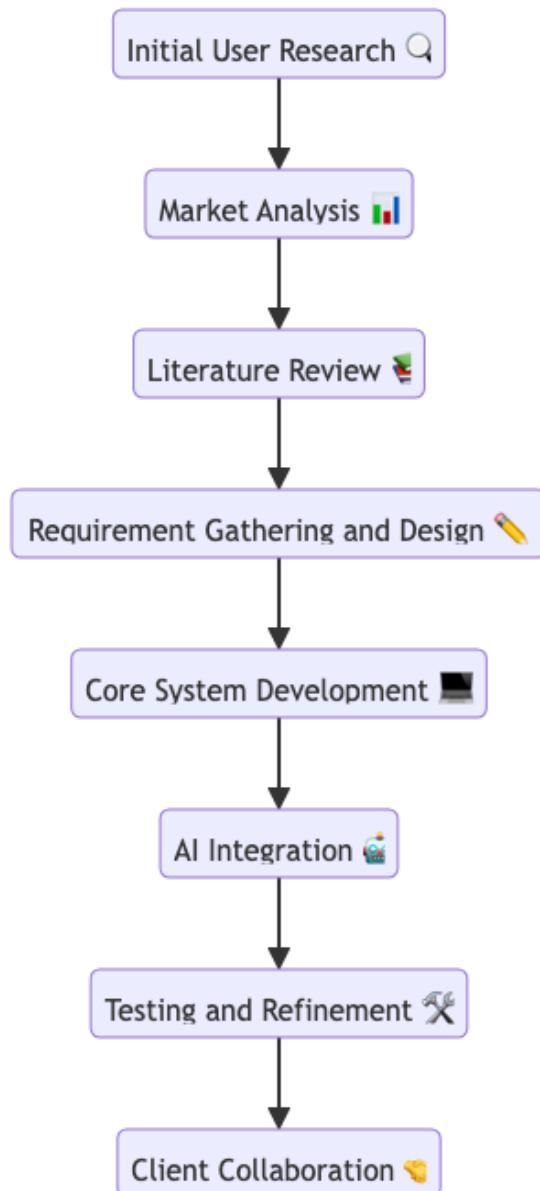


Figure 2: illustration of the Research and Development process

3.1 Research Process

3.1.1 Initial User Research

The system's foundation was laid through comprehensive interviews and discussions with Menachem¹, our customer, and the system's admin. These sessions provided insights into the limitations of existing solutions and the unique needs of museum staff and visitors. Our Key findings included:

- There is a lack of user-friendly tools for museum staff to manage exhibit content independently.
- High dependency on external developers for app updates and maintenance.
- There is a growing demand for engaging, multimedia-rich content from visitors, particularly younger audiences.

Meachem wanted us to provide him with a system he can sell to museum owners where he will be defined as the root user. Meachem can oversee all aspects of the system. He can modify terms and pricing and close/open museums.

3.1.2 Market Analysis

To understand the competitive landscape, a thorough market analysis was conducted, focusing on:

- **Existing Solutions:** Platforms like Omeka² and WordPress³ were evaluated. While they provided general content management functionalities, they lacked features tailored to museum operations, such as multimedia management and real-time updates.
- **Technological Gaps:** Identified scalability, cost-efficiency, and integration challenges with emerging technologies like AI.
- **User Preferences:** Examined feedback and user reviews from existing museum apps to identify common pain points, such as outdated interfaces, limited accessibility features, and slow loading times.

3.1.3 Literature Review

We reviewed academic and industry publications on museum technology, visitor engagement, and digital transformation. Insights from studies on user behavior and the impact of multimedia content on learning experiences were incorporated into the system

¹ [Menachem's Website](#)

² [Omeka](#)

³ [WordPress](#)

design. For more details, you can read the book Phase A.

3.1.2 Requirement Gathering and Design

The project began with a detailed requirements analysis to ensure alignment with user needs and technical feasibility. This phase included:

- **User Stories and Scenarios:** Developed based on interviews and observations to capture real-world workflows and expectations.
- **Functional Requirements:** Defined features such as multimedia content creation, AI-assisted description generation, and QR code-based navigation.
- **Non-Functional Requirements:** Addressed aspects like scalability, security, performance, and accessibility.

The system architecture was carefully crafted to support scalability, modularity, and ease of use. Key steps in this phase included:

- **Use-Case Diagrams:** Developed to outline workflows for administrators, curators, and visitors, ensuring all user interactions were accounted for. (See figure 3)
- **Technical Stack Selection:** React.js and React Native were chosen for the frontend, ensuring cross-platform compatibility. Node.js and MongoDB were selected for the backend due to their scalability and performance.

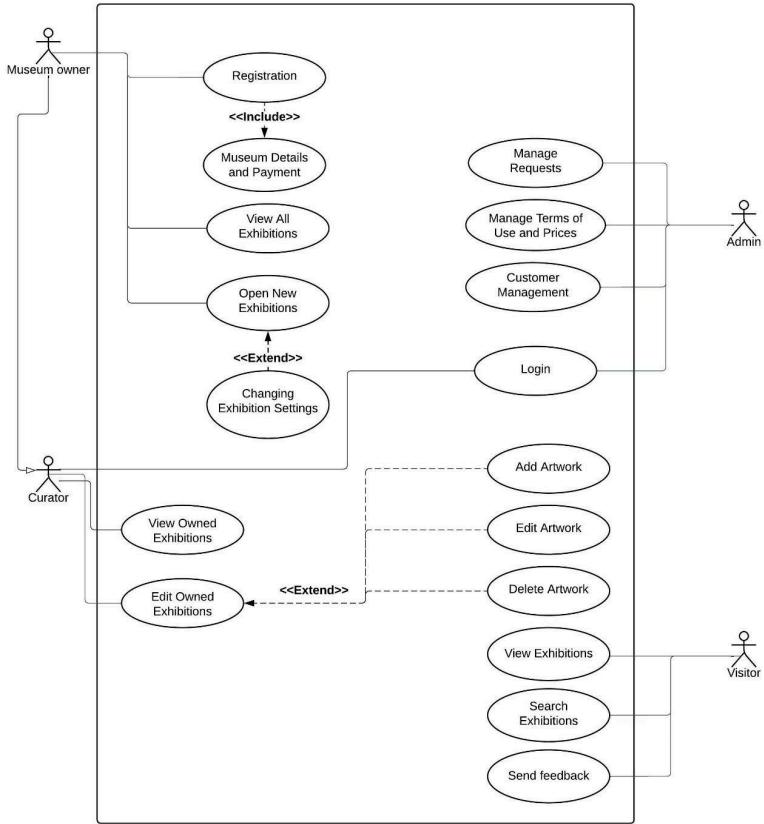


Figure 3: use case diagram showcasing different actions and users in the system

3.2.2 Development Phases

The development process of the Interactive Museum Guide and Content Creation System was structured into three iterative phases to ensure a clear, methodical, and feedback-driven approach. Each phase played a critical role in the evolution of the project, enabling continuous improvement and integration of essential features.

Phase 1: Core System Development

The primary objective of this phase was to establish a robust foundation for the system, focusing on developing essential backend and frontend components. The following key activities were undertaken:

- 1. Content Management System (CMS) Backend:**
 - **Implementation of CMS Functionalities:** Developing a sophisticated CMS that allows for the seamless creation, management, and organization of museum content. This system is accessible to multiple user types, including Admins, Museum Owners, and Curators.

- **User Roles and Permissions:** A multi-user system was established, enabling specific roles and access levels. Admins have comprehensive control, while Museum Owners and Curators have permissions that match their content creation and management needs.
- **Dashboard Creation:** Comprehensive dashboards were developed for each user type, providing them with easy access to key functionalities. For instance, an Admin can view system-wide statistics, while a Museum Owner can track exhibit performance and user engagement.

2. Visitor App Development:

- **User-Centric Design:** Focus was placed on creating a simple and intuitive user interface (UI) to ensure smooth navigation for museum visitors.
- **QR Code Scanning:** A core feature introduced was the integration of QR code scanning capabilities. Visitors can scan QR codes placed at the museums to access the app instantly.

Phase 2: AI Integration

In this phase, advanced AI capabilities were embedded into the system to enhance content creation, user engagement, and the overall visitor experience.

1. AI-Powered Content Creation:

- **Automated Text and Audio Generation:** Exhibit descriptions were automatically generated using AI models. This feature enabled the system to create text descriptions and audio tours from exhibit images and titles.
- **Image Recognition for Description Creation:** The system integrates AI-powered image recognition tools. When an image of an exhibit is uploaded, the AI can analyze it, recognize the object, and provide a relevant, context-rich description.

2. AI Enhancement of Existing Content:

- **Description Improvement:** AI tools were used to refine and improve existing exhibit descriptions. By analyzing sentence structure, grammar, and contextual accuracy, the system provides suggestions for curators to enhance their content.

3. Visitor Engagement Through AI:

- **Content Accessibility Features:** Accessibility features such as text-to-speech and image alt-text generation were introduced, enabling users with disabilities to engage with the exhibits.

Phase 3: Testing and Refinement

The third and final phase focused on extensive testing, optimization, and system enhancements to ensure reliability, user satisfaction, and cross-platform compatibility.

1. Comprehensive Testing:

- **Cross-Platform Compatibility:** The system was tested on various devices and operating systems, including iOS, Android, and desktop web browsers.
- **Bug Identification and Fixes:** We conducted a rigorous quality assurance process to identify and resolve technical issues, including software bugs, UI glitches, and inconsistencies in user flows.

2. User-Centered Refinement:

- **User Feedback Loop:** Feedback from Menachem and his co-workers was gathered. Based on this feedback, design changes, usability improvements, and feature enhancements were implemented. As part of this process, we also sent Menachem and his co-workers a System Usability Scale (SUS) interview to fill out getting a score of 90, which provided further insights into user satisfaction and areas for improvement.
- **Training and Onboarding:** Menachem received tutorials and guidance on how to use the CMS and Visitor App. User guides and support documentation were developed to ensure a smooth onboarding process for all users.

These three development phases ensured a systematic, feedback-driven process that enabled the successful creation of a comprehensive, interactive museum guide and content creation system.

3.2.3 Client Collaboration

Regular meetings with Menachem were held to align on progress, address challenges, and incorporate feedback. Visual prototypes and demos were used to communicate updates effectively. Some of Menachem's comments included adding record options and not only Text to Speech functionality, all kinds of design preferences and making sure the system is responsive for all types of screens. We made sure all of his requests were implemented.

3.2.4 Tools and Technologies

Frontend:

- Developed using **HTML, Tailwind CSS, and React.js** for the Content

Management System (CMS) and for the Visitor Mobile App.

- Provides a user-friendly interface for museum staff to create and manage content and for visitors to interact with exhibits.
- Features cross-platform compatibility, ensuring accessibility on desktops, smartphones, and tablets.

Backend:

- **Database and Storage (MongoDB and Firebase):**
 - MongoDB manages structured and unstructured data, such as exhibit metadata, user information, and content logs.
 - Firebase Firestore manages storage, enabling seamless data synchronization and real-time updates across web and mobile applications.
- **Authentication and Authorization:** Firebase Authentication handles secure user authentication, while private middlewares manage authorization to protect access to restricted resources.

