<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>ISA-GS1: Interactive Project Plan</title>  
 <script src="https://cdn.tailwindcss.com"></script>  
 <!-- Visualization & Content Choices:  
 - Report Info: Project Directory Structure -> Goal: Understand file layout -> Viz/Presentation: HTML/JS interactive tree -> Interaction: Click file to see content in modal -> Justification: Clear, hierarchical view of project assets -> Library/Method: Vanilla JS DOM manipulation.  
 - Report Info: Config Files (.env, .yaml, .json) -> Goal: Explain settings -> Viz/Presentation: Syntax-highlighted code blocks -> Interaction: View content, (future: tooltips on specific lines) -> Justification: Direct presentation of configuration details -> Library/Method: HTML <pre>, JS for content loading.  
 - Report Info: CI/CD Workflows -> Goal: Understand automation steps -> Viz/Presentation: Visual flow diagram (HTML/CSS boxes & arrows) -> Interaction: Click step to see YAML snippet in modal -> Justification: Simplifies complex workflow into understandable stages -> Library/Method: HTML/CSS for diagram, JS for modal.  
 - Report Info: Code Skeletons (Python, Dockerfile, HTML) -> Goal: Show starting code structure -> Viz/Presentation: Syntax-highlighted code blocks -> Interaction: View content -> Justification: Provides concrete examples of initial code -> Library/Method: HTML <pre>, JS for content loading.  
 - Report Info: README.md content -> Goal: Provide project overview, setup, usage -> Viz/Presentation: Rendered HTML from Markdown-like structure -> Interaction: Scroll, read -> Justification: Standard way to present project documentation -> Library/Method: HTML, JS for injection.  
 - NO Chart.js/Plotly needed as the source report is about project structure/code, not quantitative data suitable for charts.  
 -->  
 <style>  
 body {  
 font-family: 'Inter', sans-serif;  
 }  
 @import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap');  
  
 .content-section {  
 display: none;  
 }  
 .content-section.active {  
 display: block;  
 }  
 .nav-link {  
 transition: all 0.3s ease;  
 }  
 .nav-link.active {  
 background-color: #0284c7; /\* sky-600 \*/  
 color: white;  
 }  
 .nav-link:hover {  
 background-color: #0369a1; /\* sky-700 \*/  
 color: white;  
 }  
 .code-block {  
 background-color: #1f2937; /\* gray-800 \*/  
 color: #d1d5db; /\* gray-300 \*/  
 padding: 1rem;  
 border-radius: 0.5rem;  
 overflow-x: auto;  
 font-family: 'Courier New', Courier, monospace;  
 font-size: 0.875rem;  
 line-height: 1.4;  
 }  
 .file-tree ul {  
 padding-left: 1.5rem;  
 border-left: 1px solid #e5e7eb; /\* gray-200 \*/  
 }  
 .file-tree li {  
 margin-bottom: 0.25rem;  
 }  
 .file-tree-item {  
 cursor: pointer;  
 padding: 0.25rem 0.5rem;  
 border-radius: 0.25rem;  
 display: inline-block;  
 }  
 .file-tree-item:hover {  
 background-color: #f3f4f6; /\* gray-100 \*/  
 }  
 .file-tree-item.folder::before {  
 content: '📁 '; /\* Default closed folder \*/  
 margin-right: 0.25rem;  
 }  
 .file-tree-item.folder.open::before {  
 content: '📂 '; /\* Open folder \*/  
 margin-right: 0.25rem;  
 }  
 .file-tree-item.file::before {  
 content: '📄 ';  
 margin-right: 0.25rem;  
 }  
 .file-content-modal {  
 position: fixed;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 100%;  
 background-color: rgba(0,0,0,0.5);  
 display: flex;  
 align-items: center;  
 justify-content: center;  
 z-index: 50;  
 padding: 1rem;  
 }  
 .file-content-modal-body {  
 background-color: white;  
 padding: 1.5rem;  
 border-radius: 0.5rem;  
 max-width: 90%;  
 max-height: 90vh;  
 overflow-y: auto;  
 width: 800px;  
 box-shadow: 0 10px 25px rgba(0,0,0,0.1);  
 }  
 .tooltip {  
 position: relative;  
 display: inline-block;  
 border-bottom: 1px dotted #6b7280; /\* gray-500 \*/  
 cursor: help;  
 }  
 .tooltip .tooltiptext {  
 visibility: hidden;  
 width: 250px;  
 background-color: #374151; /\* gray-700 \*/  
 color: #fff;  
 text-align: left;  
 border-radius: 6px;  
 padding: 8px;  
 position: absolute;  
 z-index: 1;  
 bottom: 125%;  
 left: 50%;  
 margin-left: -125px;  
 opacity: 0;  
 transition: opacity 0.3s;  
 font-size: 0.8rem;  
 line-height: 1.4;  
 }  
 .tooltip:hover .tooltiptext {  
 visibility: visible;  
 opacity: 1;  
 }  
 .workflow-step {  
 border: 1px solid #cbd5e1; /\* slate-300 \*/  
 padding: 0.75rem;  
 margin-bottom: 0.5rem;  
 border-radius: 0.375rem;  
 background-color: #f1f5f9; /\* slate-100 \*/  
 cursor: pointer;  
 transition: background-color 0.2s;  
 }  
 .workflow-step:hover {  
 background-color: #e2e8f0; /\* slate-200 \*/  
 }  
 .workflow-arrow {  
 text-align: center;  
 font-size: 1.5rem;  
 color: #64748b; /\* slate-500 \*/  
 margin: 0.25rem 0;  
 }  
 .readme-section h3 {  
 font-size: 1.25rem;  
 font-weight: 600;  
 margin-top: 1.5rem;  
 margin-bottom: 0.5rem;  
 border-bottom: 1px solid #e5e7eb; /\* gray-200 \*/  
 padding-bottom: 0.25rem;  
 color: #0f769e; /\* Slightly darker sky for headings \*/  
 }  
 .readme-section p, .readme-section ul {  
 margin-bottom: 1rem;  
 line-height: 1.6;  
 }  
 .readme-section ul {  
 list-style-type: disc;  
 padding-left: 1.5rem;  
 }  
 .readme-section code:not(pre > code) {  
 background-color: #e5e7eb; /\* gray-200 \*/  
 padding: 0.125rem 0.375rem;  
 border-radius: 0.25rem;  
 font-family: 'Courier New', Courier, monospace;  
 font-size: 0.9em;  
 }  
 .tab-button {  
 padding: 0.5rem 1rem;  
 cursor: pointer;  
 border: 1px solid transparent;  
 border-bottom: none;  
 border-radius: 0.375rem 0.375rem 0 0;  
 background-color: #e5e7eb; /\* gray-200 \*/  
 color: #374151; /\* gray-700 \*/  
 margin-right: 0.25rem;  
 font-weight: 500;  
 }  
 .tab-button.active {  
 background-color: white;  
 border-color: #d1d5db; /\* gray-300 \*/  
 border-bottom: 1px solid white;  
 color: #0284c7; /\* sky-600 \*/  
 font-weight: 600;  
 }  
 .tab-content {  
 display: none;  
 padding: 1.5rem;  
 border: 1px solid #d1d5db; /\* gray-300 \*/  
 border-top: none;  
 border-radius: 0 0 0.375rem 0.375rem;  
 background-color: white;  
 box-shadow: 0 1px 3px rgba(0,0,0,0.05);  
 }  
 .tab-content.active {  
 display: block;  
 }  
 </style>  
</head>  
<body class="bg-slate-50 text-slate-800">  
  
