A Governance-First API for Museums and Cultural Institutions

Technical Report
RESTful API Project

Student: Louey Dridi
Course: IT 325 – Web Services
Supervisor: Prof. Montassar Ben Messaoud
Academic Year: 2025 – 2026

Engineering governance, trust, and continuity into cultural heritage systems.

Abstract

DOCENT is a purpose-driven backend REST API designed to support museum governance, operations, and cultural heritage stewardship. Unlike traditional museum systems that primarily focus on content presentation, DOCENT models museums as regulated institutions with responsibilities related to preservation, accessibility, transparency, and security.

The system provides authoritative digital representations of cities, museums, artifacts, guided sessions, ticket pricing policies, and governance entities such as inspections, certifications, and authenticity checks. A layered security architecture is implemented, combining role-based access control with Contextual Artifact Verification (CAV), a domain-aware authentication mechanism requiring museum administrators to demonstrate institutional knowledge of their collections.

DOCENT is implemented using Flask, PostgreSQL, and Docker, and documented through OpenAPI/Swagger. The project demonstrates how web services can encode governance, traceability, and contextual security as first-class architectural principles within a backend-first design.

Contents

Abstract **1**

1 Introduction and Problem Context **4**
1.1 Cultural and Institutional Context 4
1.2 Problem Statement 4
1.3 Proposed Solution Overview 5

2 Requirements Analysis **5**
2.1 Functional Requirements 5
2.1.1 Institutional and Geographic Management 5
2.1.2 Artifact and Collection Management 5
2.1.3 User Roles and Professional Profiles 6
2.1.4 Guide Scheduling 6
2.1.5 Ticketing and Access Policies 6
2.1.6 Governance and Compliance 6
2.1.7 Public-Facing Information 6
2.2 Non-Functional Requirements 7
2.2.1 Security and Access Control 7
2.2.2 Data Integrity and Traceability 7

3 Global Architecture and Design Choices **7**
3.1 Architectural Overview 7
3.2 Backend Layered Architecture 7
3.2.1 Routing Layer 7
3.2.2 Service Layer 8
3.2.3 Data Access Layer 8
3.3 Technology Stack 8
3.4 Infrastructure and Deployment 8
3.5 Client Applications 8

4 Data Model and Database Design **8**
4.1 Core Entities 8
4.1.1 City and Museum 9
4.1.2 Artifact 9
4.1.3 User and GuideProfile 9
4.1.4 GuidedSession 9
4.2 Governance and Operational Entities 9
4.2.1 Inspections and Certifications 9
4.2.2 Maintenance and Authenticity 9

4.2.3 Audit Logs . . . 10
4.3 Relationships and Integrity . . . 10
4.4 Database Schema Overview . . . 10

5 API Design and REST Principles **11**
5.1 Resource-Oriented Design . . . 11
5.2 Endpoint Organization . . . 11
5.2.1 Authentication and Security . . . 11
5.2.2 Users and Roles . . . 11
5.2.3 Cities . . . 11
5.2.4 Museums . . . 12
5.2.5 Artifacts . . . 12
5.2.6 Artifact Maintenance and Authenticity . . . 12
5.2.7 Guides and Guided Sessions . . . 12
5.2.8 Ticketing . . . 13
5.2.9 Governance and Compliance . . . 13
5.2.10 Content and Communication . . . 13
5.2.11 Geolocation and Mapping . . . 13

6 Authentication and Security Architecture **13**
6.1 Role-Based Access Control . . . 13
6.2 Standard Authentication Flow . . . 14
6.3 Contextual Artifact Verification (CAV) . . . 14
6.4 TOTP Enrollment . . . 14
6.5 Security Design Rationale . . . 15

7 Functional Modules **15**
7.1 Museums and Cities . . . 15
7.2 Artifacts and Collections . . . 15
7.3 Guide Scheduling . . . 15
7.4 Ticketing and Governance . . . 15

8 Testing, Documentation, and Deployment **15**
8.1 API Testing and Validation . . . 15
8.2 API Documentation . . . 16
8.3 Deployment Architecture . . . 16

9 Limitations and Future Improvements **16**
9.1 Current Limitations . . . 16
9.2 Future Extensions . . . 17

Conclusion **17**

References and Inspirations **17**

1 Introduction and Problem Context

1.1 Cultural and Institutional Context

Museums are not static buildings filled with objects, but institutional systems embedded within legal, educational, and social frameworks. As custodians of cultural heritage, museums are responsible for artifact preservation, accurate documentation, ethical exhibition, and public accessibility. These responsibilities are commonly reinforced by national regulations and international standards, which impose requirements related to traceability, transparency, and accountability.