API Server:

- Acts as the intermediary between the frontend and backend.
- Built using **Node.js** and **Express.js**, it handles requests and responses efficiently.
- Processes user data, handles CRUD (Create, Read, Update, Delete) operations, and manages API requests for content updates and visitor interactions.

AI Integration:

- Implemented AI algorithms using libraries and APIs for text generation, translation, and media optimization. We used Google's API for Text to Speech and Speech to Text and Gemini's API for AI prompts creation.

4. Challenges and Solutions

The development and implementation of the system presented various challenges, ranging from technical complexities to user experience considerations. Each challenge required a solution to ensure the system's robustness, scalability, and usability.

Cross-Platform Compatibility

One of the primary challenges was ensuring the system's seamless functionality across multiple platforms. The museum staff accessed the CMS through desktop devices, while

visitors interacted with the mobile app on iOS and Android. These platform differences created inconsistencies due to variations in rendering engines and operating systems. This was particularly apparent when implementing QR code scanning and multimedia playback features.

To solve this, we selected React.js for the CMS and React Native for the mobile app. These technologies provided a unified development framework, reducing platform-specific inconsistencies. Rigorous testing was conducted using tools like BrowserStack and real devices to uncover and resolve platform-specific issues. Additionally, the team incorporated responsive design principles using flexible layouts and media queries to ensure a consistent experience across various screen sizes.

User Experience Challenges

Another significant challenge was designing an intuitive and engaging interface for two distinct user groups—museum staff and visitors. Museum staff required a CMS that balanced professional-grade tools with simplicity, while visitors expected an app that was visually appealing, easy to navigate, and interactive.

The team tackled this by adopting an iterative design approach. During the development process, wireframes and prototypes were shared with museum staff and visitors to gather feedback and make adjustments. For museum staff, step-by-step onboarding tutorials were added to the CMS to simplify the learning curve. For visitors, the app's interface was designed to focus on clarity, incorporating features like QR code scanning and visually rich multimedia displays. The CMS also included a feedback mechanism to allow staff to report issues or suggest improvements.

AI Integration

Incorporating AI tools to automate text and audio generation, improve exhibit descriptions, and create descriptions based on photos was a complex but essential feature. Ensuring AI-generated outputs were accurate, contextually relevant, and aligned with the museum's tone posed significant challenges.

To tackle this, the team leveraged pre-trained AI models like OpenAI's APIs for text and audio generation. These models were fine-tuned to understand museum-specific terminology and contexts, resulting in more accurate outputs. A feedback loop was also built into the system, allowing curators to review and refine AI-generated content. This iterative process ensured high-quality results while maintaining curator oversight.

Results and Conclusions

The "Interactive Museum Guide and Content Creation System" successfully achieved its primary objectives, addressing the needs of museum staff and visitors while overcoming the challenges encountered during development. The project results demonstrate significant progress in enhancing museum operations and visitor engagement. Below is a detailed analysis of the results and conclusions drawn from the project.

5. Results

5.1 Functional Success

The system was fully operational by the end of the development phase, meeting all functional requirements:

- The **CMS** provided museum staff with a user-friendly platform to create, edit, and manage exhibit content independently. Features such as multimedia integration, role-based permissions, and AI tools (for text generation and description improvements) were implemented successfully.
- The **Visitor Mobile App** delivered a seamless experience, allowing users to access exhibits via QR codes, browse personalized tours, and interact with multimedia content. The app was compatible with both iOS and Android devices and performed well under testing conditions.

5.2 Feedback and Expectations

The feedback gathered from stakeholders, including Menachem, highlighted several points of excellence and opportunities for further development. From the satisfaction surveys, specific areas were commended, such as:

- **Ease of Use:** The system's user-friendly design was recognized as a highlight, with particular appreciation for the intuitive CMS structure, which facilitates smooth content creation and management.
- **Future Potential:** Menachem noted, "You went above and beyond what was expected. You took this project to places I had not even thought of and opened my mind to more future ideas." This feedback reflects the innovative approach and the project's alignment with the evolving needs of cultural institutions.
- **Preservation Recommendations:** The friendly CMS structure was specifically suggested for preservation, emphasizing its potential as a robust tool for museums.

Such endorsements underscore the project's success in meeting and exceeding expectations, inspiring confidence in its scalability and future impact on the museum and cultural sectors.

5.3 Real-Time Updates

Real-time synchronization between the CMS and the visitor app worked flawlessly, ensuring that any changes made by museum staff were immediately reflected in the app. This feature eliminated delays and allowed museum staff to adapt content dynamically based on visitor feedback or changing exhibit requirements.

5.4 Performance Improvements

Performance optimization techniques, including image compression, caching, and lazy loading, resulted in a fast and responsive system. Testing demonstrated:

- A **reduction in load times** for multimedia content.
- Smooth operation across various devices, even in environments with limited internet connectivity.

5.5 AI Integration

One of Menachem's requests was to interrogate AI tools in the systems. The AI tools introduced in the system yielded significant results:

- Automated text and audio generation provided high-quality, contextual exhibit descriptions, reducing the workload for curators.
- AI-powered improvements to existing descriptions enhanced clarity and engagement, receiving positive feedback during user testing.
- Description generation based on photos and titles was particularly beneficial for rapidly digitizing new exhibits.

5.6 Technology Stack Selection

Considerations:

Choosing technologies for the frontend, backend, and database was a foundational decision. It was essential to select tools and frameworks that could deliver cross-platform compatibility, scalability, and ease of development while minimizing the learning curve for the team.

Reasoning:

- **React.js and React Native:** These were chosen for their ability to support both web-based CMS and mobile app development using a single codebase. This significantly reduced development time and ensured consistency across platforms.
- **Node.js and Express.js:** Node.js was selected for its scalability and efficiency in handling asynchronous operations, making it ideal for a system requiring real-time updates and high concurrency.
- **MongoDB:** MongoDB's flexibility in handling unstructured data (such as multimedia content) and its scalability made it an obvious choice for storing exhibit metadata, user profiles, and other content.
- **Firebase:** Firebase was authentication; its easy integration with JavaScript-based frameworks further solidified this decision.

User-Centered Design Approach

Considerations:

Designing an intuitive and accessible system for museum staff and visitors required a focus on usability and user experience (UX). The target audience included non-technical museum staff and visitors with varying levels of tech proficiency, from children to elderly individuals.

6. Lessons Learned

The development and implementation of the "Interactive Museum Guide and Content Creation System" offered valuable insights into software engineering, user experience design, and project management. By reflecting on the successes and challenges faced during this project, several lessons emerge that can guide future endeavors.

One significant takeaway was the importance of starting with a **user-centered design approach**. Early involvement of museum staff and visitors helped align the system's design with real-world needs. For instance, curators and administrators appreciated the simplicity of the CMS's drag-and-drop interface, which was developed based on their feedback. However, involving a more diverse set of users, such as non-technical staff or visitors with disabilities, from the beginning could have ensured a more inclusive design. User testing highlighted issues like overly complex navigation for the mobile app, which were simplified in later iterations, but addressing these earlier would have saved time and resources.

The decision to prioritize **scalability and modularity** proved crucial for the system's success. AWS provided reliable and scalable infrastructure, while the modular architecture allowed for isolated development and testing of components like the CMS, backend, and mobile app. This approach facilitated smoother debugging and future feature additions. However, the initial onboarding process for smaller museums could have been streamlined further, as their limited technical resources required additional guidance. Future projects could include a lightweight "starter package" tailored for smaller organizations to minimize onboarding challenges.

Integrating AI tools was a significant innovation and a valuable learning experience. Automated text and audio generation features significantly reduced curators' workloads, while AI-powered description improvements enhanced content quality. The AI tools, however, required ongoing fine-tuning to ensure the relevance and accuracy of generated content. One lesson learned was the importance of domain-specific training for AI models, as pre-trained models occasionally lacked the contextual depth needed for museum-specific content. A larger dataset of museum artifacts and descriptions could have further improved AI performance and reduced the need for human oversight.

Collaboration within the development team presented its own challenges, particularly during the early stages of the project. Distributed work led to occasional misaligned priorities and code conflicts, which delayed progress. These challenges were mitigated through Agile practices, such as daily standups and sprint reviews, as well as the adoption of a clear Git branching strategy and peer code reviews. Implementing these measures earlier in the project would have prevented initial delays. In addition, using collaborative tools like task management software and shared documentation platforms helped streamline communication and align team efforts.

Real-time synchronization between the CMS and the visitor app was both a technical success and a learning opportunity. Firebase enabled instant updates, ensuring visitors always accessed the latest content. However, stress testing revealed that synchronization could falter under heavy traffic, particularly when handling large volumes of multimedia updates. Conducting stress tests earlier in the development cycle would have allowed the team to optimize performance proactively and avoid last-minute fixes.

7. Project Benchmarks

The project successfully met, and in several areas exceeded, the benchmarks set at the beginning of the development phase. These benchmarks, designed to measure functionality, performance, scalability, and user experience, were guiding metrics to ensure the project's success.

The system's core functionality, including the CMS and visitor app, was fully realized. Museum staff gained the ability to independently create and manage multimedia-rich content while visitors engaged with interactive features like QR code scanning and personalized tours. Both components worked seamlessly, delivering on the promise of empowering museum operations and enhancing visitor engagement.

Meeting Menachem's expectations and requests was a vital benchmark for the project. His insights and priorities shaped the system's development, ensuring it aligned with the practical needs of museum operations while integrating innovative features to enhance both staff efficiency and visitor engagement. By addressing each of his requirements, the project delivered a tool that is not only technically sound but also highly functional and user-friendly, positioning it as a transformative solution for modern museums.

8. User Guide

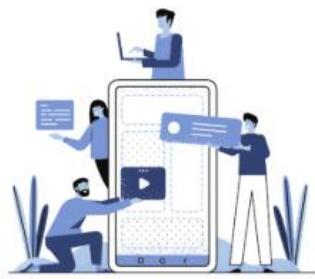
Overview

The registration process for the museum app is designed to be intuitive, streamlined, and fully responsive. Whether you're using a desktop, tablet, or mobile device, the app adapts seamlessly to provide a smooth user experience.

Welcome Screen

- **Purpose:** Introduces users to the app and its capabilities.
- **How to Use:** Read the introductory message about creating your virtual museum. Click “**Let’s Start**” to proceed, or use the “**Already a User? Click Here**” link to log in if you have an account.

Phone view:



CREATE AN APP FOR YOUR MUSEUM

Transform your museum with our Interactive Guide! Manage exhibits and engage visitors using our user-friendly app.

• • •



EDIT, UPLOAD AND RECORD

Edit, upload, and record content effortlessly. Capture audio, video, and photos on your mobile device to manage exhibits and engage visitors.

• • •



CREATE YOUR VIRTUAL MUSEUM

Get ready to embark on a journey! Prepare to create, upload, and manage content seamlessly as you bring your museum to life.

• • •

Next

Previous

[Already a user? Click here](#)

Next

Previous

[Already a user? Click here](#)

Let's Start

[Already a user? Click here](#)

Sign up screens:

Purpose: Allows users to create an account to access the app.