 <nav class="bg-sky-700 text-white p-4 shadow-md sticky top-0 z-40">  
 <div class="container mx-auto flex flex-wrap justify-center items-center">  
 <h1 class="text-2xl font-bold mr-6 whitespace-nowrap">ISA-GS1 Explorer</h1>  
 <div class="flex flex-wrap justify-center space-x-1 sm:space-x-2 mt-2 md:mt-0" id="navLinks">  
 <button data-target="overview" class="nav-link px-2 sm:px-3 py-2 rounded-md text-xs sm:text-sm font-medium">📜 Overview</button>  
 <button data-target="projectStructure" class="nav-link px-2 sm:px-3 py-2 rounded-md text-xs sm:text-sm font-medium">📁 Structure</button>  
 <button data-target="envConfig" class="nav-link px-2 sm:px-3 py-2 rounded-md text-xs sm:text-sm font-medium">⚙️ Env</button>  
 <button data-target="cloudDeploy" class="nav-link px-2 sm:px-3 py-2 rounded-md text-xs sm:text-sm font-medium">🚀 Deploy</button>  
 <button data-target="cicd" class="nav-link px-2 sm:px-3 py-2 rounded-md text-xs sm:text-sm font-medium">🔁 CI/CD</button>  
 <button data-target="backend" class="nav-link px-2 sm:px-3 py-2 rounded-md text-xs sm:text-sm font-medium">🧠 Backend</button>  
 <button data-target="frontend" class="nav-link px-2 sm:px-3 py-2 rounded-md text-xs sm:text-sm font-medium">🖥️ Frontend</button>  
 </div>  
 </div>  
 </nav>  
  
 <main class="container mx-auto p-4 md:p-8">  
 <section id="overview" class="content-section readme-section bg-white p-6 rounded-lg shadow">  
 <h2 class="text-3xl font-bold text-sky-700 mb-6">Project Overview & README</h2>  
 </section>  
  
 <section id="projectStructure" class="content-section">  
 <h2 class="text-3xl font-bold text-sky-700 mb-6">Project Directory Structure</h2>  
 <p class="mb-4 text-slate-600">Explore the conceptual file and directory layout for the ISA-GS1 project. Click on a file to view its skeleton content.</p>  
 <div class="bg-white p-6 rounded-lg shadow">  
 <div id="fileTreeView" class="file-tree"></div>  
 </div>  
 </section>  
  
 <section id="envConfig" class="content-section">  
 <h2 class="text-3xl font-bold text-sky-700 mb-6">Environment Configuration</h2>  
 <p class="mb-4 text-slate-600">This section shows the example environment configuration for the backend. These variables are crucial for connecting to various Google Cloud services.</p>  
 <div class="bg-white p-6 rounded-lg shadow">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>backend/.env.example</code></h3>  
 <pre id="envExampleContent" class="code-block"></pre>  
 </div>  
 </section>  
  
 <section id="cloudDeploy" class="content-section">  
 <h2 class="text-3xl font-bold text-sky-700 mb-6">Cloud Deployment & Configuration Files</h2>  
 <p class="mb-4 text-slate-600">Configuration files for deploying services to Cloud Run and Firebase, and for defining Firestore security rules.</p>  
 <div class="bg-white p-2 rounded-lg shadow">  
 <div id="cloudDeployTabs" class="mb-0 flex flex-wrap border-b border-gray-300 px-2 pt-2">  
 <button class="tab-button active" data-target="cloudRunYamlTab">cloud\_run.yaml</button>  
 <button class="tab-button" data-target="firebaseJsonTab">firebase.json</button>  
 <button class="tab-button" data-target="firebasercTab">.firebaserc</button>  
 <button class="tab-button" data-target="firestoreRulesTab">firestore.rules</button>  
 </div>  
 <div id="cloudRunYamlTab" class="tab-content active">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>config/cloud\_run.yaml</code></h3>  
 <pre id="cloudRunYamlContent" class="code-block"></pre>  
 </div>  
 <div id="firebaseJsonTab" class="tab-content">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>config/firebase.json</code></h3>  
 <pre id="firebaseJsonContent" class="code-block"></pre>  
 </div>  
 <div id="firebasercTab" class="tab-content">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>config/.firebaserc</code></h3>  
 <pre id="firebasercContent" class="code-block"></pre>  
 </div>  
 <div id="firestoreRulesTab" class="tab-content">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>config/firestore.rules</code></h3>  
 <pre id="firestoreRulesContent" class="code-block"></pre>  
 </div>  
 </div>  
 </section>  
  
