Project Design Phase
Solution Architecture

Date	17 February 2026
Team ID	LTVIP2026TMIDS24224
Project Name	Gemini Historical Artifact Description System
Maximum Marks	4 Marks

Solution Architecture

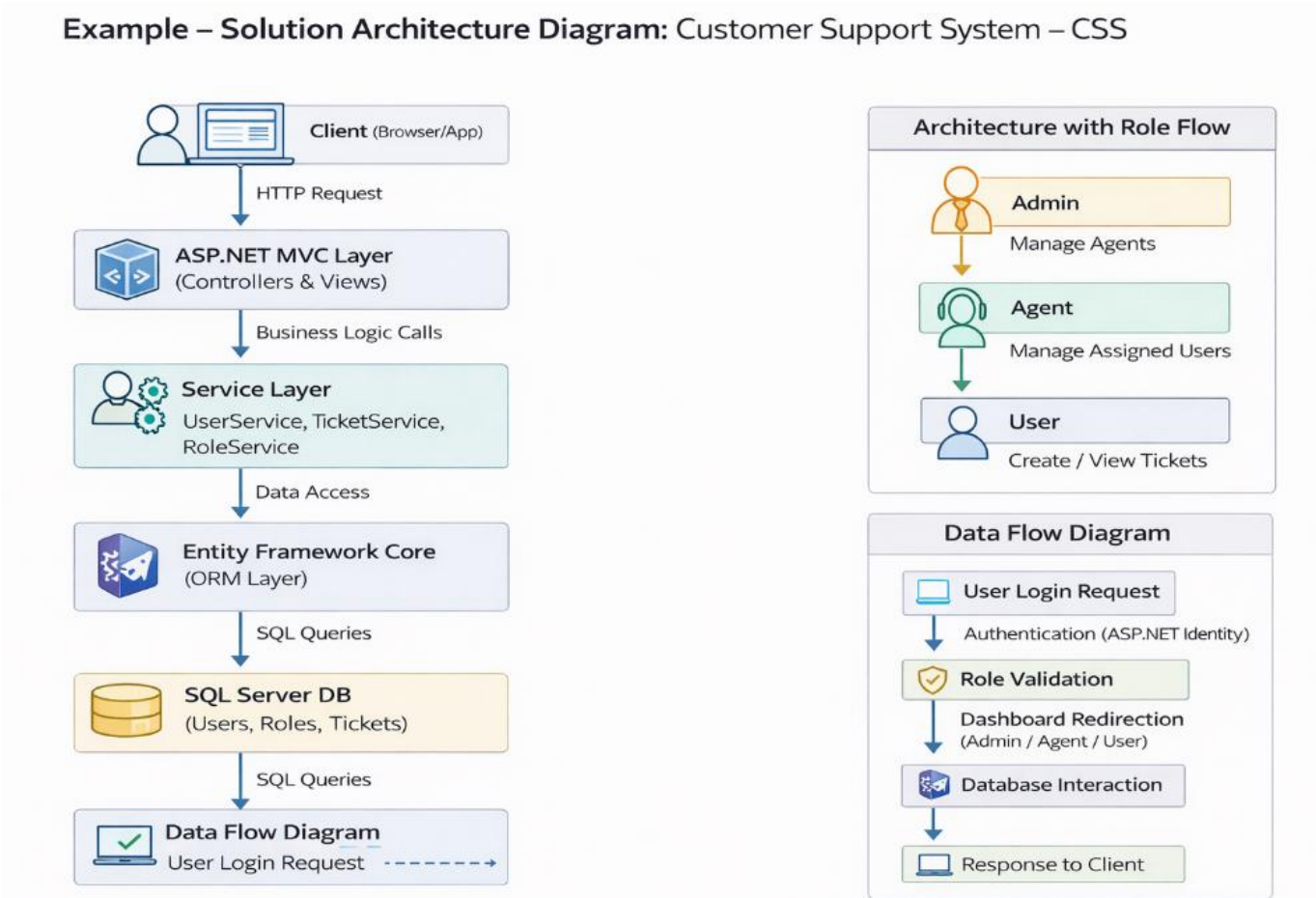
1. Overview

The proposed solution follows a **Cloud-Based 3-Tier Architecture** that separates:

1. Presentation Layer (User Interface)
2. Application Layer (Business Logic & AI Processing)
3. Data Layer (Database & Storage)

This architecture ensures scalability, performance, security, and maintainability.

2. High-Level Architecture Diagram



3. Architectural Components

A. Presentation Layer

Technology Used: Streamlit, HTML, CSS

Responsibilities:

- Accept artifact name input
- Accept word count selection
- Accept image upload
- Display generated description
- Show historical facts
- Provide regenerate option

B. Application Layer

Technology Used: Python

Sub-components:

1. Input Processing Module
    - Validates user inputs
    - Handles image preprocessing
2. Prompt Engineering Module
    - Structures prompt for AI model
    - Adds formatting instructions
3. AI Integration Module
    - Sends request to Gemini API
    - Receives generated output
4. Response Formatting Module
    - Organizes output into sections:
        - Origin
        - Time Period
        - Description
        - Historical Significance
        - Interesting Facts

C. AI / Machine Learning Layer

Model Used: Gemini 1.5 Flash (Multimodal AI)

Capabilities:

- Text-based generation
- Image-based artifact understanding
- Contextual historical explanation
- Structured content generation

D. Data Layer

1. User Database (SQLite / Cloud DB)
    - User credentials
    - Login information
2. Description History Storage
    - Previously generated artifact descriptions
3. Image Storage
    - Uploaded artifact images

4. Data Flow Explanation

1. User enters artifact name or uploads image.
2. Frontend sends request to backend.
3. Backend validates input and prepares AI prompt.
4. Request is sent to Gemini API.
5. AI model generates structured description.
6. Backend formats response.
7. Output is displayed to user.
8. Optional: Data is stored in database.

5. Architectural Characteristics

Cloud-Based Deployment
API-Driven AI Integration
Modular Design
Scalable Infrastructure
Secure Communication (HTTPS)
Separation of Concerns (3-Tier Model)

6. Development Phases

Phase 1: UI Development
Phase 2: Backend Logic Implementation
Phase 3: AI Integration
Phase 4: Database Integration
Phase 5: Testing & Deployment

7. Why This Architecture is Suitable

- Supports multimodal AI processing
- Easily scalable with cloud deployment

- Secure API communication
- Modular and maintainable
- High performance for real-time AI generation