How to Use:

- Enter your personal information: Name, Email, Password, and Phone Number.
- Use the **eye icon** to toggle password visibility.
- Tap “**Next**” to proceed to the museum registration

Museum Owner Sign Up

Name

Last Name

Email

Password

Confirm Password

Phone Number

Museum Registration

Museum Name

Address

City

State

Zip Code

Phone Number

Email

Choose your plan

Name: Free try plan
Price: \$0
Features: Free try
Exhibitions: Up to 1 exhibition
Artworks: Manage up to 5 artworks

Name: Introductory Plan
Price: \$50
Features: Basic editing and uploading
Exhibitions: Up to 5 exhibitions
Artworks: Manage up to 100 artworks

Name: Basic Plan
Price: \$100
Features: Editing and uploading
Exhibitions: Up to 10 exhibitions
Artworks: Manage up to 150 artworks

Payment

Card Number

Expiry Date

CVV

Email

I agree to the [terms and conditions](#)

Previous
Next
Previous
Next
Previous
Next



User Information

Personal Details

Name: test
Last Name: test
Email: test@gmail.com
Phone Number: 0521234567

Museum Details

Museum Name: Museum Test
Address: test
City: test
State: test
Zip Code: 1234
Phone Number: 0521234567
Email: test@gmail.com

Payment Details

Plan: Basic Plan
Price: \$100
Card Number: ****1234
Expiry Date: 12/24
CVV: **1

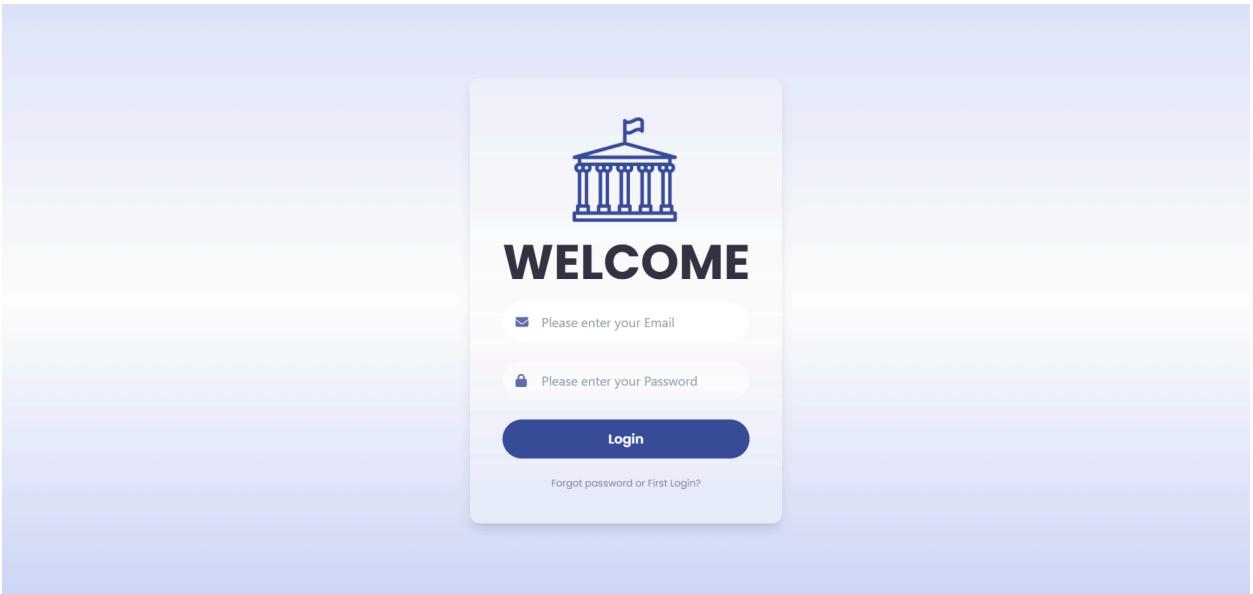
Previous

Confirm & Pay

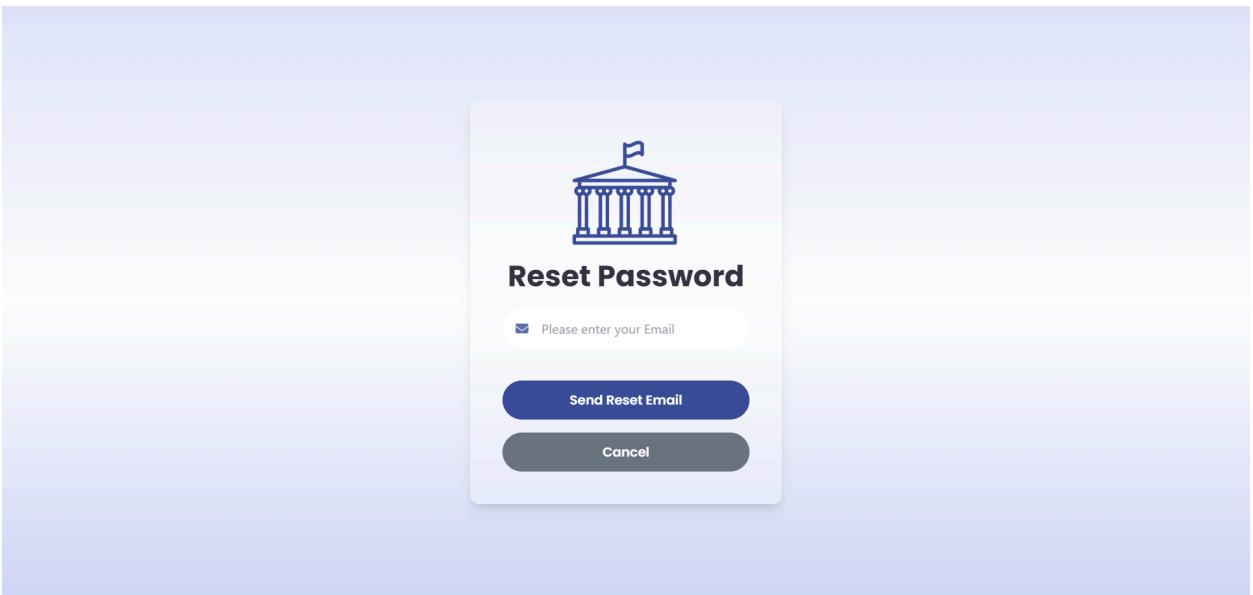
Terms and Conditions

Terms Accepted: Yes

Login Screens:



Reset password:



Admin screens:

Admin Dashboard

Purpose: Displays key statistics on exhibitions and user activity for system monitoring.

Features:

1. Exhibitions by Museum Status: Bar chart showing the number of open and closed exhibitions.
2. User Registrations Over Time: Line graph showing user registration trends.
3. Exhibitions Over Time: Tracks the total number of exhibitions created over time.



Admin Sidebar

Purpose: The sidebar provides quick navigation to key sections of the admin panel.

Sections:

1. **Admin Dashboard:** View overall system statistics.
2. **Watch All:** Monitor all activity within the app.
3. **Museums:** Manage registered museums.
4. **Exhibitions:** Manage exhibitions created by museums.
5. **Customers:** View and manage customer profiles.
6. **Requests Management:** Handle pending and approved requests.
7. **Terms, Prices, and Packages:** Edit app terms, pricing, and available packages.
8. **Language:** Switch language to Hebrew/English
9. **Dark Mode:** Toggle dark mode / light mode for the interface.
10. **Logout:** Log out of the admin panel.



Main Page:

Purpose: Displays top visited museums and exhibitions.

Sections:

1. **Most Visited Museums:** Shows museum name, location, exhibitions, and artworks count.
2. **Most Visited Exhibitions:** Lists exhibition name, description, artworks, curators, and status.

The most visited museums

**Garbuz Games****Location:** Lilach 2/50, Kiryat Bialik, Israel**Exhibitions:** 6**Artworks:** 21**test****test****Location:** test, test, test**Exhibitions:** 0**Artworks:** 0

The most visited exhibitions

**Game of Thrones**

Immerse yourself in the world of Westeros. Explore the artistry, history, and legacy of HBO's Game of Thrones.

Artworks: 10
Curators: 1
Status: Open**Board games and strategy**

Journey through time and across cultures as you explore a world of classic board games. Discover rare and unique versions of chess, G...

Artworks: 4
Curators: 2
Status: Open**Harry Potter**

Journey into the Wizarding World. Discover authentic props and costumes from the Harry Potter films.

Artworks: 5
Curators: 2
Status: Open**Test**

test

Artworks: 0
Curators: 0
Status: Closed

mensch-edutainment © 2024 All rights reserved.

Made by Shay Garbuz & May Caspi

Museums Management

Purpose: Allows admins to view and manage museums registered in the system.

Features:

1. Filters:

- **Show All:** Displays all registered museums.
- **Show Opened Museums:** Displays only museums with an "open" status.
- **Show Closed Museums:** Displays only museums with a "closed" status.

2. Table Information:

- **Name:** Name of the museum.
- **Museum Owner:** Owner of the museum account.
- **Location:** Address or city of the museum.
- **Plan:** The subscription or service plan chosen by the museum.
- **Created At:** The date the museum was registered.
- **Status:** Current status of the museum (open or closed).
- **Actions:** Option to **View and Edit** museum details.

The screenshot shows a user interface for managing museums. At the top right, it says "System Admin". Below that is a navigation bar with three buttons: "Show All" (highlighted in blue), "Show Opened Museums", and "Show Closed Museums". The main area is titled "Museums" and contains a table with the following data:

| Name | Museum Owner | Location | Plan | Created At | Status | Actions |
|--------------|--------------|------------------------------------|-------------------|------------|--------|-------------------------------|
| Garbuz Games | Shay Garbuz | Lilach 2/50, Kiryat Bialik, Israel | Introductory Plan | 11.11.2024 | open | View and Edit |
| test | test test | test, test, test | Introductory Plan | 23.12.2024 | open | View and Edit |

At the bottom of the page, there is a small footer note: "mench-edutainment © 2024 All rights reserved. Made by Shay Garbuz & May Caspi".

Purpose: Allows admins to update museum details.

☰ System Admin

Edit Museum

[Visit the museum](#)

| | |
|---------|--|
| Name | Garbz Games |
| Address | Lilach 2/50 |
| City | Kiryat Bialik |
| Plan | Introductory Plan (Up to 5 exhibitions, Manage up to 100 artworks) |
| Status | Open |

Save Changes

mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbz & May Caspi

Exhibitions Management

Purpose: View and manage exhibitions.

Features:

- Filters:** Show All, Open, or Closed exhibitions.
- Table:** Displays exhibit name, museum, curators, status, creation date, and actions (**View and Edit**).

☰ System Admin

Exhibitions

| Exhibit Name | Museum | Curators | Status | Created at | Actions |
|--------------------------|-------------|--------------------|--------|------------|-------------------------------|
| Board games and strategy | Garbz Games | curator1, curator2 | open | 11.11.2024 | View and Edit |
| Classic video games | Garbz Games | curator1 | open | 11.11.2024 | View and Edit |
| Game of Thrones | Garbz Games | curator1 | open | 13.11.2024 | View and Edit |
| The food exhibition | Garbz Games | curator1 | open | 14.11.2024 | View and Edit |
| Test | Garbz Games | | closed | 14.11.2024 | View and Edit |
| Harry Potter | Garbz Games | curator1, curator2 | open | 14.11.2024 | View and Edit |

mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbz & May Caspi

Edit Exhibit

Purpose: Allows admins to update exhibit details.