 <section id="cicd" class="content-section">  
 <h2 class="text-3xl font-bold text-sky-700 mb-6">CI/CD Workflows (GitHub Actions)</h2>  
 <p class="mb-4 text-slate-600">Automated workflows for building and deploying the backend and frontend components. Click on a step to see the corresponding YAML snippet (conceptual).</p>  
 <div class="grid md:grid-cols-2 gap-6">  
 <div class="bg-white p-6 rounded-lg shadow">  
 <h3 class="text-xl font-semibold text-slate-700 mb-4">Backend Deployment Workflow</h3>  
 <div id="backendCicdFlow">  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Trigger">Trigger: Push to `main` (backend paths)</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Checkout">Checkout Code</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Auth GCP">Authenticate to Google Cloud (WIF)</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Setup GCloud">Set up Cloud SDK</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Configure Docker">Configure Docker for Artifact Registry</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Build Image">Build Docker Image</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Push Image">Push Image to Artifact Registry</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="backend\_deploy\_yml" data-step-name="Deploy CloudRun">Deploy to Cloud Run</div>  
 </div>  
 </div>  
 <div class="bg-white p-6 rounded-lg shadow">  
 <h3 class="text-xl font-semibold text-slate-700 mb-4">Frontend Deployment Workflow</h3>  
 <div id="frontendCicdFlow">  
 <div class="workflow-step" data-workflow-file="frontend\_deploy\_yml" data-step-name="Trigger">Trigger: Push to `main` (frontend paths)</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="frontend\_deploy\_yml" data-step-name="Checkout">Checkout Code</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="frontend\_deploy\_yml" data-step-name="Setup Node">Set up Node.js</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="frontend\_deploy\_yml" data-step-name="Install Deps">Install Frontend Dependencies</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="frontend\_deploy\_yml" data-step-name="Build App">Build Frontend Application</div>  
 <div class="workflow-arrow">↓</div>  
 <div class="workflow-step" data-workflow-file="frontend\_deploy\_yml" data-step-name="Deploy Firebase">Deploy to Firebase Hosting</div>  
 </div>  
 </div>  
 </div>  
 </section>  
  
 <section id="backend" class="content-section">  
 <h2 class="text-3xl font-bold text-sky-700 mb-6">Backend Skeletons (Python/FastAPI)</h2>  
 <p class="mb-4 text-slate-600">Initial code structure for the backend application, including Docker configuration, dependencies, and FastAPI application setup.</p>  
 <div class="bg-white p-2 rounded-lg shadow">  
 <div id="backendTabs" class="mb-0 flex flex-wrap border-b border-gray-300 px-2 pt-2">  
 <button class="tab-button active" data-target="dockerfileTab">Dockerfile</button>  
 <button class="tab-button" data-target="requirementsTab">requirements.txt</button>  
 <button class="tab-button" data-target="backendConfigTab">app/core/config.py</button>  
 <button class="tab-button" data-target="backendMainTab">app/main.py</button>  
 </div>  
 <div id="dockerfileTab" class="tab-content active">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>backend/Dockerfile</code></h3>  
 <pre id="dockerfileContent" class="code-block"></pre>  
 </div>  
 <div id="requirementsTab" class="tab-content">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>backend/requirements.txt</code></h3>  
 <pre id="requirementsContent" class="code-block"></pre>  
 </div>  
 <div id="backendConfigTab" class="tab-content">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>backend/app/core/config.py</code></h3>  
 <pre id="backendConfigContent" class="code-block"></pre>  
 </div>  
 <div id="backendMainTab" class="tab-content">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>backend/app/main.py</code></h3>  
 <pre id="backendMainContent" class="code-block"></pre>  
 </div>  
 </div>  
 </section>  
  
 <section id="frontend" class="content-section">  
 <h2 class="text-3xl font-bold text-sky-700 mb-6">Minimal Frontend Skeleton</h2>  
 <p class="mb-4 text-slate-600">A very basic HTML structure for the frontend. This would be replaced by a full Vue.js application in the actual project.</p>  
 <div class="bg-white p-6 rounded-lg shadow">  
 <h3 class="text-xl font-semibold text-slate-700 mb-3"><code>frontend/public/index.html</code></h3>  
 <pre id="frontendIndexContent" class="code-block html"></pre>  
 </div>  
 </section>  
  
 </main>  
  
 <footer class="bg-slate-800 text-slate-300 text-center p-6 mt-12">  
 <p>© ISA-GS1 Interactive Project Plan Explorer. For demonstration purposes.</p>  
 </footer>  
  
 <div id="fileContentModal" class="file-content-modal" style="display: none;">  
 <div class="file-content-modal-body">  
 <div class="flex justify-between items-center mb-4">  
 <h3 id="modalFileName" class="text-xl font-semibold text-sky-700">File Content</h3>  
 <button id="closeModalButton" class="text-slate-500 hover:text-slate-700 text-2xl font-bold">×</button>  
 </div>  
 <pre id="modalFileContent" class="code-block max-h-[70vh]"></pre>  
 </div>  
 </div>  
  
<script>  
 const projectData = {  
 envExampleContent: `  
# GCP Configuration  
PROJECT\_ID="your-gcp-project-id"  
REGION="europe-west1"  
SERVICE\_NAME="isa-api"  
GCP\_SERVICE\_ACCOUNT\_KEY\_PATH="/path/to/your/service-account-key.json" # For local dev; on Cloud Run, use runtime SA  
  
# Vertex AI Configuration  
GEMINI\_PRO\_MODEL\_NAME="gemini-1.5-pro-001"  
GEMINI\_FLASH\_MODEL\_NAME="gemini-1.5-flash-001"  
TEXT\_EMBEDDING\_MODEL\_NAME="text-embedding-preview-0409" # or textembedding-gecko@latest  
VECTOR\_SEARCH\_INDEX\_ID="your-isa-gs1-vector-index"  
VECTOR\_SEARCH\_INDEX\_ENDPOINT\_ID="your-isa-gs1-vector-endpoint"  
VECTOR\_SEARCH\_DEPLOYED\_INDEX\_ID="your\_deployed\_index\_id\_on\_endpoint" # Often needed for querying  
VECTOR\_SEARCH\_NAMESPACE="gs1\_docs\_live"  
  
# Firestore Configuration  
FIRESTORE\_CHUNK\_COLLECTION="gs1\_document\_chunks"  
FIRESTORE\_FEEDBACK\_COLLECTION="isa\_feedback"  
FIRESTORE\_USER\_HISTORY\_COLLECTION="isa\_user\_history"  
FIRESTORE\_KG\_TRIPLES\_COLLECTION="gs1\_kg\_triples" # If storing KG in Firestore  
  
# GCS Configuration  
GCS\_RAW\_DOCS\_BUCKET="gs1-isa-raw-documents-bucket"  
GCS\_KG\_BUCKET="gs1-isa-kg-data-bucket"  
GCS\_PROCESSED\_DOCS\_BUCKET="gs1-isa-processed-docs-bucket"  
  
# Firebase Project ID (if different from GCP Project ID, for Admin SDK)  
FIREBASE\_PROJECT\_ID="your-firebase-project-id" # Usually same as GCP\_PROJECT\_ID  
  