Beyond conservation, museums serve an educational mission that extends beyond guided tours. Visitors may differ in language, cultural background, availability of guides, or accessibility needs. As a result, museums must support multiple modes of knowledge transmission, including self-guided exploration, clear visitor rules, and contextual information that remains available independently of staff presence.

Digital systems supporting museums therefore operate at the intersection of governance, operations, and public engagement. A purely content-oriented platform is insufficient to capture this complexity. Backend systems must instead reflect real-world institutional structures such as inspections, maintenance cycles, certifications, and regulated access, while preserving public trust and institutional legitimacy.

1.2 Problem Statement

Across many regions, particularly outside major tourist centers, museums face structural challenges that directly affect visitor experience and institutional sustainability. Visitor rules are often unclear or inconsistently communicated, leading to frustration and security tension. Guide availability may be limited due to staffing constraints, resulting in exhibitions that lack contextual explanation.

Operationally, artifact maintenance and authenticity verification are frequently managed through fragmented or informal processes. The absence of structured digital records complicates conservation planning, provenance verification, and compliance during inspections. Transparency around governance processes such as inspections and certifications is often limited, weakening public confidence.

These issues are amplified by seasonal dynamics. Periods of overcrowding may be followed by extended intervals of low attendance, without sufficient analytical insight to adapt operational strategies. Ticket pricing policies are often rigid, offering limited support for regional inclusion, educational access, or differentiated visitor categories.

Although these challenges are sometimes treated as secondary operational concerns, they directly influence institutional credibility, visitor trust, and long-term viability. Addressing them requires a coherent digital representation of institutional responsibilities, processes, and oversight.

1.3 Proposed Solution Overview

DOCENT is designed as a backend REST API that addresses these challenges by embedding governance, operational structure, and security at the core of the system. Rather than serving as a presentation-layer application, DOCENT functions as an authoritative engine for institutional data and rules, modeling museums as regulated entities with explicit responsibilities.

Core entities such as museums, artifacts, guides, inspections, certifications, and maintenance records are treated as first-class digital objects with well-defined relationships and auditability. Role-based access control enforces institutional responsibility, while non-destructive data management preserves historical traceability.

Security is implemented as a contextual concern aligned with institutional accountability. Administrative access is protected through domain-aware mechanisms, while public and professional users follow proportionate authentication flows. This API-first approach allows multiple client applications to consume a single authoritative backend without duplicating governance logic.

By prioritizing structure, transparency, and extensibility, DOCENT provides a stable foundation for secure and sustainable digital museum services.

2 Requirements Analysis

2.1 Functional Requirements

2.1.1 Institutional and Geographic Management

The system must support:

- Management of cities as geographic entities
- Association of museums with cities
- Retrieval of museum information for public and administrative use
- Storage of geographic attributes to support spatial services

2.1.2 Artifact and Collection Management

DOCENT must enable:

- Management of artifact metadata and classification
- Association of artifacts with museums
- Recording of maintenance activities
- Recording of authenticity checks
- Support for QR-ready identifiers

2.1.3 User Roles and Professional Profiles

The system must distinguish identity from professional responsibility through:

- User accounts for authentication
- Role assignment (visitor, guide, museum administrator, platform administrator)
- Separate guide profiles containing professional attributes

2.1.4 Guide Scheduling

Operational coordination must be supported through:

- Definition of guided sessions with time and capacity
- Assignment of guides to sessions
- Museum-level association of sessions

2.1.5 Ticketing and Access Policies

DOCENT must provide:

- Definition of ticket pricing categories
- Association of ticket categories with visits and sessions

2.1.6 Governance and Compliance

Governance processes must be explicitly represented through:

- Recording of inspections
- Management of certifications
- Preservation of governance history
- Support for authenticity verification workflows

2.1.7 Public-Facing Information

The system must support publication of:

- Events and activities
- News and announcements
- Visitor rules and institutional guidelines

2.2 Non-Functional Requirements

2.2.1 Security and Access Control

DOCENT must enforce:

- Role-based access control
- Secure password storage
- Context-aware authentication for administrators
- Audit logging of sensitive actions

2.2.2 Data Integrity and Traceability

The system must ensure:

- Referential integrity through relational constraints
- Non-destructive data management via soft deletion
- Timestamped records for historical traceability

3 Global Architecture and Design Choices

3.1 Architectural Overview

DOCENT adopts a backend-first architecture in which the REST API acts as the single source of truth for institutional data, governance rules, and security policies. All critical logic is enforced at the API level rather than delegated to client applications.