Fields:

- **Exhibit Name:** Edit the name of the exhibit.
- **Description:** Update the exhibit's description.
- **Max Artworks:** Set the maximum number of artworks.
- **Curators:** Assign or modify curators.
- **Museum:** Select the associated museum.
- **Status:** Change exhibit status (Open/Closed).

Actions:

- **Save Changes:** Apply updates.
- **Back:** Return without saving.

The screenshot shows a user interface for editing an exhibit. At the top right, it says "System Admin". The main title is "Edit Exhibit" with a link "Click here to view the Artworks".
The form fields are:

- Exhibit Name:** The food exhibition
- Description:** A journey through the history and culture of food. Explore its impact on society and our lives.
- Max Artworks:** 7
- Curators:** curator1 s1
- Museum:** Garbz Games
- Status:** Open

At the bottom, there are "Back" and "Save Changes" buttons. The "Save Changes" button is highlighted in blue. A footer note at the bottom right reads: "menach-edutainment © 2024 All rights reserved.
Made by Shay Garbz & May Caspi".

Users Management

Purpose: Manage museum owners and curators.

Sections:

1. Museum Owners:

- Displays name, associated museum, email, and phone number.

2. Curators:

- Displays name, email, phone number, and **Remove** action to delete a curator.

System Admin

Museum Owners

| Name | Museum Name | Email | Phone Number |
|-------------|--------------|-----------------------|---------------|
| Shay Garbuz | Garbuz Games | Shay.garbuz@gmail.com | +972522835144 |
| test test | test | test@gmail.com | +972531234567 |

Curators

| Name | Email | Phone Number | Actions |
|-------------|--------------------|---------------|-------------------------|
| curator1 s1 | curator1@gmail.com | +972521111111 | <button>Remove</button> |
| curator2 s2 | curator2@gmail.com | +972521111112 | <button>Remove</button> |

mersch-edutainment © 2024 All rights reserved.
Made by Shay Garbuz & May Capri

Prices and Packages

Purpose: Manage and customize subscription plans for museums.

Features:

- Add Plan:** Create a new plan with specific limits on exhibitions, artworks, and features.
- Plan Table:** Shows existing plans with details like maximum exhibitions, artworks, price, and features. Use **Edit** to update any plan.

Manage Terms of Use:

Edit the terms and conditions that museums must agree to when using the app.

The screenshot shows a mobile application interface. At the top right, it says "System Admin". Below that is a header "Prices and Packages" with a green "Add Plan" button. A table lists five subscription plans with columns for Package, Exhibitions, Artworks, Features, Price, and Actions (Edit). The plans are: Free try plan (Up to 1 exhibitions, Up to 5 artworks, Free try, 0\$, Edit), Introductory Plan (Up to 5 exhibitions, Up to 100 artworks, Basic editing and uploading, 50\$, Edit), Basic Plan (Up to 10 exhibitions, Up to 150 artworks, Editing and uploading, 100\$, Edit), Premium Plan (Up to 50 exhibitions, Up to 1000 artworks, Editing, uploading, and exclusive content access, 300\$, Edit), and VIP Plan (Up to 100 exhibitions, Up to 2000 artworks, All Premium features + VIP support, 500\$, Edit). Below this is a "Manage Terms of Use" section with a blue "Open and Edit Terms" button. At the bottom, there's a footer with small text: "mensch-edutainment © 2024 All rights reserved. Made by Shay Garbuz & May Caspi".

| Package | Exhibitions | Artworks | Features | Price | Actions |
|-------------------|-----------------------|---------------------|--|-------|---------|
| Free try plan | Up to 1 exhibitions | Up to 5 artworks | Free try | 0\$ | Edit |
| Introductory Plan | Up to 5 exhibitions | Up to 100 artworks | Basic editing and uploading | 50\$ | Edit |
| Basic Plan | Up to 10 exhibitions | Up to 150 artworks | Editing and uploading | 100\$ | Edit |
| Premium Plan | Up to 50 exhibitions | Up to 1000 artworks | Editing, uploading, and exclusive content access | 300\$ | Edit |
| VIP Plan | Up to 100 exhibitions | Up to 2000 artworks | All Premium features + VIP support | 500\$ | Edit |

Add New Plan

Create subscription plans by entering **Name**, **Max Exhibitions**, **Max Artworks**, **Features**, and **Price**. Click **Add** to save or **Cancel** to discard.

Plan Table

View all plans with options to **Edit** details.

Manage Terms of Use

Click **Open and Edit Terms** to update terms.

System Admin

Prices and Packages

Add Plan

Name

MaxExhibitions

MaxArtWorks

Features

Price

Add Cancel

| Package | Exhibitions | Artworks | Features | Price | Actions |
|-------------------|-----------------------|---------------------|--|-------|---------|
| Free try plan | Up to 1 exhibitions | Up to 5 artworks | Free try | 0\$ | Edit |
| Introductory Plan | Up to 5 exhibitions | Up to 100 artworks | Basic editing and uploading | 50\$ | Edit |
| Basic Plan | Up to 10 exhibitions | Up to 150 artworks | Editing and uploading | 100\$ | Edit |
| Premium Plan | Up to 50 exhibitions | Up to 1000 artworks | Editing, uploading, and exclusive content access | 300\$ | Edit |
| VIP Plan | Up to 100 exhibitions | Up to 2000 artworks | All Premium features + VIP support | 500\$ | Edit |

Manage Terms of Use

Open and Edit Terms

mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbuz & May Caspi

System Admin

Price

Edit Terms and Conditions

| Package | | Actions |
|-------------------|--|---------|
| Free try plan | | Edit |
| Introductory Plan | | Edit |
| Basic Plan | | Edit |
| Premium Plan | | Edit |
| VIP Plan | | Edit |

This document outlines the official regulations and terms of use (hereafter referred to as "Terms") for all users accessing and using the services, features, and content provided by [Your Company Name] (hereafter referred to as "the Platform"). By accessing or using the Platform, you agree to be bound by these Terms.

1. Acceptance of Terms
By accessing or using the Platform, you affirm that you are legally capable of entering into binding contracts and agree to comply with these Terms. If you do not agree to these Terms, you must not use or access the Platform.

2. Account Registration and Security

CLOSE SAVE

Manage Terms of Use

Open and Edit Terms

mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbuz & May Caspi

Requests Management

Purpose: Manage and track requests from museums.

Sections:

1. Pending Requests:

- Displays requests awaiting admin action.
- Columns: Museum Name, Requestor, Type, Status, Created At, and Actions.

2. Answered Requests:

- Shows previously processed requests.
- Columns: Museum Name, Requestor, Type, Status, and Created At.

System Admin

Requests Management

Pending Requests

| Museum Name | Requestor | Type | Status | Created At | Actions |
|-------------|-----------|------|--------|------------|---------|
| | | | | | |

Answered Requests

| Museum Name | Requestor | Type | Status | Created At |
|-------------|-----------|----------------|----------|------------|
| test | test | Museum-Opening | Approved | 23.12.2024 |
| Garbuz | Shay | Museum-Opening | Approved | 11.11.2024 |

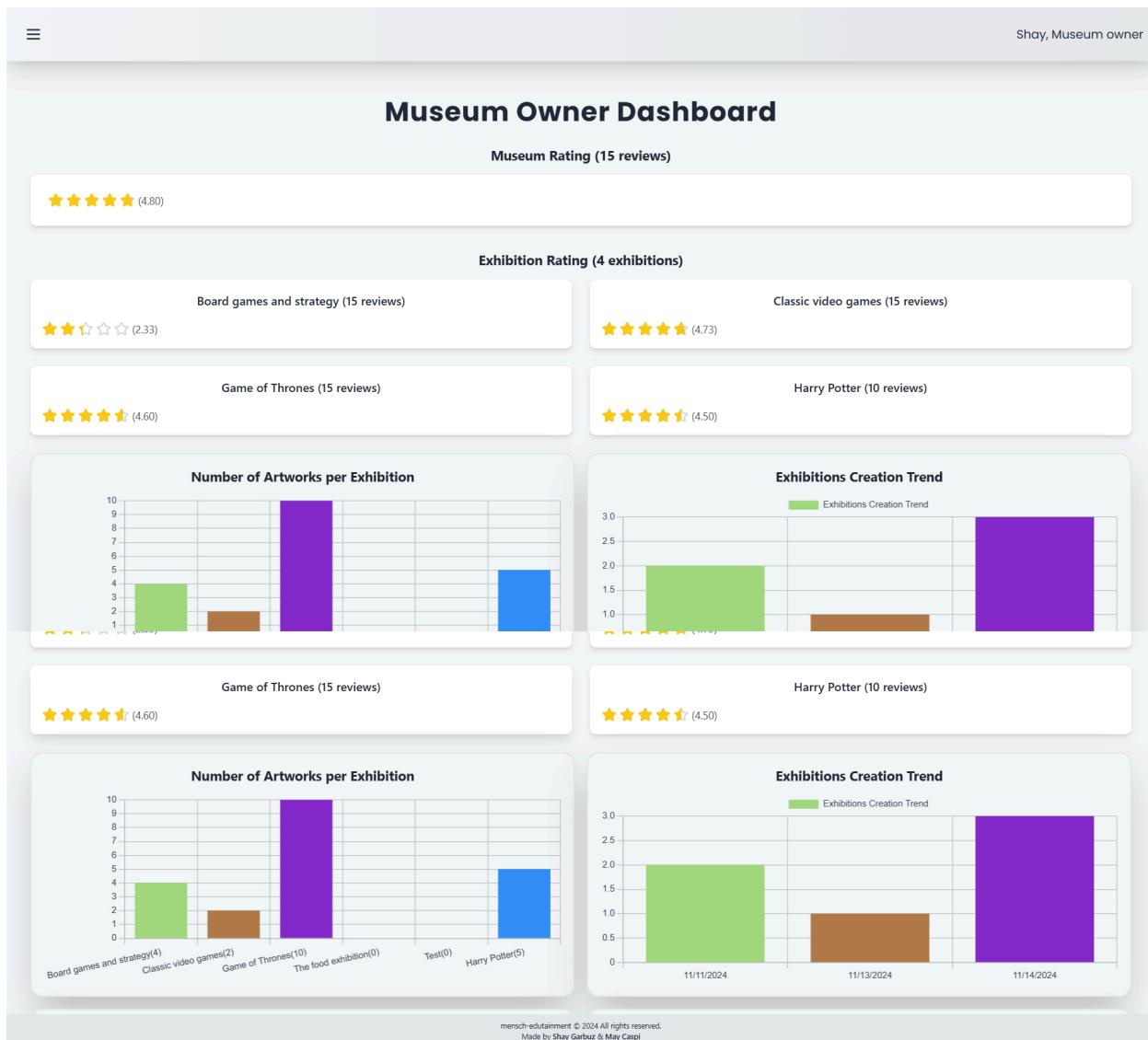
mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbuz & May Caspi

Museum Owner screens:

Purpose: Provides insights into museum and exhibition performance.

Features:

1. Museum Rating: Displays the average rating based on user reviews.
2. Exhibition Ratings: Shows individual ratings for each exhibition.
3. Number of Artworks per Exhibition: Bar chart showing how many artworks are in each exhibition.
4. Exhibitions Creation Trend: Bar chart illustrating the number of exhibitions created over time.