# API Settings  
API\_V1\_STR="/api/v1"  
# Add any other backend specific settings, e.g., for symbolic reasoner if it's a separate service  
 `.trim(),  
 cloudRunYamlContent: `  
apiVersion: serving.knative.dev/v1  
kind: Service  
metadata:  
 name: isa-api  
 namespace: your-gcp-project-id  
 annotations:  
 run.googleapis.com/ingress: all  
 run.googleapis.com/launch-stage: BETA  
spec:  
 template:  
 metadata:  
 annotations:  
 autoscaling.knative.dev/minScale: '0'  
 autoscaling.knative.dev/maxScale: '3'  
 run.googleapis.com/cpu-throttling: 'false'  
 spec:  
 containerConcurrency: 80  
 timeoutSeconds: 300  
 serviceAccountName: isa-cloud-run-sa@your-gcp-project-id.iam.gserviceaccount.com  
 containers:  
 - image: europe-west1-docker.pkg.dev/your-gcp-project-id/isa-repo/isa-api:latest  
 ports:  
 - name: http1  
 containerPort: 8000  
 env:  
 - name: PROJECT\_ID  
 valueFrom: { secretKeyRef: { name: isa-api-secrets, key: PROJECT\_ID } }  
 - name: REGION  
 valueFrom: { secretKeyRef: { name: isa-api-secrets, key: REGION } }  
 - name: GOOGLE\_APPLICATION\_CREDENTIALS  
 value: ""  
 resources:  
 limits: { cpu: "1000m", memory: "1Gi" }  
 startupProbe:  
 timeoutSeconds: 240  
 periodSeconds: 10  
 failureThreshold: 3  
 tcpSocket: { port: 8000 }  
 traffic:  
 - percent: 100  
 latestRevision: true  
 `.trim(),  
 firebaseJsonContent: `  
{  
 "hosting": {  
 "public": "frontend/dist",  
 "ignore": ["firebase.json", "\*\*/.\*", "\*\*/node\_modules/\*\*"],  
 "rewrites": [  
 {  
 "source": "/api/\*\*",  
 "run": { "serviceId": "isa-api", "region": "europe-west1" }  
 },  
 { "source": "\*\*", "destination": "/index.html" }  
 ],  
 "headers": [/\* ... headers ... \*/]  
 },  
 "emulators": { /\* ... emulators config ... \*/ }  
}  
 `.trim(),  
 firebasercContent: `  
{  
 "projects": {  
 "default": "your-firebase-project-id"  
 }  
}  
 `.trim(),  
 firestoreRulesContent: `  
rules\_version = '2';  
service cloud.firestore {  
 match /databases/{database}/documents {  
 function isAuthenticated() { return request.auth != null; }  
 function isOwner(userId) { return isAuthenticated() && request.auth.uid == userId; }  
  
 match /gs1\_document\_chunks/{chunkId} {  
 allow read: if isAuthenticated();  
 allow write: if false;  
 }  
 match /isa\_user\_history/{userId}/{historyId} {  
 allow read, write, delete: if isOwner(userId);  
 }  
 match /isa\_feedback/{feedbackId} {  
 allow create: if isAuthenticated();  
 allow read: if false;  
 }  
 match /gs1\_kg\_triples/{tripleId} {  
 allow read: if isAuthenticated();  
 allow write: if false;  
 }  
 }  
}  
 `.trim(),  
 dockerfileContent: `  
FROM python:3.11-slim  
WORKDIR /app  
ENV PYTHONDONTWRITEBYTECODE 1  
ENV PYTHONUNBUFFERED 1  
COPY requirements.txt .  
RUN pip install --no-cache-dir --upgrade pip  
RUN pip install --no-cache-dir -r requirements.txt  
COPY . .  
EXPOSE 8000  
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]  
 `.trim(),  
 requirementsContent: `  
fastapi>=0.110.0  
uvicorn[standard]>=0.29.0  
pydantic>=2.0  
pydantic-settings>=2.0  
python-dotenv>=1.0.0  
  
# Google Cloud Libraries  
google-cloud-aiplatform>=1.40.0  
google-cloud-firestore>=2.15.0  
google-cloud-storage>=2.14.0  
google-cloud-documentai>=1.25.0  
  
# KG & Symbolic Reasoning (examples)  
# rdflib>=7.0.0  
# z3-solver>=4.12.0  
 `.trim(),  
 backendConfigContent: `  
from pydantic\_settings import BaseSettings, SettingsConfigDict  
from functools import lru\_cache  
  
class Settings(BaseSettings):  
 PROJECT\_ID: str = "your-gcp-project-id"  
 REGION: str = "europe-west1"  
 # ... (other settings from .env.example) ...  
 VECTOR\_SEARCH\_INDEX\_ID: str # Make sure these are defined in .env or have defaults  
 VECTOR\_SEARCH\_INDEX\_ENDPOINT\_ID: str  
 VECTOR\_SEARCH\_DEPLOYED\_INDEX\_ID: str  
 API\_V1\_STR: str = "/api/v1"  
 model\_config = SettingsConfigDict(env\_file=".env", env\_file\_encoding='utf-8', extra='ignore')  
  
@lru\_cache()  
def get\_settings(): return Settings()  
settings = get\_settings()  
 `.trim(),  
 backendMainContent: `  
from fastapi import FastAPI, HTTPException, Depends  
from .core.config import settings  
from pydantic import BaseModel  
  
app = FastAPI(title="ISA API", version="0.1.0")  
  
class QueryRequest(BaseModel): query: str; user\_id: str | None = None  
class AnswerResponse(BaseModel): answer: str; sources: list[dict]; debug\_info: dict | None = None  
  
@app.on\_event("startup")  
async def startup\_event(): print(f"ISA API starting up for project: {settings.PROJECT\_ID}")  
  
