PROJECT REPORT

SEP 5, 2025

Prepared for
Depi

Prepared by
Loay Ehab
Amr Khaled
Khaled Waleed
Abdullah Sabry

INTRODUCTION

a) Purpose

The purpose of this document is to define the scope, stakeholders, and high-level requirements for the development of a comprehensive web application for a modern museum. This application aims to serve as the primary digital interface between the museum and its global audience, enhancing visitor engagement, streamlining operations, and providing universal access to the museum's collections and resources. It will facilitate online ticket and membership sales, event bookings, digital collection exploration, and educational outreach, ultimately driving both physical attendance and online patronage.

b) Project Scope

The project encompasses the design, development, testing, and deployment of a full-stack, responsive web application.

In-Scope:

A public-facing website with the following sections:
***Visit*:** Ticketing and booking system for individual and group visits.
Exhibitions & Events: Information, booking, and management system for temporary exhibitions and events, including a recommendation engine for members.
Collection: A searchable and browsable online database of the museum's artifacts with rich media and filtering options.
***Learn*:** Dedicated resources and booking systems for schools, researchers, and community programs.
Membership: A portal for users to purchase, renew, and manage museum memberships, accessing exclusive benefits.
Auxiliary Services: Information and booking for cafes, the library, archive, and study rooms.
A secure Administrative Dashboard for museum staff to manage all content, users, bookings, and inventory.
Integration with third-party services for payment processing (e.g., Stripe), email marketing (e.g., SendGrid), and analytics (e.g., Google Analytics).
User authentication and authorization for public users and administrative staff.

Out-of-Scope:

Development of native mobile apps (iOS/Android). However, the web app will be fully responsive and mobile-friendly (PWA principles may be applied).
Point-of-Sale (POS) systems for physical gift shops or cafes (though online merchandise sales could be a future phase).
Complex Enterprise Resource Planning (ERP) or internal HR systems for museum staff.
Physical hardware installation (e.g., ticketing kiosks, on-site servers).

c) Glossary and Abbreviations

API (Application Programming Interface): A set of rules that allows different software applications to communicate with each other.
Backend: The server-side part of the application that handles logic, database interactions, and authentication.

CMS (Content Management System): A software application used to create, manage, and modify digital content without needing specialized technical knowledge.
Frontend: The client-side part of the application that users interact with directly in their web browser.
PWA (Progressive Web App): A web application that uses modern web capabilities to provide an app-like experience to users.
Relational Database: A type of database that stores and provides access to data points that are related to one another (e.g. MySQL).
Responsive Design: A web design approach that makes web pages render well on a variety of devices and window sizes.
SEO (Search Engine Optimization): The process of improving the quality and quantity of website traffic from search engines.
Stakeholder: Any person or group with an interest in the project's success.
UI/UX (User Interface / User Experience): The design of the user interface for software and the experience a user has when interacting with it.

d) List of the System Stakeholders

Stakeholder Category	Description & Examples	Primary Interest
Primary Users	General Public, Visitors, Researchers, Students, Teachers	Easily find information, book tickets/events, explore collections, access educational resources.
Members	Individuals with paid memberships.	Manage their account, access exclusive content and booking privileges, renew membership.
Museum Staff	Content Curators, Marketing Team, Event Managers, Administrators.	Update website content, manage event calendars, view booking reports, manage user accounts.
Management	Museum Directors, Board of Trustees, Department Heads.	View analytics on attendance, revenue, and user engagement to inform strategic decisions.
Third-Party Services	payment Processors (Stripe), Email Services (SendGrid), Cloud Hosting (AWS/Azure).	Provide specialized services and integrate seamlessly with the application.
Developers &Designers	The project team building the application.	Create a maintainable, scalable, and well-documented system that meets all requirements.