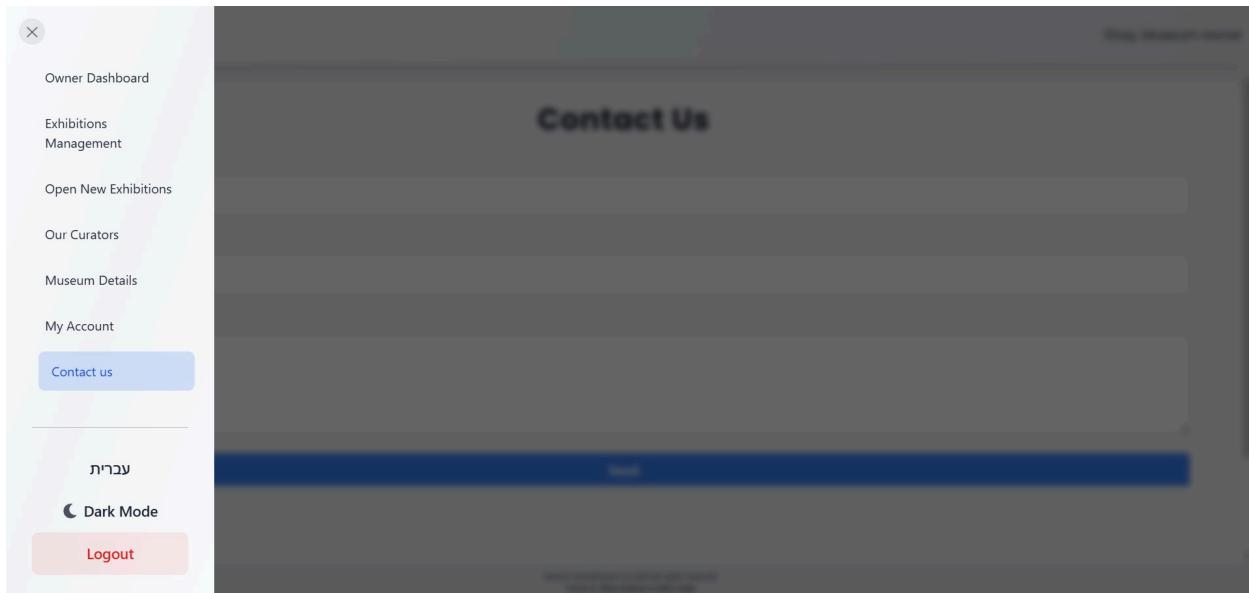


Sidebar:

Purpose: Provides museum owners with quick access to key sections.

Menu Items:

- Owner Dashboard: View overall performance and insights.
- Exhibitions Management: Manage all exhibitions.
- Open New Exhibitions: Create new exhibitions.
- Our Curators: View and manage curator profiles.
- Museum Details: Update museum information.
- My Account: Manage account settings.
- Contact Us: Send inquiries or feedback.



Your Exhibitions:

Purpose: Manage and explore all exhibitions (in his museum).

Features:

1. Filters:

- **Show All:** Display all exhibitions.
- **Show Opened Exhibitions:** View only active exhibitions.
- **Show Closed Exhibitions:** View inactive exhibitions.

2. Exhibition Cards:

- Each card displays the exhibition's name, number of artworks, curators, and a brief description.
- Click a card to view or edit the exhibition.

Shay, Museum owner

Your Exhibitions

To view or edit an exhibition, simply click on the tab and start exploring.

Show All **Show Opened Exhibitions** Show Closed Exhibitions



Board games and strategy

Artworks: 4
Curators: curator1, curator2
Journey through time and across cultures as you explore a world of classic board games. Discover rare and unique versions...



Classic video games

Artworks: 2
Curators: curator1
Relive the golden age of gaming with classic consoles and games from the 70s, 80s, and beyond. Experience the evolution...

Edit Exhibition

Purpose: Modify exhibition details, manage artworks, and assign curators.

Fields:

- **Exhibition Name:** Update the title.
- **Max Artworks:** Set a limit for artworks.
- **Description:**
 - Write manually or use **AI Generate**.
 - **AI Generate:** Creates a description based on the title and image by analyzing key themes.
- **Image:** Upload, replace, or view the exhibition image.

Curators:

Shay, Museum owner

Exhibition name: Classic video games

Edit Exhibition View Artworks Add New

Exhibition Name: Classic video games

Maximum Number of Artworks: 10

Description: Relive the golden age of gaming with classic consoles and games from the 70s, 80s, and beyond. Experience the evolution...

Generate AI Description

Image: https://firebasestorage.googleapis.com/

View Image

Curators:

| |
|---------------------------|
| Name: curator1 |
| Surname: s1 |
| Email: curator1@gmail.com |
| Phone: +972521111111 |

+ Add Curator

Select from Curator's list

Save Changes

Would you like to close this exhibition?

- View, add (+ Add Curator) or remove assigned curators.

Actions:

- **Save Changes:** Apply updates.
- **Close Exhibition:** Mark as closed.
- **Go Back:** Return to the previous page.

[Open New Exhibition](#)

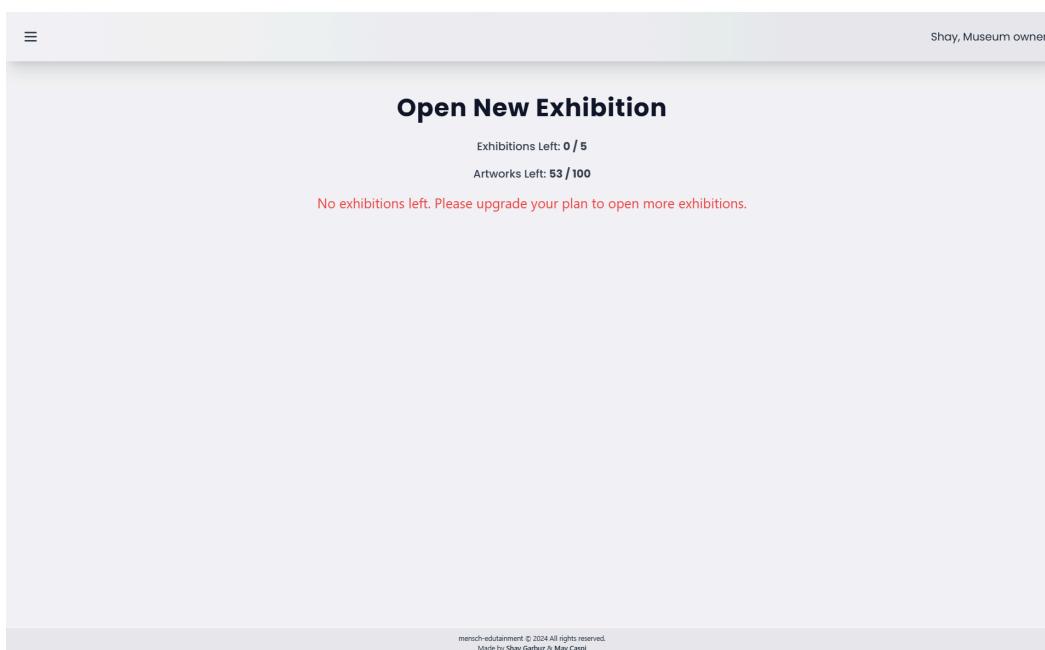
Purpose: Allows museum owners to create new exhibitions within their subscription limits.

Details:

- **Exhibitions Left:** Shows the remaining number of exhibitions that can be created based on the current plan.
- **Artworks Left:** Displays the remaining number of artworks allowed in the current plan.

Notice:

If the limit is reached (e.g., **Exhibitions Left: 0**), a message prompts the user to **upgrade the plan** to open more exhibitions.



View Artworks

Purpose: Displays all artworks in the selected exhibition.

Features:

- **Artwork Cards:** Each card shows:
 - **Artwork Name**
 - **Artist**
 - **Created Date**
 - A brief description
- **Edit Artwork:** Click on an artwork image to edit its details.

Action:

- **Go Back:** Return to the previous page.

The screenshot shows a user interface for managing an exhibition named "Classic video games". The top navigation bar includes "Edit Exhibition", "View Artworks" (which is currently selected), and "Add New Artwork". Below the navigation, a message says "In order to edit artwork, please click on the image". Two artwork cards are displayed: one for a "Game Boy" (a red handheld console) and another for an "Atari Flashback" (a black portable game console with controllers). Each card contains the artwork's name, artist, creation date, and a brief description.

The screenshot shows a form for creating a new artwork. The top navigation bar includes "Edit Exhibition", "View Artworks", and "Add New Artwork" (which is currently selected). The main form area is titled "Create Artwork" and includes fields for "Title", "Artist Name", "Created Date by Artist" (with a date input field), "Image Upload" (with a file input field and a note in Hebrew: "אנו בוחר קובץ / נבחר קובץ"), "Description", and "Record Description", "Use Speech to Text", and "Generate AI Description" buttons. A "Create Artwork" button is at the bottom. A "Go Back" button is located at the bottom right.

Add New Artwork

Purpose: Create and add a new artwork to the selected exhibition.

Fields:

- **Title:** Enter the name of the artwork.
- **Artist Name:** Provide the artist's name.
- **Created Date by Artist:** Specify the creation date of the artwork.
- **Image Upload:** Upload an image of the artwork.
- **Description:** Add a brief description of the artwork.

Description Tools:

- **Record Description:** Record an audio description of the artwork.

- **Use Speech to Text:** Convert speech to a text description.
- **Generate AI Description:** Auto-generate a description based on the uploaded image and title.

Update Artwork

Purpose: Modify existing artwork details in the selected exhibition.

Fields:

- **Title:** Edit the artwork's title.
- **Artist Name:** Update the name of the artist.
- **Created Date by Artist:** Change the creation date of the artwork.
- **Image Upload:** Upload a new image if necessary.
- **Existing Image:** View or replace the current image.
- **Description:** Update the artwork's description.

Description Tools:

- **Record Description:** Record an audio description.
- **Use Speech to Text:** Convert speech into a text description.
- **Generate AI Description:** Automatically create a description based on the image and title.

The screenshot shows a user interface for updating artwork details. At the top, it says "Shay, Museum owner". Below that, it shows the "Exhibition name: Game of Thrones". There are three tabs: "Edit Exhibition", "View Artworks" (which is currently selected), and "Add New Artwork".

The main section is titled "Update Artwork". It has fields for "Title" (Daenerys Targaryen), "Artist Name" (HBO), and "Created Date by Artist" (17/05/2011). There is a "Image Upload" button with the text "אֶל בְּבוֹרָ קִוְעַן חֲבַרְתָּ קִוְעַן". Below that is a "Existing Image" section showing a portrait of Daenerys Targaryen, with a "View Image" link. A "Description" field contains the text: "'Daenerys Targaryen,' a still image from the HBO series created on". Below the description are three buttons: "Record Description", "Use Speech to Text", and "Generate AI Description". At the bottom are "Update Artwork" and "Delete Artwork" buttons, and a "Go Back" link.

Edit Museum Details

Purpose: Allows museum owners to update their museum's information.

Fields:

- **Museum Name:** Update the name of the museum.
- **Address:** Edit the street address, city, state, and zip code.
- **Phone Number:** Enter or update the contact number.
- **Email:** Update the registered email address.
- **Plan:** Displays the current subscription plan.
- **Image URL:** Add or change the museum's display image.
- **Password:** Set a new password or generate a random one using **Generate Password**.