@app.get("/")  
async def root(): return {"message": "Welcome to ISA-GS1 API"}  
  
@app.post(f"{settings.API\_V1\_STR}/query", response\_model=AnswerResponse)  
async def handle\_query(request: QueryRequest):  
 if not request.query: raise HTTPException(status\_code=400, detail="Query empty")  
 # --- Placeholder for actual RAG, Gemini, Symbolic logic ---  
 return AnswerResponse(  
 answer=f"Placeholder for: '{request.query}'",  
 sources=[{"document": "GS1 Gen Spec", "text\_snippet": "..."}],  
 debug\_info={"status": "dummy\_response"}  
 )  
# if \_\_name\_\_ == "\_\_main\_\_":   
# import uvicorn  
# uvicorn.run(app, host="0.0.0.0", port=8000)  
 `.trim(),  
 frontendIndexContent: `  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>ISA-GS1 - Intelligent Standards Assistant</title>  
 <style> /\* Basic styles from bundle \*/   
 body { font-family: sans-serif; margin: 0; background-color: #f4f7f6; display: flex; flex-direction: column; min-height: 100vh; }  
 header { background-color: #0073e6; color: white; padding: 1em; text-align: center; }  
 main { flex-grow: 1; max-width: 800px; margin: 2em auto; padding: 1em; background-color: white; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }  
 .input-area { display: flex; flex-direction: column; gap: 0.5em; margin-bottom: 1.5em; }  
 textarea { width: 100%; min-height: 80px; padding: 0.5em; border-radius: 4px; border: 1px solid #ccc; box-sizing: border-box; }  
 button { padding: 0.75em 1.5em; background-color: #0073e6; color: white; border: none; border-radius: 4px; cursor: pointer; font-size: 1em; }  
 button:hover { background-color: #005cb3; }  
 .response-area { margin-top: 1.5em; border-top: 1px solid #eee; padding-top: 1.5em; }  
 .response-area h3 { margin-top: 0; }  
 .response-text { white-space: pre-wrap; background-color: #e9ecef; padding: 1em; border-radius: 4px; font-family: monospace; }  
 .sources-list { list-style-type: none; padding-left: 0; }  
 .sources-list li { background-color: #f8f9fa; border: 1px solid #dee2e6; padding: 0.5em; margin-bottom: 0.5em; border-radius: 4px; font-size: 0.9em; }  
 footer { text-align: center; padding: 1em; background-color: #333; color: white; margin-top: auto; }  
 .loader { border: 4px solid #f3f3f3; border-top: 4px solid #0073e6; border-radius: 50%; width: 30px; height: 30px; animation: spin 1s linear infinite; margin: 1em auto; display: none; }  
 @keyframes spin { 0% { transform: rotate(0deg); } 100% { transform: rotate(360deg); } }  
 </style>  
</head>  
<body>  
 <header><h1>ISA-GS1</h1></header>  
 <main>  
 <div class="input-area">  
 <label for="queryInput">Ask ISA:</label>  
 <textarea id="queryInput" placeholder="e.g., What is a GTIN?"></textarea>  
 <button id="submitQuery">Ask ISA</button>  
 </div>  
 <div id="loader" class="loader" style="display:none;"></div>  
 <div id="responseArea" style="display:none;">  
 <h3>Answer:</h3><div id="responseText" class="response-text"></div>  
 <h4>Sources:</h4><ul id="sourcesList" class="sources-list"></ul>  
 <h4>Debug Info (Optional):</h4><pre id="debugInfo" class="response-text" style="font-size: 0.8em;"></pre>  
 </div>  
 <div id="errorArea" class="response-area" style="display:none; color: red;">  
 <h3>Error:</h3><div id="errorText" class="response-text"></div>  
 </div>  
 </main>  
 <footer><p>© 2025 ISA-GS1 Prototype</p></footer>  
 <script>  
 const queryInput = document.getElementById('queryInput');  
 const submitButton = document.getElementById('submitQuery');  
 const responseArea = document.getElementById('responseArea');  
 const responseText = document.getElementById('responseText');  
 const sourcesList = document.getElementById('sourcesList');  
 const debugInfo = document.getElementById('debugInfo');  
 const errorArea = document.getElementById('errorArea');  
 const errorText = document.getElementById('errorText');  
 const loader = document.getElementById('loader');  
  
 async function getAuthToken() { return null; }  
  
 submitButton.addEventListener('click', async () => {  
 const query = queryInput.value.trim();  
 if (!query) { alert('Please enter a query.'); return; }  
 loader.style.display = 'block';  
 responseArea.style.display = 'none';  
 errorArea.style.display = 'none';  
 try {  
 const token = await getAuthToken();  
 const headers = { 'Content-Type': 'application/json' };  
 if (token) { headers['Authorization'] = \`Bearer \${token}\`; }  
 const res = await fetch('/api/v1/query', {  
 method: 'POST', headers: headers, body: JSON.stringify({ query: query })  
 });  
 loader.style.display = 'none';  
 if (!res.ok) {  
 const errorData = await res.json().catch(() => ({ detail: "Unknown error." }));  
 throw new Error(\`API Error (\${res.status}): \${errorData.detail || JSON.stringify(errorData)}\`);  
 }  
 const data = await res.json();  
 responseText.textContent = data.answer;  
 sourcesList.innerHTML = '';   
 if (data.sources && data.sources.length > 0) {  
 data.sources.forEach(source => {  
 const li = document.createElement('li');  
 li.textContent = \`Doc: \${source.document}, Sec: \${source.section}, Page: \${source.page} - Snippet: "\${source.text\_snippet ? source.text\_snippet.substring(0,100) : ''}..."\`;  
 sourcesList.appendChild(li);  
 });  
 } else { /\* ... no sources ... \*/ }  
 debugInfo.textContent = data.debug\_info ? JSON.stringify(data.debug\_info, null, 2) : 'N/A';  
 responseArea.style.display = 'block';  
 } catch (err) {  
 loader.style.display = 'none';  
 errorText.textContent = err.message;  
 errorArea.style.display = 'block';  
 }  
 });  
 <\/script>  
</body>  
</html>  
 `.trim(),  
 readmeContent: `  
<div class="readme-section">  
  
<h3># ISA-GS1: Intelligent Standards Assistant</h3>  
<p>ISA-GS1 is an AI-powered assistant designed to help experts interpret and work with GS1 global data standards. It leverages Google Cloud services including Vertex AI (Gemini, Embeddings, Vector Search), Cloud Run, Firebase, and Firestore.</p>  
  
<h3>## Features (Target)</h3>  
<ul>  
 <li>Interpret and explain GS1 standards (GTIN, GLN, EPCIS, GDSN, Digital Link).</li>  
 <li>Provide traceable, explainable answers using AI (RAG) and logic.</li>  
 <li>Support symbolic reasoning for rule validation.</li>  
 <li>Integrate with a Knowledge Graph of GS1 concepts.</li>  
 <li>Highly usable by a non-technical solo expert.</li>  
</ul>  
  
<h3>## Tech Stack</h3>  
<ul>  
 <li><strong>Frontend</strong>: Vue.js 3 (Target), Firebase Hosting</li>  
 <li><strong>Backend</strong>: Python (FastAPI), Cloud Run</li>  
 <li><strong>AI/ML</strong>: Vertex AI (Gemini, Embeddings API, Vector Search)</li>  
 <li><strong>Database</strong>: Firestore (metadata, user data, KG triples potentially)</li>  
 <li><strong>Storage</strong>: Google Cloud Storage (raw documents, KG files)</li>  
 <li><strong>CI/CD</strong>: GitHub Actions</li>  
 <li><strong>MLOps</strong>: Vertex AI Pipelines</li>  
</ul>  
  