This approach ensures:

- Consistent enforcement of institutional rules
- Centralized governance and auditability
- Client-agnostic consumption of authoritative data

Client applications function exclusively as presentation layers and do not encode business or security logic.

3.2 Backend Layered Architecture

3.2.1 Routing Layer

Responsible for:

- Exposing RESTful endpoints
- Handling HTTP requests and responses
- Enforcing role-based access control

3.2.2 Service Layer

Responsible for:

- Implementing business rules and workflows
- Coordinating operations across entities
- Applying contextual security checks

3.2.3 Data Access Layer

Responsible for:

- Defining database schemas and relationships
- Enforcing data integrity constraints
- Supporting soft deletion and traceability

3.3 Technology Stack

DOCENT is implemented using:

- Python 3.11
- Flask and Flask-RESTX
- SQLAlchemy ORM
- PostgreSQL

3.4 Infrastructure and Deployment

Docker and Docker Compose provide:

- Reproducible environments
- Service isolation between API and database
- Environment-based configuration

3.5 Client Applications

Client applications are developed separately and consume the API exclusively through documented endpoints. Governance logic, validation, and security enforcement remain strictly centralized in the backend.

4 Data Model and Database Design

4.1 Core Entities

DOCENT defines a set of core entities that represent the foundational components of the museum ecosystem.

4.1.1 City and Museum

Cities are modeled as geographic and administrative containers for museums. This design choice allows museums to be contextualized within a territorial structure, enabling regional organization, geographic queries, and future spatial services.

Museums are modeled as central institutional entities. Each museum acts as an owner and anchor for most domain objects in the system, including artifacts, guided sessions, inspections, and certifications.

4.1.2 Artifact

Artifacts represent the primary cultural assets managed by museums. Each artifact is associated with a single museum and includes descriptive metadata, classification information, and identifiers suitable for physical–digital linkage (e.g., QR codes). Artifacts are designed to serve as reference points for multiple processes, including maintenance, authenticity verification, and contextual security mechanisms.

4.1.3 User and GuideProfile

Users represent authentication identities within the system and are responsible for login, authorization, and role assignment. However, professional roles are not embedded directly into the user entity.

Guide profiles are modeled as separate entities that store professional information such as languages, areas of specialization, and institutional affiliation. This separation allows the system to distinguish between identity, authorization, and professional responsibility, which reflects real-world organizational structures and improves flexibility.

4.1.4 GuidedSession

Guided sessions model scheduled visits led by guides. Each session is associated with a museum, a guide profile, and a defined time window. Additional attributes such as group size and ticket category allow sessions to serve both operational planning and future analytical purposes.

4.2 Governance and Operational Entities

In addition to core entities, DOCENT explicitly models governance and operational processes as independent database tables. This design ensures that oversight and accountability are embedded into the system rather than treated as optional extensions.

4.2.1 Inspections and Certifications

Inspections record formal evaluations of museums, including dates, outcomes, and remarks. Certifications represent compliance with standards, such as ISO-style quality or safety requirements, and include validity periods.

4.2.2 Maintenance and Authenticity

Artifact maintenance records capture conservation activities performed on artifacts over time. Authenticity checks record verification processes and outcomes related to provenance and legit-

imacy.

4.2.3 Audit Logs

Audit logs record security-sensitive actions and changes within the system. Each log entry captures contextual information such as the acting user, the affected entity, and the time of the action.

4.3 Relationships and Integrity

The data model is designed around a museum-centric ownership principle. Most entities are either directly or indirectly associated with a museum, ensuring that data remains scoped to institutional boundaries.

To preserve integrity and traceability, the following mechanisms are employed:

- UUID primary keys to ensure global uniqueness
- Explicit foreign key relationships to enforce referential integrity
- Soft deletion flags to prevent irreversible data loss
- Timestamped records to support historical reconstruction

4.4 Database Schema Overview

Figure 4.1 presents an overview of the relational database schema implemented in DOCENT. The schema illustrates the central role of museums, the separation between core, operational, and governance entities, and the explicit relationships that support traceability and institutional oversight.