Actions:

- **Save Changes:** Apply the updated details.
- **Get QR Code:** Generate a QR code for the museum app.

The screenshot shows a user interface for editing museum details. At the top right, it says "Shay, Museum owner". Below that are several input fields: "State" (set to Israel), "Zipcode" (set to 2725302), "Phone Number" (set to 0522835144), "Email" (set to shay.garbz@gmail.com), and a "Plan" section showing "Introductory Plan - 5 Exhibitions, 100 Artworks" with an "Image URL" link to https://travel.geek.nz/wp-c and a thumbnail image. There are also sections for "Email" (shay.garbz@gmail.com) and "Plan" (Introductory Plan - 5 Exhibitions, 100 Artworks, shay.garbz@gmail.com). At the bottom are "Save Changes" and "Get QR code" buttons, with a note at the very bottom about copyright.

Edit Your Information

Purpose: Allows users to update their personal details.

The screenshot shows a user interface for editing personal information. At the top right, it says "Shay, Museum owner". The main title is "Edit Your Information". Below are three input fields: "First Name" (Shay), "Last Name" (Garbz), and "Phone Number" (+972522835144). At the bottom is a large blue "Save Changes" button. A copyright notice at the bottom states "mensch-edutainment © 2024 All rights reserved. Made by Shay Garbz & May Caspi".

Curators Management

Purpose: View and manage curators associated with exhibitions.

Features:

- **Curator List:** Displays each curator's full name, email, phone number, and the exhibitions they are assigned to.
- **Disable Curator:** Click **Disable Curator** to remove a curator's access and involvement in exhibitions.

Actions:

- **Disable Curator:** Prevents the curator from managing exhibitions.

Contact Us

Purpose: Allows museum owners to send an email to the admin.

Phone view:

The screenshot shows a mobile application interface for sending a message. At the top, there is a header bar with three horizontal lines on the left and the text "Shay, Museum owner" on the right. Below this is a section titled "Contact Us". It contains three input fields: "Name" with the value "Shay Garbuz", "Email" with the value "Shay.garbuz@gmail.com", and "Message" which is currently empty. At the bottom of the screen is a large blue rectangular button labeled "Send". At the very bottom of the phone screen, there is a small grey footer bar with the text "mensch-edutainment © 2024 All rights reserved. Made by Shay Garbuz & May Caspi".

The screenshot shows a tablet displaying the "Curators" section. At the top, there is a header bar with three horizontal lines on the left and the text "Shay, Museum owner" on the right. Below this is a section titled "Curators". It lists two curators: "curator1 s1" and "curator2 s2". Each curator entry includes their name, email, phone number, and a list of exhibitions they are involved in. To the right of each curator entry is a red rectangular button labeled "Disable Curator". At the bottom of the tablet screen, there is a small grey footer bar with the text "mensch-edutainment © 2024 All rights reserved. Made by Shay Garbuz & May Caspi".

Curator's screens

Manage Your Exhibitions:

Purpose: Enables curators to view, edit, and manage exhibitions assigned to them.

Features:

- **Exhibition Cards:** Each card displays:
 - Exhibition Title
 - Artworks: List of artworks included.
 - Curators: Assigned curators for the exhibition.
 - Museum: Associated museum name.
 - Brief Description: A short description of the exhibition.
- **Search Bar:** Allows curators to quickly find exhibitions by museum name, exhibition title, or artwork name.

Actions:

- Click on an Exhibition Card: Opens the detailed view or editing options for the selected exhibition.

The screenshot shows a mobile application interface titled "Manage your exhibitions". At the top, there is a search bar with the placeholder "Search by museum, exhibition, or artwork name". Below the search bar are two exhibition cards. The first card is titled "Board games and strategy" and features a thumbnail image of a board game setup. It includes details such as "Artworks: The Settlers of Catan, Monopol, Star Wars Shatterpoint, Twilight Struggle", "Curators: curator1, curator2", and a brief description: "Journey through time and across cultures as you explore a world of classic board games. Discover rare and unique versions of chess, G...". The second card is titled "Classic video games" and features a thumbnail image of a collection of vintage video game consoles and boxes. It includes details such as "Artworks: Game Boy, Atari Flashback", "Curators: curator1", and a brief description: "Relive the golden age of gaming with classic consoles and games from the 70s, 80s, and beyond. Experience the evolution of video...". Both cards have a "Museum: Garbz Games" footer.

Edit Exhibition

Purpose: Enables curators to update the exhibition's details and image.

Fields:

- **Exhibition Name:** Modify the title of the exhibition.
- **Description:** Edit the exhibition description to reflect new or updated information.
- **Generate AI Exhibit Description:** Automatically generate a description based on the exhibition's content.
- **Upload Exhibition Image:** Add or replace the exhibition's main image.
- **Uploaded Image:** Displays the currently uploaded image with a preview option.

The screenshot shows a mobile application interface titled "Edit Exhibition" for the exhibition "Board games and strategy". At the top, there are three navigation buttons: "Edit Exhibition" (which is active), "View Artworks", and "Add New Artwork". Below the buttons, there is a form field for "Exhibition Name" containing the value "Board games and strategy". There is also a "Description" field with a placeholder: "Journey through time and across cultures as you explore a world of classic board games. Discover rare and unique versions...". A "Generate AI Exhibit Description" button is located next to the description field. Below the description field is an "Upload Exhibition Image" section with a file input field labeled "לadd נבחר דוחץ" and a preview image of the board game setup. A "Save Changes" button is at the bottom right, and a "Go Back" button is at the bottom center.

View Artworks

Purpose: Displays all artworks in the selected exhibition.

Features:

- **Artwork Cards:**
 - **Title:** Artwork title.
 - **Artist:** Name of the artist who created the artwork.
 - **Created Date:** The date the artwork was created.
 - **Description:** A brief description of the artwork.
- **Navigation Tabs:**
 - **Edit Exhibition:** Modify exhibition details.
 - **View Artworks:** Current view showing all artworks in the exhibition.
 - **Add New Artwork:** Opens the form to add new artworks to the exhibition.

The screenshot shows a digital interface for viewing artworks. At the top, there's a navigation bar with three horizontal lines on the left, the text "curator1, Curator" in the center, and a search bar on the right. Below the navigation bar, the title "Board games and strategy" is displayed, followed by a blue link "Visit the museum". There are three tabs at the top of the main content area: "Edit", "View Artworks" (which is underlined in blue), and "Add New Artwork".

The first artwork card features an image of the board game "The Settlers of Catan". The title "The Settlers of Catan" is in bold, followed by "Artist: Klaus Teuber" and "Created Date: 1.1.1995". A small note below states: "Klaus Teuber's 'The Settlers of Catan' (01/01/95) is a seminal work of board game design presented as an artistic...".

The second artwork card features an image of the board game "Monopoly". The title "Monopoly" is in bold, followed by "Artist: Charles Darrow" and "Created Date: 12.4.1935". A note below says: "'Monopoly,' created by Charles Darrow on 12/04/35, is a board game that has become a global phenomenon. The...".

The third artwork card features an image of the board game "Star Wars Shatterpoint". The title "Star Wars Shatterpoint" is in bold, followed by "Artist: Atomic Mass Games" and "Created Date: 2.6.2023". A note below says: "'Star Wars Shatterpoint,' a vibrant artwork by Atomic Mass Games (02/06/23), depicts a dynamic skirmish between iconic Star...".

The fourth artwork card features an image of the board game "Twilight Struggle". The title "Twilight Struggle" is in bold, followed by "Artist: GMT Games" and "Created Date: 1.1.2005". A note below says: "'Twilight Struggle,' a 2005-01-01 artwork by GMT Games, presents a stylized depiction of the Cold War. Using a graph...".

At the bottom right of the page, there is a blue button labeled "Go Back". In the bottom right corner of the entire page, there is a small footer note: "merch-ed-ajamment © 2014 All rights reserved. Made by Shay Garber & May Gepf".

Update Artwork Screen

Purpose:

Allows curators to update details of an existing artwork in an exhibition.

Key Features:

- **Editable Fields:** Title, Artist Name, Created Date, Description.
- **Media:** Record description, Update image, play existing audio, or upload new.
- **AI Tools:**
 - Speech-to-text.
 - Auto-generate description.

Actions:

- **Update Artwork:** Save changes.
- **Delete Artwork:** Remove artwork.
- **Go Back:** Return to artworks list.

The screenshot shows the 'Update Artwork' interface. At the top right, it says 'curatorl, Curator'. Below that is the title 'Update Artwork'. The form includes fields for 'Title' (The Settlers of Catan), 'Artist Name' (Klaus Teuber), 'Created Date by Artist' (01/01/1995), and an 'Image Upload' section with a placeholder 'Existing Image' showing a thumbnail of the board game. There's a 'View Image' button. Below this is a 'Description' field containing the text 'Klaus Teuber's "The Settlers of Catan" (01/01/95) is a seminal work of board game'. An 'Existing Audio' section shows a play button at 0:00 / 0:01. Below are three buttons: 'Record Description', 'Use Speech to Text', and 'Generate AI Description'. At the bottom are two main buttons: 'Update Artwork' (blue) and 'Delete Artwork' (red). A 'Go Back' button is located at the very bottom.

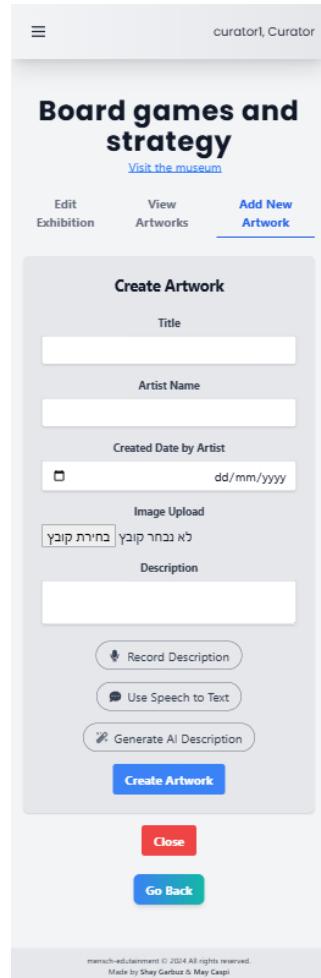
Create Artwork Screen

Purpose: Add a new artwork to an exhibition.