2) Functional Requirements

a) User Requirements Specification (User Stories)

As a casual visitor, I want to browse current exhibitions and book a timed-entry ticket online so that I can guarantee my entry and avoid long queues.
As a teacher, I want to find curriculum-linked resources and book a school trip for my class so that I can provide an educational outing without administrative hassle.
As a member, I want to receive personalized event recommendations based on my interests so that I can get the most value from my membership.
As a researcher, I want to search the digital archive with advanced filters (date, material, provenance) so that I can find specific artifacts for my academic work.
As a content curator, I want to easily upload new artifact images and descriptions through an admin panel so that I can keep the online collection updated without developer assistance.
As a marketing manager, I want to see a dashboard of ticket sales and popular collection pages so that I can measure the effectiveness of our campaigns.

B) SYSTEM REQUIREMENTS SPECIFICATION (SYSTEM FUNCTIONS)

The system shall provide a user registration and authentication system (login, logout, password reset).
The system shall allow users to browse and purchase different types of tickets (adult, child, group) for specific dates and times.
The system shall integrate with a payment gateway (e.g., Stripe) to securely process transactions.
The system shall provide a CMS for authorized staff to create, read, update, and delete (CRUD) content for events, exhibitions, and artifact entries.
The system shall provide a search engine for the digital collection with filters by category, era, culture, and keyword.
The system shall allow users to create and save a personal list of favorite artifacts.
The system shall send automated confirmation emails for bookings and purchases.
The system shall provide an admin dashboard to view and manage all bookings, users, and content.

c) Requirements' Priorities (MoSCoW Scheme)

MUST HAVE: User auth, ticket booking, payment processing, basic collection browse/search, CMS for core content, responsive design.
SHOULD HAVE: Advanced collection filters, personal "Favorites" list, school booking portal, member recommendation engine, admin analytics dashboard.
COULD HAVE: User-generated public galleries, integration with library catalog for researchers, table booking for cafes, multi-language support.
WON'T HAVE (this time): Native mobile apps, augmented reality features, complex internal ERP integration, online merchandise store.

3) Non-Functional Requirements

a) General Categories

We will follow these categories: Performance, Security, Usability, Reliability, Scalability, and Maintainability.

b) & c) Non-Functional Requirements Specification with Fit Criteria

Category, Requirement, Fit Criteria (Testable)
Performance,
Usability, The design shall be responsive and function on viewports

Testing on Chrome DevTools device emulator and real devices (e.g., iPhone, Android, iPad, desktop).
Maintainability:The codebase shall be well-documented. All API endpoints shall have documentation., All major functions and modules have comments. API documentation is generated with Swagger/OpenAPI.
d) Impact on System Architecture
Performance/Scalability: Will dictate a cloud-based, microservices-oriented architecture (e.g., on AWS/Azure) allowing us to scale components (like the search service) independently. A CDN will be essential for serving high-resolution images globally.
Security: Will require a well-defined authentication service (e.g., JWT tokens), strict input validation, and regular security dependency scanning. Database will be encrypted at rest.
Reliability: Will lead to the use of load balancers, redundant servers, and automated failover processes in the cloud infrastructure.
Maintainability: Will enforce the use of a modular code structure, a consistent coding style guide, and comprehensive documentation, making it easier for new developers to onboard.

4) Design & Implementation Constraints

Technology Stack: The backend must be implemented in Asp.net. The frontend must use the Bootstrap framework.
Database: Must use MySQL database for its strong relational data integrity.

5) Requirements Validation Techniques

Formal Reviews: Organizing structured meetings where stakeholders systematically review the Software Requirements Specification (SRS) document to sign off on the requirements.
Prototype Demos: Walking stakeholders through a clickable prototype to validate that the proposed system flow and functionality meet their expectations. This helps catch misunderstandings early.
Test Case Reviews: Ensuring that the test cases written by the QA team accurately reflect the requirements. If a requirement is untestable, it needs to be rewritten.
User Acceptance Testing (UAT): Before final launch, a group of real end-users (e.g., teachers, members) will be given access to a staging environment to perform realistic tasks. Their success and feedback are the ultimate validation.
Model Validation: Using tools to check UML diagrams or data models for consistency and completeness, ensuring there are no logical conflicts in the requirements.