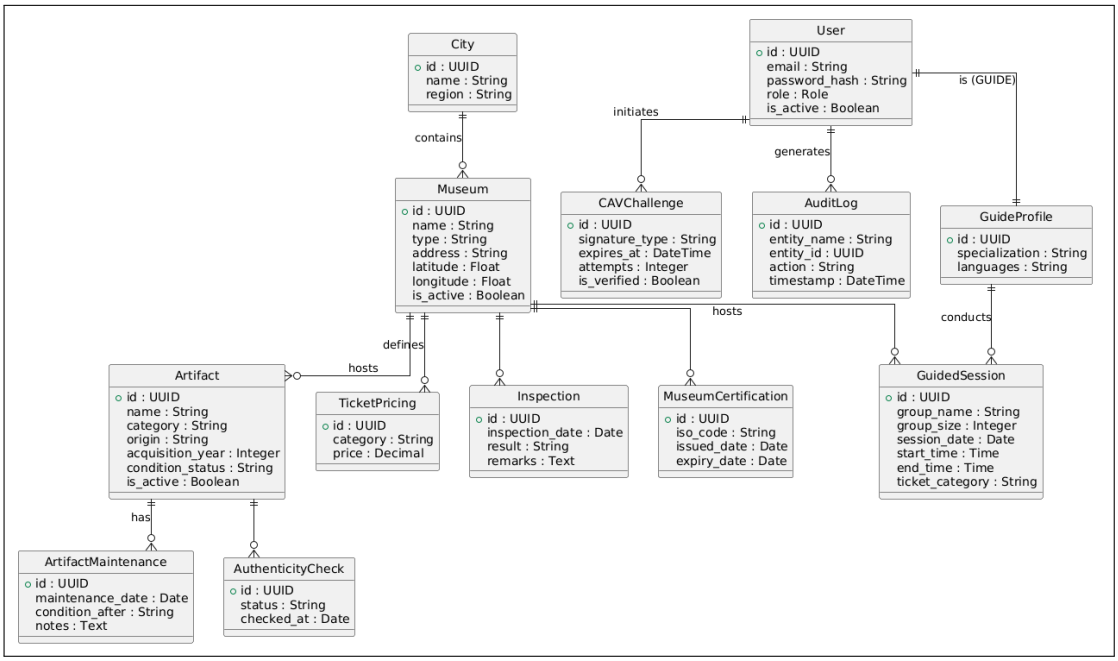

Figure 4.1: DOCENT Relational Database Schema

5 API Design and REST Principles

5.1 Resource-Oriented Design

DOCENT follows REST architectural principles by modeling domain concepts as resources accessible through stable and predictable URLs. Each resource is manipulated using appropriate HTTP methods (GET, POST, PUT, DELETE), and endpoints are named using plural nouns to reflect collections.

5.2 Endpoint Organization

Endpoints are grouped by domain to reflect institutional responsibilities and improve discoverability. The following list presents the complete set of API endpoints implemented in DOCENT.

5.2.1 Authentication and Security

- POST `/auth/login`
- POST `/auth/register`
- POST `/auth/otp/verify`
- POST `/auth/cav/challenge`
- POST `/auth/cav/verify`

5.2.2 Users and Roles

- GET `/users`
- GET `/users/{id}`
- PUT `/users/{id}`
- DELETE `/users/{id}`

5.2.3 Cities

- GET `/cities`
- POST `/cities`
- GET `/cities/{id}`
- PUT `/cities/{id}`
- DELETE `/cities/{id}`

5.2.4 Museums

- GET `/museums`
- POST `/museums`
- GET `/museums/{id}`
- PUT `/museums/{id}`
- DELETE `/museums/{id}`

5.2.5 Artifacts

- GET `/artifacts`
- POST `/artifacts`
- GET `/artifacts/{id}`
- PUT `/artifacts/{id}`
- DELETE `/artifacts/{id}`

5.2.6 Artifact Maintenance and Authenticity

- GET `/artifacts/{id}/maintenance`
- POST `/artifacts/{id}/maintenance`
- GET `/artifacts/{id}/authenticity`
- POST `/artifacts/{id}/authenticity`

5.2.7 Guides and Guided Sessions

- GET `/guides`
- GET `/guides/{id}`
- GET `/guides/me/schedule`
- GET `/guides/me/sessions`
- GET `/guides/sessions`
- POST `/guides/sessions`
- PUT `/guides/sessions/{id}`
- DELETE `/guides/sessions/{id}`