<h3>## Prerequisites</h3>  
<ul>  
 <li>Google Cloud SDK (<code>gcloud</code> CLI) installed and authenticated.</li>  
 <li>Firebase CLI installed and authenticated.</li>  
 <li>Node.js and npm (for frontend development).</li>  
 <li>Python 3.10+ and pip (for backend development).</li>  
 <li>Docker (for building backend container).</li>  
 <li>Access to a GCP Project with billing enabled and necessary APIs (Vertex AI, Cloud Run, Firestore, Firebase, Artifact Registry, IAM, Secret Manager, etc.).</li>  
 <li>A Firebase Project linked to the GCP Project.</li>  
</ul>  
  
<h3>## Setup & Deployment</h3>  
  
<h4>### 1. Clone the Repository</h4>  
<pre class="code-block"><code>git clone <your-repo-url>  
cd ISA-GS1</code></pre>  
  
<h4>### 2. Configure Environment Variables</h4>  
<ul>  
 <li><strong>Backend</strong>: Copy <code>backend/.env.example</code> to <code>backend/.env</code> and fill in your GCP project details, API keys (if any for local dev), and service configurations. <em>For Cloud Run, it's recommended to use Secret Manager for sensitive environment variables.</em></li>  
 <li><strong>Frontend</strong>: If your Vue.js app requires build-time environment variables (e.g., Firebase config for client SDK), set them up as per Vite/Vue CLI conventions (e.g., <code>.env</code> files in <code>frontend/</code> prefixed with <code>VITE\_</code> or <code>VUE\_APP\_</code>).</li>  
</ul>  
  
<h4>### 3. Set up GCP & Firebase</h4>  
<ul>  
 <li>Create necessary Service Accounts (e.g., for Cloud Run runtime, GitHub Actions CI/CD). Grant them appropriate IAM roles.</li>  
 <li>Enable APIs in your GCP console: Vertex AI, Cloud Run, Firestore, Firebase, Artifact Registry, IAM, Secret Manager, Document AI, Cloud Storage.</li>  
 <li>Set up Artifact Registry to host your Docker images.</li>  
 <li>Initialize Firebase in your project: <code>firebase init hosting</code> (select your project, use <code>frontend/dist</code> as public dir), <code>firebase init firestore</code>.</li>  
 <li>Deploy Firestore rules: <code>firebase deploy --only firestore:rules</code>.</li>  
</ul>  
  
<h4>### 4. Backend Deployment (Cloud Run)</h4>  
<p>Details on building Docker image, pushing to Artifact Registry, and deploying to Cloud Run using <code>gcloud run services replace config/cloud\_run.yaml</code> or CI/CD.</p>  
  
<h4>### 5. Frontend Deployment (Firebase Hosting)</h4>  
<p>Details on installing dependencies, building the frontend, and deploying using <code>firebase deploy --only hosting</code> or CI/CD.</p>  
  
<h4>### 6. Data Ingestion & Indexing (MLOps - Vertex AI Pipelines)</h4>  
<p>Develop Kubeflow components, define Vertex AI Pipeline, upload initial documents, and trigger the pipeline.</p>  
  
<h3>## Local Development</h3>  
<p>Instructions for running backend (FastAPI with Uvicorn) and frontend (Vue.js dev server) locally, and using Firebase Emulators.</p>  
  
<h3>## Contributing</h3>  
<p>Details on contributing, coding standards, and PR process will be added here.</p>  
  