Fields: Title, Artist Name, Created Date, Image, Description

AI Tools: Record, Speech-to-Text, AI Description

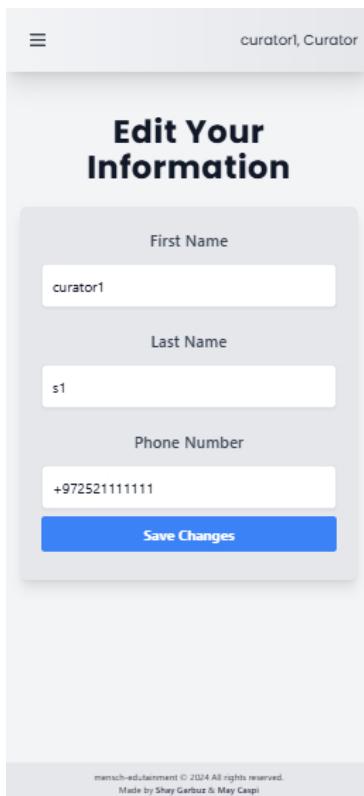
Actions: Create, Close, Go Back



Edit Your Information

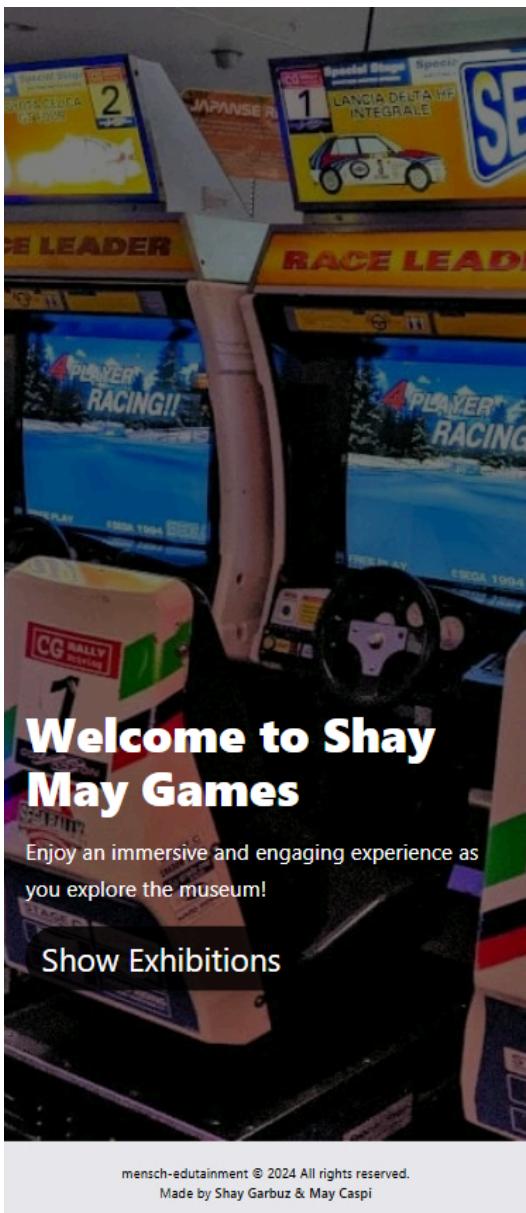
Purpose: This page allows curators to update their personal information, ensuring that contact details

Phone view:



Visitor screens (only phone view):

The visitor page opens after scanning the QR code. It provides a welcoming interface for users to explore a museum's exhibitions.



Museum screen:

This page allows museum visitors to browse, search, and review exhibitions, providing a user-friendly interface for exploring the museum's content.

1. **Header:** Shows museum name, address, and a "Leave Museum" button.
2. **Search Bar:** Quickly find exhibitions by keywords.
3. **Popular Exhibitions:** Grid view with exhibit names and images.
4. **Add Review:** Allows visitors to give feedback.

The screenshot displays the museum's digital interface. At the top, it shows the museum's name, "Shay May Games", its location, "Lilach 2/50, Israel", and a "Leave Museum" button. Below this is a search bar with the placeholder "Search Exhibit" and a magnifying glass icon. A large, semi-transparent overlay asks, "Which Exhibit Are You Looking For?". The main content area is titled "Popular Exhibitions" and includes a "★ Add Review" button. It features a 2x2 grid of exhibition thumbnails: "Board games and strategy" (showing a board game setup), "Classic video games" (showing a collection of vintage gaming consoles and controllers), "Game of Thrones" (showing a group of characters in medieval-style costumes), and "Harry Potter" (showing two characters from the Harry Potter series). At the bottom, a footer notes "mensch-edutainment © 2024 All rights reserved. Made by Shay Garbuz & May Caspi".

Exhibition screen:

This page offers a detailed view of a specific exhibition. It enhances the visitor's experience by providing rich content about the exhibition, including an audio description feature and a gallery of associated artworks. Visitors can explore artworks, search for specific items, and easily return to the main museum page.

Shay May Games [Leave Museum](#)

Lilach 2/50, Israel

Board games and strategy

Journey through time and across cultures as you explore a world of classic board games. Discover rare and unique versions of chess, Go, checkers, Risk, Monopoly, and more!

[Listen to Description](#)

Artworks

Search Artworks 



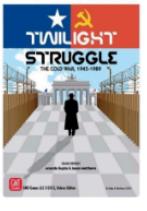
The Settlers of Catan



Monopol



Star Wars
Shatterpoint



Twilight Struggle

[Back to museum](#)

mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbuz & May Caspi

Single artwork screen:

This page is designed to offer visitors an immersive and informative experience by showcasing detailed information about a selected artwork. It includes a description, an audio commentary, and smooth navigation back to the exhibition or museum, enhancing engagement and accessibility.

The screenshot displays a mobile application interface for a museum exhibit. At the top, it shows the user's name "Shay May Games" and location "Lilach 2/50, Israel", along with a "Leave Museum" button. Below this is a header section titled "Board games and strategy" with a sub-section "Journey through time and across cultures as you explore a world of classic board games. Discover rare and unique versions of chess, Go, checkers, Risk, Monopoly, and more!". A "Listen to Description" button is present. The main content area features a large image of the board game "The Settlers of Catan" with its box. Below the image, the title "The Settlers of Catan" and the author "Klaus Teuber" are displayed. A "Description" section follows, containing a detailed paragraph about the game's history and impact. A "To hear the curator's commentary, press 'Play'" button is shown, which, when pressed, reveals a "Play" button. Navigation buttons for "Back to Artworks" and "Back to Museum" are at the bottom.

Shay May Games
Lilach 2/50, Israel
[Leave Museum](#)

Board games and strategy

Journey through time and across cultures as you explore a world of classic board games. Discover rare and unique versions of chess, Go, checkers, Risk, Monopoly, and more!

[Listen to Description](#)

[Back to Artworks](#)


The Settlers of Catan
Klaus Teuber

Description

Klaus Teuber's "The Settlers of Catan" (01/01/95) is a seminal work of board game design presented as an artistic installation. This iconic game's richly detailed hexagonal landscape visually represents its innovative gameplay and enduring cultural impact, showcasing a unique blend of strategy, resource management, and negotiation that established a milestone in modern board game history.

To hear the curator's commentary, press 'Play'

[Play](#)

[Back to Museum](#)

mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbuz & May Caspi

The feedback form aims to collect visitor ratings for specific exhibitions and their overall museum experience, helping museum administrators understand visitor satisfaction and improve future exhibits.

Shay May Games [Leave Museum](#)
Lilach 2/50, Israel



Museum Visit Feedback

Shay May Games

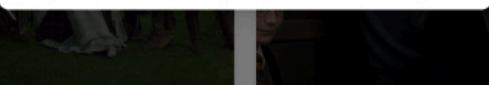
Exhibition Ratings

| | |
|--------------------------|-------|
| Board games and strategy | ★★★★★ |
| Classic video games | ★★★★★ |
| Game of Thrones | ★★★★★ |
| Harry Potter | ★★★★★ |

Overall Museum Experience

| | |
|------------------------------|-------|
| Rate your overall experience | ★★★★★ |
|------------------------------|-------|

Submit Feedback



Game of Thrones Harry Potter

mensch-edutainment © 2024 All rights reserved.
Made by Shay Garbuz & May Caspi

A user can login again to the app through this page:



9. Maintenance guide

MuseumApp Backend Programmer's Guide

1. Introduction

The **MuseumApp backend** is a robust system built to manage various museum operations, such as user management, exhibitions, artwork, subscriptions, and AI-generated descriptions using Google Gemini.

This guide provides a comprehensive understanding of how the backend functions, how to configure and extend it, and how to troubleshoot common issues.

2. Project Overview

Purpose and Objectives

The backend is designed to:

- Manage museums, exhibitions, artworks, and curators through CRUD operations.
- Provide role-based access control for users (Admins, Museum Owners, Curators, Visitors).
- Generate AI-driven descriptions for exhibitions and artworks using Google Gemini.
- Enable subscription management with plans and limitations.

Key Technologies

| Technology | Purpose |
|--------------------|--|
| Node.js | Runtime environment for the backend. |
| Express | Framework for routing and middleware. |
| MongoDB | Database for storing structured data. |
| Firebase Admin SDK | User authentication and role-based access control. |
| Google Gemini API | AI-powered text generation for descriptions. |
| Winston | Logging for debugging and monitoring. |
| Nodemailer | Sending emails for contact and request handling. |

Features Overview

1. **Authentication:** Sign-up, login, JWT-based authentication, and role-based access control.
2. **Museum Management:** Manage museums, assign owners and link exhibitions.
3. **Exhibition Management:** Create, update, and delete exhibitions with curator and artwork assignments.
4. **Artwork Management:** Manage artworks, link them to exhibitions, and generate AI-powered descriptions.
5. **Feedback System:** Collect and display reviews for museums and exhibitions.
6. **Subscription Plans:** Define and manage subscription plans with limitations.

3. Environment Setup

Prerequisites

Before working with the backend, ensure the following tools are installed:

- **Node.js:** Version 16 or above.
- **MongoDB:** Community Server (or use MongoDB Atlas).
- **Firebase:** Admin SDK setup for authentication.
- **Google Cloud:** Enabled Generative AI API.

Installing Dependencies

Run the following command to install all required packages:

```
C:\Users\shay\Desktop\MuseumApp\Backend> npm install
```

Database Setup (MongoDB)

1. Create a cluster on [MongoDB Atlas](#).
2. Replace the MONGO_CONN in .env with the connection string.

Firebase Configuration

1. Go to the Firebase Console.
2. Create a project and download the service account JSON.
3. Add the credentials to the .env file.

Google Generative AI (Gemini) Integration

1. Enable the Generative AI API on Google Cloud.
2. Create an API key and add it to .env as GEMINI_API_KEY.

Configuration and Environment Variables

1. Create a .env file in the root directory
2. Fill the .env according to:

```

# General Configuration
PORT=8080
# MongoDB Configuration
MONGO_CONN=mongodb://localhost:27017/museumapp
# JWT Configuration
JWT_SECRET=YourJWTSecret
JWT_EXPIRATION=1h
# Firebase Configuration
FIREBASE_TYPE=service_account
FIREBASE_PROJECT_ID=YourProjectID
FIREBASE_PRIVATE_KEY=YourFirebasePrivateKey
FIREBASE_CLIENT_EMAIL=YourFirebaseClientEmail
# Google Gemini API
GEMINI_API_KEY=YourGeminiAPIKey
# Email Configuration
EMAIL_USER=YourEmail@example.com
EMAIL_PASS=YourEmailPassword

```

4. Project Architecture

Folder Structure

```

Backend/
|
|__ Controllers/      # Handles API requests and responses
|__ Middlewares/     # Authentication, authorization, and validation
|__ Models/          # Mongoose schemas for MongoDB collections
|__ Routes/          # API route definitions
|__ Services/         # Business logic and database interaction
|__ Utils/           # Helper functions (JWT, logging, etc.)
|__ index.js          # Application entry point
|__ config.js         # Configuration settings
└__ package.json      # Project dependencies

```

High-Level Application Flow

1. **Request Handling:**
 - o Requests are routed via Routes/.
 - o Middleware is applied for validation, authentication, and authorization.
2. **Controller Layer:**
 - o Controllers (e.g., ExhibitionController.js) process requests and call services.