5.2.8 Ticketing

- GET `/tickets/pricing`
- POST `/tickets/pricing`
- PUT `/tickets/pricing/{id}`
- DELETE `/tickets/pricing/{id}`

5.2.9 Governance and Compliance

- GET `/governance/inspections`
- POST `/governance/inspections`
- GET `/governance/certifications`
- POST `/governance/certifications`

5.2.10 Content and Communication

- GET `/events`
- POST `/events`
- GET `/news`
- POST `/news`
- GET `/rules`

5.2.11 Geolocation and Mapping

- GET `/geo/museums`
- GET `/geo/routes`

6 Authentication and Security Architecture

6.1 Role-Based Access Control

DOCENT enforces access control through a role-based access control (RBAC) model. Four distinct roles are defined:

- **Visitor**: public users with read-only access to public information
- **Guide**: professional users with access to assigned guided sessions
- **Museum Administrator**: institutional users responsible for managing collections and governance data
- **Platform Administrator**: system-level users with global oversight

Access restrictions are enforced at the API level using role-based decorators applied to endpoints. This ensures that authorization decisions are centralized, consistent, and independent of client-side behavior. By applying the principle of least privilege, DOCENT limits each role to the minimum set of actions required to fulfill its responsibilities.

6.2 Standard Authentication Flow

All users authenticate using a standard password-based login process. Passwords are never stored in plaintext and are secured using cryptographic hashing (bcrypt). Upon successful authentication, the system issues a JSON Web Token (JWT) that encapsulates the user's identity and role information.

For users with lower-risk profiles, such as visitors and guides, the authentication flow may include a one-time password (OTP) verification step delivered via email or SMS. This mechanism provides an additional layer of assurance while maintaining usability and accessibility for public-facing roles.

6.3 Contextual Artifact Verification (CAV)

Museum administrators have elevated privileges and direct responsibility over cultural assets. As a result, DOCENT introduces Contextual Artifact Verification (CAV) as a step-up authentication mechanism specifically designed for institutional roles.

During the login process, administrators are presented with a contextual challenge derived from their own museum's collection. The system randomly selects an artifact associated with the administrator's institution and requests verification of a hidden signature field, such as an acquisition year or internal catalog attribute. These signature fields are stored securely and are never exposed to the client.

CAV provides several security and governance advantages:

- It verifies institutional knowledge rather than possession of a generic credential
- It is resistant to phishing and credential reuse attacks
- It introduces no dependency on external authentication services
- It reinforces accountability by tying access to curatorial familiarity

All CAV attempts are time-limited, attempt-limited, and fully logged in dedicated audit tables. This ensures traceability and supports forensic analysis in case of suspicious activity.

6.4 TOTP Enrollment

After successfully completing Contextual Artifact Verification, museum administrators enroll a Time-Based One-Time Password (TOTP) authenticator. Enrollment is performed through a QR-code-based setup compatible with common authenticator applications.

TOTP adds a device-based second factor that strengthens security for administrative access without replacing contextual verification.

6.5 Security Design Rationale

The authentication and security architecture of DOCENT is guided by the principle that security should reflect institutional responsibility rather than applying uniform controls to all users. By combining RBAC, standard authentication, contextual verification, and optional multi-factor mechanisms, DOCENT achieves a balance between usability, accountability, and risk mitigation.

7 Functional Modules

7.1 Museums and Cities

Museums are organized within cities to provide geographic and administrative context. Museums act as the central ownership entities for artifacts, guided sessions, and governance records. Public visibility is controlled through role-based access.

7.2 Artifacts and Collections

Artifacts are managed as institutional assets with associated metadata, maintenance records, and authenticity checks. QR-ready identifiers support linkage between physical artifacts and digital records, enabling traceability and lifecycle management.

7.3 Guide Scheduling

Guide profiles store professional attributes independently from authentication identities. Guided sessions define scheduled visits associated with museums and guides, enabling operational planning and role-specific schedule access.

7.4 Ticketing and Governance

Ticket pricing categories support differentiated access policies and provide a foundation for future analysis. Governance records, including inspections and certifications, are managed as independent entities and exposed through controlled endpoints to promote transparency.

8 Testing, Documentation, and Deployment

8.1 API Testing and Validation

DOCENT was tested primarily through manual and exploratory testing. Testing activities were conducted throughout development to validate both functional and security-related behavior.