<h3>## License</h3>  
<p>Specify your project license (e.g., MIT, Apache 2.0).</p>  
</div>  
 `.trim(),  
 projectStructure: {  
 name: 'ISA-GS1', type: 'folder', children: [  
 { name: '.github', type: 'folder', children: [  
 { name: 'workflows', type: 'folder', children: [  
 { name: 'backend\_deploy.yml', type: 'file', contentKey: 'backend\_deploy\_yml' },  
 { name: 'frontend\_deploy.yml', type: 'file', contentKey: 'frontend\_deploy\_yml' }  
 ]}  
 ]},  
 { name: 'backend', type: 'folder', children: [  
 { name: 'app', type: 'folder', children: [  
 { name: '\_\_init\_\_.py', type: 'file', content: '# backend/app/\_\_init\_\_.py' },  
 { name: 'main.py', type: 'file', contentKey: 'backendMainContent' },  
 { name: 'routers', type: 'folder', children: [  
 { name: '\_\_init\_\_.py', type: 'file', content: '# backend/app/routers/\_\_init\_\_.py' },  
 { name: 'qa.py', type: 'file', content: '# backend/app/routers/qa.py\n# Placeholder for Q&A endpoint logic' }  
 ]},  
 { name: 'core', type: 'folder', children: [  
 { name: '\_\_init\_\_.py', type: 'file', content: '# backend/app/core/\_\_init\_\_.py' },  
 { name: 'config.py', type: 'file', contentKey: 'backendConfigContent' }  
 ]},  
 { name: 'services', type: 'folder', children: [  
 { name: '\_\_init\_\_.py', type: 'file', content: '# backend/app/services/\_\_init\_\_.py' },  
 { name: 'rag\_service.py', type: 'file', content: '# backend/app/services/rag\_service.py\n# Placeholder for RAG logic' },  
 { name: 'gemini\_service.py', type: 'file', content: '# backend/app/services/gemini\_service.py\n# Placeholder for Gemini interaction' },  
 { name: 'symbolic\_service.py', type: 'file', content: '# backend/app/services/symbolic\_service.py\n# Placeholder for symbolic reasoning' }  
 ]},  
 { name: 'models', type: 'folder', children: [  
 { name: '\_\_init\_\_.py', type: 'file', content: '# backend/app/models/\_\_init\_\_.py' },  
 { name: 'qa\_models.py', type: 'file', content: '# backend/app/models/qa\_models.py\n# Pydantic models for Q&A' }  
 ]}  
 ]},  
 { name: 'Dockerfile', type: 'file', contentKey: 'dockerfileContent' },  
 { name: 'requirements.txt', type: 'file', contentKey: 'requirementsContent' },  
 { name: '.env.example', type: 'file', contentKey: 'envExampleContent' }  
 ]},  
 { name: 'frontend', type: 'folder', children: [  
 { name: 'public', type: 'folder', children: [  
 { name: 'index.html', type: 'file', contentKey: 'frontendIndexContent' }  
 ]},  
 { name: 'src', type: 'folder', children: [  
 { name: 'App.vue', type: 'file', content: '' },  
 { name: 'main.js', type: 'file', content: '// frontend/src/main.js (Vue App Initialization)' },  
 { name: 'components', type: 'folder', children: [] },  
 { name: 'views', type: 'folder', children: [] },  
 { name: 'router', type: 'folder', children: [] },  
 { name: 'store', type: 'folder', children: [] }  
 ]},  
 { name: 'package.json', type: 'file', content: '{ "name": "isa-frontend", "version": "0.1.0", "scripts": { "dev": "vite", "build": "vite build" } }' },  
 { name: 'vite.config.js', type: 'file', content: '// frontend/vite.config.js (or vue.config.js)' },  
 { name: 'tailwind.config.js', type: 'file', content: '// frontend/tailwind.config.js' }  
 ]},  
 { name: 'scripts', type: 'folder', children: [  
 { name: 'data\_ingestion', type: 'folder', children: [  
 { name: 'run\_pipeline.py', type: 'file', content: '# scripts/data\_ingestion/run\_pipeline.py' },  
 { name: 'components', type: 'folder', children: [  
 { name: 'doc\_parser.py', type: 'file', content: '# scripts/data\_ingestion/components/doc\_parser.py' }  
 ]}  
 ]},  
 { name: 'kg\_builder', type: 'folder', children: [  
 { name: 'build\_rdf.py', type: 'file', content: '# scripts/kg\_builder/build\_rdf.py' }  
 ]}  
 ]},  
 { name: 'docs', type: 'folder', children: [  
 { name: 'ISA\_System\_Design.md', type: 'file', content: '# ISA\_System\_Design.md\nThis would contain the full system design document.' }  
 ]},  
 { name: 'config', type: 'folder', children: [  
 { name: 'cloud\_run.yaml', type: 'file', contentKey: 'cloudRunYamlContent' },  
 { name: 'firebase.json', type: 'file', contentKey: 'firebaseJsonContent' },  
 { name: '.firebaserc', type: 'file', contentKey: 'firebasercContent' },  
 { name: 'firestore.rules', type: 'file', contentKey: 'firestoreRulesContent' }  
 ]},  
 { name: 'README.md', type: 'file', contentKey: 'readmeContentRaw' },  
 { name: '.gitignore', type: 'file', content: '# .gitignore\nvenv/\n\_\_pycache\_\_/\n\*.pyc\n.env\nnode\_modules/\ndist/' }  
 ]  
 },  
 readmeContentRaw: `... (This will be replaced by the actual readmeContent below) ...`,  
 backend\_deploy\_yml: `  
name: Deploy Backend to Cloud Run  
on:  
 push:  
 branches: [main]  
 paths: ['backend/\*\*', '.github/workflows/backend\_deploy.yml']  
env:  
 GCP\_PROJECT\_ID: your-gcp-project-id  
 GCP\_ARTIFACT\_REGISTRY\_REGION: europe-west1  
 GCP\_CLOUD\_RUN\_REGION: europe-west1  
 SERVICE\_NAME: isa-api  
 IMAGE\_NAME: isa-api  
jobs:  
 build-and-deploy:  
 runs-on: ubuntu-latest  
 permissions: { contents: 'read', id-token: 'write' }  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v4  
 - name: Authenticate to Google Cloud (WIF)  
 uses: google-github-actions/auth@v2  
 with:  
 workload\_identity\_provider: 'projects/YOUR\_GCP\_PROJECT\_NUMBER/locations/global/workloadIdentityPools/YOUR\_POOL\_NAME/providers/YOUR\_PROVIDER\_NAME'  
 service\_account: 'github-actions-sa@your-gcp-project-id.iam.gserviceaccount.com'  
 - name: Set up Cloud SDK  
 uses: google-github-actions/setup-gcloud@v2  
 - name: Configure Docker  
 run: gcloud auth configure-docker \${{ env.GCP\_ARTIFACT\_REGISTRY\_REGION }}-docker.pkg.dev  
 - name: Build Docker image  
 run: docker build -t \${{ env.GCP\_ARTIFACT\_REGISTRY\_REGION }}-docker.pkg.dev/\${{ env.GCP\_PROJECT\_ID }}/\${{ env.IMAGE\_NAME }}/\${{ env.IMAGE\_NAME }}:\${{ github.sha }} ./backend  
 - name: Push Docker image  
 run: docker push \${{ env.GCP\_ARTIFACT\_REGISTRY\_REGION }}-docker.pkg.dev/\${{ env.GCP\_PROJECT\_ID }}/\${{ env.IMAGE\_NAME }}/\${{ env.IMAGE\_NAME }}:\${{ github.sha }}  
 - name: Deploy to Cloud Run  
 uses: google-github-actions/deploy-cloudrun@v2  
 with:  
 service: \${{ env.SERVICE\_NAME }}  
 region: \${{ env.GCP\_CLOUD\_RUN\_REGION }}  
 image: \${{ env.GCP\_ARTIFACT\_REGISTRY\_REGION }}-docker.pkg.dev/\${{ env.GCP\_PROJECT\_ID }}/\${{ env.IMAGE\_NAME }}/\${{ env.IMAGE\_NAME }}:\${{ github.sha }}  
 flags: '--allow-unauthenticated --platform managed'  
 `.trim(),  
 frontend\_deploy\_yml: `  
name: Deploy Frontend to Firebase Hosting  
on:  
 push:  
 branches: [main]  
 paths: ['frontend/\*\*', '.github/workflows/frontend\_deploy.yml']  
jobs:  
 build-and-deploy:  
 runs-on: ubuntu-latest  
 steps:  
 - name: Checkout code  
 uses: actions/checkout@v4  
 - name: Set up Node.js  
 uses: actions/setup-node@v4  
 with: { node-version: '20' }  
 - name: Install frontend dependencies  
 run: npm ci  
 working-directory: ./frontend  
 - name: Build frontend application  
 run: npm run build  
 working-directory: ./frontend  
 env:  
 VITE\_FIREBASE\_API\_KEY: \${{ secrets.FIREBASE\_API\_KEY }}  
 VITE\_FIREBASE\_PROJECT\_ID: your-firebase-project-id  
 - name: Deploy to Firebase Hosting  
 uses: FirebaseExtended/action-hosting-deploy@v0  
 with:  
 repoToken: '\${{ secrets.GITHUB\_TOKEN }}'  
 firebaseServiceAccount: '\${{ secrets.FIREBASE\_SERVICE\_ACCOUNT\_KEY\_JSON }}'  
 projectId: your-firebase-project-id  
 `.trim()  
 };  
 projectData.readmeContentRaw = projectData.readmeContent;  
  