3. **Service Layer:**
 - o Services handle complex business logic and interact with models.
4. **Model Layer:**
 - o Mongoose models define the structure and relationships of data.

5. API Reference

Authentication

- **Sign-Up:** POST /api/auth/signup
 - o Validates input using AuthValidation.js.
 - o Creates a Firebase user and MongoDB user record.
- **Login:** POST /api/auth/login
 - o Validates credentials, issues a JWT.

Museums

- **Get All Museums:** GET /api/museums
- **Create Museum:** POST /api/museums
 - o Assigns the museum to an owner and links exhibitions.

Exhibitions

- **Create Exhibition:** POST /api/exhibitions
- **Update Exhibition:** PUT /api/exhibitions/:id
 - o Updates status (open, closed) and curator assignments.

Artworks

- **Create Artwork:** POST /api/artworks
- **Generate Description:** POST /api/artworks/description
 - o Uses Google Gemini for AI-generated descriptions.

Plans and Requests

- **Create Plan:** POST /api/plans
- **Handle Requests:** POST /api/requests

6. Detailed Explanation of Code and Logic

Middleware

1. **AuthenticateUser:** Validates JWT, attaches user data to req.
2. **AuthorizeUser:** Ensures role-based access.

Controllers

- Each controller (e.g., ExhibitionController) handles API routes, validates input, and delegates logic to services.

Services

- Services (e.g., ExhibitionService) encapsulate business logic and interact with Mongoose models.

Models

- Models define the schema for each collection (e.g., UserModel, MuseumModel).

7. Deployment

```
bash
npm install -g vercel
```

Install the Vercel CLI:

```
bash
vercel deploy
```

Deploy the app:

8. Debugging and Logs

Logging

- Logs are managed with Winston.
- Logs are saved in error.log and combined.log.

9. Extending the Codebase

Adding New Features

1. Define a new model in Models/.
2. Write business logic in Services/.
3. Expose functionality in Controllers/.
4. Add routes in Routes/.

MuseumApp CMS Frontend Programmer's Guide

1. Introduction

The **MuseumApp Frontend** provides a user interface for managing museums, exhibitions, artworks, and subscriptions. Built with **ReactJS** and styled with **TailwindCSS**, it integrates with Firebase for authentication, utilizes REST APIs to fetch and update data, and **MongoDB** to store and manage structured data.

The frontend also leverages AI-powered features using Google Gemini to generate artwork and exhibition descriptions.

2. Project Overview

Purpose and Objectives

The frontend application is designed to:

- Provide a user-friendly interface for various roles: Admins, Museum Owners, and Curators.
- Facilitate CRUD operations for museums, exhibitions, and artworks.
- Support AI-driven descriptions for artworks and exhibitions.
- Enable dynamic feedback and subscription management.

Key Technologies

| Technology | Purpose |
|-------------|---|
| ReactJS | Component-based frontend framework. |
| TailwindCSS | Utility-first styling framework. |
| Firebase | User authentication and storage. |
| MongoDB | NoSQL database for flexible data storage. |
| Axios | API communication and HTTP requests. |

3. Environment Setup

Prerequisites

Ensure the following tools are installed on your system:

- **Node.js**: Version 16 or above.
- **npm** or **yarn**: For dependency management.

Installing Dependencies

```
bash
npm install
```

Run the following command in the project directory:

Firebase Configuration

1. Go to the Firebase Console and set up a project.
2. Enable authentication and download the service account JSON.
3. Replace the placeholders in firebaseConfig.js with your Firebase credentials.

Environment Variables

1. Create a .env file in the root directory
2. Fill the .env according to:

```
env

REACT_APP_FIREBASE_API_KEY=<your-firebase-api-key>
REACT_APP_FIREBASE_AUTH_DOMAIN=<your-auth-domain>
REACT_APP_FIREBASE_PROJECT_ID=<your-project-id>
REACT_APP_FIREBASE_STORAGE_BUCKET=<your-storage-bucket>
REACT_APP_FIREBASE_MESSAGING_SENDER_ID=<your-messaging-sender-id>
REACT_APP_FIREBASE_APP_ID=<your-app-id>
REACT_APP_FIREBASE_MEASUREMENT_ID=<your-measurement-id>
REACT_APP_API_BASE_URL=<backend-api-url>
```

Running the Project

```
bash
npm start
```

To start the development server, use:

4. Project Architecture

Folder Structure

```
src/
├── api/           # API service files
├── assets/        # Static assets like images
├── components/   # Reusable components
├── contexts/     # Context providers for state management
├── hooks/         # Custom hooks
├── pages/         # Page-level components
├── styles/        # Global styles
├── utils/         # Utility functions
└── validators/   # Form validation logic
├── App.js          # Main app component
└── index.js        # Entry point of the app
```

High-Level Application Flow

1. **Authentication:** Firebase handles user authentication and token management.
2. **Routing:** Routes are defined in App.js with role-based access using PrivateRoute.js.
3. **State Management:** Contexts manage global states, e.g., user data and theme settings.
4. **API Calls:** Axios-based services in api/ handle communication with the backend.

5. Key Features

Authentication

- **Login and Registration:** Firebase is used for secure user authentication, supporting login and registration workflows. Firebase tokens are utilized to maintain user sessions and ensure secure communication with the backend.
- **Role-Based Routing:** The application employs PrivateRoute.js to restrict access to specific routes based on the user's role (e.g., Admin, Curator, Museum Owner).

Role-Based Access Control

The app enforces role-based access control (RBAC), ensuring only authorized users can access or perform certain actions. For example:

- Admins can manage all museums and users.
- Curators can manage exhibitions and artworks assigned to their museums.

- Museum Owners can manage their museums and assign curators.

Museum and Exhibition Management

- **CRUD Operations:** Users can create, update, and delete museums, exhibitions, and artworks through dedicated pages.
- Dynamic UI for curators and owners to update or close exhibitions.

AI-Powered Features

- Generate descriptions for artworks and exhibitions using Google Gemini APIs.

Speech-to-Text Functionality

- The app incorporates a **Speech-to-Text feature** using the Web Speech API, allowing users to dictate text instead of typing.

Email Communication with Nodemailer

- The app includes an integrated **contact form** that allows users to send messages to the museum's administrators.

User Feedback System

- Reviews are displayed dynamically, allowing museum managers to assess visitor feedback and improve their offerings.

6. API Integration

Base API Logic

The BaseApi.js file defines a reusable API service class with token-based authentication and error handling.

Individual API Services

Services like GeminiApi, ArtworksApi, and ExhibitionsApi extend the BaseApi class for specific endpoints.

Example Usage in Hooks

The useUserMuseum hook demonstrates how API services are consumed to fetch and update museum data.

7. Detailed Explanation of Code and Logic

Contexts

- UserContext.js: Manages authenticated user data.
- MuseumContext.js: Handles museum-related state.

Hooks

- useExhibitions.js: Fetches and updates exhibition data.

- useSpeechToText.js: Implements speech-to-text functionality.

Component-Based Structure

- Reusable components like ArtworkCard and Footer ensure consistent UI.

Styling

- TailwindCSS is used for responsive and modular styling.

8. Deployment

Building the Project

```
bash
npm run build
```

To create a production build, run:

Deploying

Upload the contents of the build/ folder to your hosting platform (Vercel).

9 . Debugging and Logs

Debugging Techniques

- Use browser dev tools to inspect network requests and errors.
- Add console.log statements for debugging.

Error Handling

- All API services include error-handling logic to display user-friendly messages.

10 . Extending the Codebase

Adding New Features

- Add a new API service in api/.
- Create a custom hook for business logic.
- Build reusable components for UI.

Best Practices

- Follow the existing folder structure.
- Use contexts for shared states.
- Test new features before merging.

MuseumApp Visitor Frontend Programmer's Guide

1. Introduction

The **MuseumApp Visitor Frontend** is an interactive React-based application designed to provide a seamless experience for museum visitors. Users can explore exhibitions, view artworks, leave feedback, and engage with AI-driven features.

2. Project Overview

Purpose and Objectives

The app focuses on enhancing visitor engagement by providing:

- Easy navigation of museums and exhibitions.
- AI-powered descriptions and text-to-speech features.
- Interactive feedback submission.

Key Technologies

| Technology | Purpose |
|----------------|-------------------------------------|
| ReactJS | Component-based frontend framework. |
| TailwindCSS | Utility-first styling framework. |
| Axios | User authentication and storage. |
| Web Speech API | Text-to-speech functionality. |

3. Environment Setup

Prerequisites

Ensure the following tools are installed on your system:

- **Node.js**: Version 16 or above.
- **npm** or **yarn**: For dependency management.

Installing Dependencies

```
bash
npm install
```

Run the following command in the project directory:

Environment Variables

1. Create a `.env` file in the root directory

- Fill the .env according to:

```
REACT_APP_API_BASE_URL=<Backend API Base URL>
```

Running the Project

```
bash
npm start
```

To start the development server, use:

4. Project Architecture

Folder Structure

```
visitor-front
├── public
│   └── assets
└── src
    ├── api
    ├── assets
    ├── components
    ├── contexts
    ├── hooks
    ├── layouts
    └── pages
```

High-Level Application Flow

- User logs in via MuseumLoginPage.
- Upon successful login, the app validates credentials via VisitorContext.
- Museum data is fetched using MuseumContext and displayed.
- Users interact with exhibitions, artworks, and feedback forms.

5. Key Features

Authentication

- Login and session management using **MongoDB**.
- Tokens are stored in **localStorage** for persistence.

Museum, Exhibition, and Artwork Management

- Fetch and display museums, exhibitions, and artwork details dynamically.
- Search and filter functionalities for exhibitions and artworks.

Text-to-Speech:

- Converts exhibition descriptions to audio using Web Speech API.

Feedback:

- Visitors can submit reviews for museums and exhibitions.

6. API Integration

Base API Logic

The BaseApi.js file defines a reusable API service class with token-based authentication and error handling.

Individual API Services

Services like museumApi, ArtworksApi, and ExhibitionsApi extend the BaseApi class for specific endpoints.

Example Usage in Hooks

The useUserMuseum hook demonstrates how API services are consumed to fetch and update museum data.

7. Detailed Explanation of Code and Logic

Contexts

- VisitorContext: Manages authentication and login/logout functionality.
- MuseumContext.js: Handles museum-related state.

Hooks

- useMuseumApi: Abstracts API logic for fetching and validating data.
- useTextToSpeech: Implements text-to-speech functionality for exhibition descriptions.

Component-Based Structure

- **MuseumLayout**: Provides a consistent layout for museum-related pages.
- **ProtectedRoute**: Restricts access to authenticated users.
- **ArtworkCard**: Displays individual artwork details in a grid.
- **MuseumFeedbackForm**: Handles feedback submission.

8. Deployment

Building the Project

```
bash
npm run build
```

To create a production build, run:

Deploying

Upload the contents of the build/ folder to your hosting platform (Vercel).

9 . Debugging and Logs

Debugging Techniques

- Use browser dev tools to inspect network requests and errors.
- Add console.log statements for debugging.

Error Handling

- All API services include error-handling logic to display user-friendly messages.

10 . Extending the Codebase

Adding New Features

1. Add a new API service in src/api/.
2. Create a custom hook for logic.
3. Build reusable components for UI.

Best Practices

- Follow the existing folder structure.
- Use contexts for shared states.
- Test new features before merging.