The following testing methods were used:

- Interactive testing using the Swagger UI to verify endpoint availability, request validation, and response formats
- API client testing using Insomnia to simulate real client interactions

- Manual verification of role-based access control to ensure that protected endpoints are accessible only to authorized roles
- End-to-end testing of authentication flows, including standard login, OTP verification, and Contextual Artifact Verification (CAV)

Special attention was given to security-sensitive workflows. Administrative authentication flows were tested under different scenarios, including incorrect credentials, expired tokens, and invalid CAV responses, to verify that access was consistently denied and properly logged.

8.2 API Documentation

DOCENT is documented using the OpenAPI specification through Flask-RESTX, providing a machine-readable and human-accessible description of the API. The generated Swagger UI serves as the primary documentation interface.

The documentation includes:

- A complete list of available endpoints grouped by domain
- Detailed request and response schemas
- Required authentication and authorization information
- Interactive request execution for testing purposes

8.3 Deployment Architecture

DOCENT is deployed using Docker Compose, which orchestrates the application and database services in a reproducible environment.

The deployment includes:

- A containerized Flask API service
- A PostgreSQL database service for persistent storage
- Environment-based configuration using variables
- Health checks to ensure service availability

9 Limitations and Future Improvements

9.1 Current Limitations

The current version of DOCENT focuses on providing a solid and well-structured backend architecture rather than a feature-complete production system. As a result, several capabilities are either simplified or intentionally deferred.

Advanced analytical features, such as visitor flow analysis, peak-hour detection, and artifact popularity metrics, are not implemented beyond foundational data structures. While the

data model supports future analysis, the current implementation does not include a dedicated analytics processing layer.

AI-assisted features, including automated artifact explanations or natural-language search, are not active in the current system. These capabilities are considered future-facing extensions and are deliberately excluded to avoid introducing opaque decision-making into governance-critical workflows.

9.2 Future Extensions

DOCENT is designed to be extensible, and several future improvements are anticipated without requiring fundamental changes to the existing architecture.

Planned extensions include AI-assisted artifact explanations tailored to different audiences (e.g., visitors, students, or researchers), implemented as external services consuming authoritative backend data. Natural-language search capabilities could translate user queries into structured filters without altering core governance logic.

Advanced analytics could be introduced to analyze visitor composition, guide utilization, and ticket category distribution, providing museums with actionable insights while preserving data integrity. Mobile applications and additional client interfaces could further improve accessibility and outreach.

Optional AR/VR or 3D model integrations may be explored to enhance visitor engagement. These features would be treated as complementary visualization layers, ensuring that DOCENT remains the authoritative source of institutional data and rules.

Conclusion

DOCENT demonstrates how a RESTful backend API can be designed to reflect institutional governance, accountability, and contextual security rather than focusing solely on content delivery. By modeling museums as regulated entities and encoding operational and oversight processes directly into the data model, the system establishes a clear and authoritative foundation for cultural heritage management. The introduction of Contextual Artifact Verification illustrates how security mechanisms can be aligned with institutional responsibility instead of relying exclusively on generic authentication methods.

References and Inspirations

- ISO 21127:2014. *Information and Documentation — A Reference Ontology for the Interchange of Cultural Heritage Information (CIDOC CRM).* International Organization for Standardization. Available at: `https://www.iso.org/standard/57832.html`
- OpenAPI Initiative. *OpenAPI Specification.* Available at: `https://swagger.io/specification/`
- Python Software Foundation. *Python 3 Documentation.* Available at: `https://docs.python.org/3/`

- Flask-RESTX Contributors. *Flask-RESTX Documentation.* Available at: `https://flask-restx.readthedocs.io/`
- SQLAlchemy Authors. *SQLAlchemy ORM Documentation.* Available at: `https://docs.sqlalchemy.org/`
- PostgreSQL Global Development Group. *PostgreSQL Documentation.* Available at: `https://www.postgresql.org/docs/`
- Docker Inc. *Docker Documentation.* Available at: `https://docs.docker.com/`
- JSON Web Token (JWT) Working Group. *JSON Web Token (JWT) RFC 7519.* Available at: `https://datatracker.ietf.org/doc/html/rfc7519`
- PyJWT Contributors. *PyJWT Documentation.* Available at: `https://pyjwt.readthedocs.io/`
- Google. *Gmail SMTP Server Documentation.* Available at: `https://support.google.com/mail/answer/7126229`
- Python Software Foundation. *smtplib — SMTP Protocol Client.* Available at: `https://docs.python.org/3/library/smtplib.html`
- Insomnia REST Client. *Insomnia Documentation.* Available at: `https://docs.insomnia.rest/`