 document.addEventListener('DOMContentLoaded', () => {  
 const navLinksContainer = document.getElementById('navLinks');  
 const contentSections = document.querySelectorAll('.content-section');  
 const navLinks = navLinksContainer.querySelectorAll('.nav-link');  
  
 function setActiveSection(targetId) {  
 contentSections.forEach(section => {  
 section.classList.toggle('active', section.id === targetId);  
 });  
 navLinks.forEach(link => {  
 link.classList.toggle('active', link.dataset.target === targetId);  
 });  
 window.scrollTo(0, 0);  
 }  
  
 navLinksContainer.addEventListener('click', (event) => {  
 if (event.target.matches('.nav-link')) {  
 const targetId = event.target.dataset.target;  
 setActiveSection(targetId);  
 }  
 });  
  
 document.getElementById('overview').insertAdjacentHTML('beforeend', projectData.readmeContent);  
 document.getElementById('envExampleContent').textContent = projectData.envExampleContent;  
 document.getElementById('cloudRunYamlContent').textContent = projectData.cloudRunYamlContent;  
 document.getElementById('firebaseJsonContent').textContent = projectData.firebaseJsonContent;  
 document.getElementById('firebasercContent').textContent = projectData.firebasercContent;  
 document.getElementById('firestoreRulesContent').textContent = projectData.firestoreRulesContent;  
 document.getElementById('dockerfileContent').textContent = projectData.dockerfileContent;  
 document.getElementById('requirementsContent').textContent = projectData.requirementsContent;  
 document.getElementById('backendConfigContent').textContent = projectData.backendConfigContent;  
 document.getElementById('backendMainContent').textContent = projectData.backendMainContent;  
 document.getElementById('frontendIndexContent').textContent = projectData.frontendIndexContent;  
  
 const fileTreeView = document.getElementById('fileTreeView');  
 const modal = document.getElementById('fileContentModal');  
 const modalFileName = document.getElementById('modalFileName');  
 const modalFileContent = document.getElementById('modalFileContent');  
 const closeModalButton = document.getElementById('closeModalButton');  
  
 function createFileTree(parentElement, items) {  
 const ul = document.createElement('ul');  
 items.forEach(item => {  
 const li = document.createElement('li');  
 const span = document.createElement('span');  
 span.textContent = item.name;  
 span.classList.add('file-tree-item', item.type);  
   
 if (item.type === 'file') {  
 span.addEventListener('click', () => {  
 modalFileName.textContent = item.name;  
 let fileContent = `// Content for ${item.name} is conceptual or not fully shown.`;  
 if (item.contentKey) {  
 fileContent = projectData[item.contentKey] || fileContent;  
 } else if (item.content) {  
 fileContent = item.content;  
 }  
 modalFileContent.textContent = fileContent;  
 modal.style.display = 'flex';  
 });  
 } else if (item.type === 'folder' && item.children) {  
 const subUl = createFileTree(li, item.children);  
 subUl.style.display = 'none';   
 span.classList.add('folder');   
 span.addEventListener('click', (e) => {  
 e.stopPropagation();   
 const isOpen = subUl.style.display === 'block';  
 subUl.style.display = isOpen ? 'none' : 'block';  
 span.classList.toggle('open', !isOpen);  
 });  
 }  
 li.appendChild(span);  
 ul.appendChild(li);  
 });  
 parentElement.appendChild(ul);  
 return ul;  
 }  
 createFileTree(fileTreeView, projectData.projectStructure.children);  
   
 closeModalButton.addEventListener('click', () => { modal.style.display = 'none'; });  
 modal.addEventListener('click', (e) => {  
 if (e.target === modal) { modal.style.display = 'none'; }  
 });  
  
 function setupTabs(tabsContainerId) {  
 const tabsContainer = document.getElementById(tabsContainerId);  
 if (!tabsContainer) { return; }  
  
 const tabButtons = tabsContainer.querySelectorAll('.tab-button');  
 const tabContentsWrapper = tabsContainer.nextElementSibling.classList.contains('tab-content') ?   
 tabsContainer.nextElementSibling.parentElement : tabsContainer.parentElement;  
   
 const tabContents = Array.from(tabContentsWrapper.children).filter(child => child.classList.contains('tab-content'));  
  
  
 if (tabButtons.length === 0 || tabContents.length === 0 && tabsContainerId !== 'cloudDeployTabs' && tabsContainerId !== 'backendTabs') {  
 // The above condition for tabContents.length was too restrictive for the new structure  
 }  
  
  
 tabButtons.forEach(button => {  
 button.addEventListener('click', () => {  
 tabButtons.forEach(btn => btn.classList.remove('active'));  
 button.classList.add('active');  
 const targetContentId = button.dataset.target;  
   
 let foundTarget = false;  
 tabContents.forEach(content => {  
 const isActive = content.id === targetContentId;  
 content.classList.toggle('active', isActive);  
 if(isActive) foundTarget = true;  
 });  
 // Fallback for cases where tabContents might not be direct siblings or structured differently  
 if (!foundTarget) {  
 const directTarget = document.getElementById(targetContentId);  
 if (directTarget && directTarget.classList.contains('tab-content')) {  
 Array.from(directTarget.parentElement.children)  
 .filter(c => c.classList.contains('tab-content'))  
 .forEach(tc => tc.classList.remove('active'));  
 directTarget.classList.add('active');  
 }  
 }  
 });  
 });  
 }  
  
 setupTabs('cloudDeployTabs');  
 setupTabs('backendTabs');  
   
 document.querySelectorAll('.workflow-step').forEach(step => {  
 step.addEventListener('click', () => {  
 const workflowFileKey = step.dataset.workflowFile;  
 const stepName = step.dataset.stepName;  
 modalFileName.textContent = `${workflowFileKey} - Step: ${stepName}`;  
 let contentToShow = projectData[workflowFileKey] || `// Content for ${workflowFileKey} (step: ${stepName}) is conceptual.`;  
 modalFileContent.textContent = contentToShow;  
 modal.style.display = 'flex';  
 });  
 });  
  
 setActiveSection('overview');  
 });  
</script>  
</body>  
</html>