**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID52055 |
| Project Name | Heritage Treasures: An in-depth analysis of UNESCO World Heritage Sites In Tableau |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

**┌─────────────────────────────────────────────────────────────────────┐**

**│ USER LAYER │**

**├─────────────────┬─────────────────┬─────────────────┬───────────────┤**

**│ Researchers │ Policy Makers │ Heritage Orgs │ Tourists │**

**└─────────────────┴─────────────────┴─────────────────┴───────────────┘**

**│**

**▼**

**┌─────────────────────────────────────────────────────────────────────┐**

**│ PRESENTATION LAYER │**

**├─────────────────────────────────────────────────────────────────────┤**

**│ Tableau Desktop │ Tableau Server │ Web Interface │ Mobile App │**

**└─────────────────────────────────────────────────────────────────────┘**

**│**

**▼**

**┌─────────────────────────────────────────────────────────────────────┐**

**│ APPLICATION LAYER │**

**├─────────────────────────────────────────────────────────────────────┤**

**│ Data Processing │ Analytics Engine │ Export Service │ Auth Service │**

**│ (Python) │ (Tableau) │ (Node.js) │ (OAuth) │**

**└─────────────────────────────────────────────────────────────────────┘**

**│**

**▼**

**┌─────────────────────────────────────────────────────────────────────┐**

**│ DATA LAYER │**

**├─────────────────────────────────────────────────────────────────────┤**

**│ UNESCO API │ PostgreSQL DB │ File Storage │ Cache (Redis) │**

**│ (External) │ (Heritage Data) │ (AWS S3) │ (Performance) │**

**└─────────────────────────────────────────────────────────────────────┘**

**Table-1: Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| **1.** | **User Interface** | Interactive dashboard interface for data visualization and analysis | Tableau Desktop/Server, HTML5, CSS3, JavaScript |
| **2.** | **Data Ingestion** | Automated data extraction and processing from UNESCO sources | Python with pandas, requests libraries, Apache Airflow |
| **3.** | **Data Processing** | Data cleaning, transformation, and validation logic | Python (pandas, numpy), SQL, Tableau Prep |
| **4.** | **Analytics Engine** | Core visualization and analytical processing | Tableau Desktop/Server, R integration, Statistical functions |
| **5.** | **Web Application** | Web-based interface for dashboard access and user management | Node.js, Express.js, React.js for admin portal |
| **6.** | **Database** | Primary data storage for heritage site information | PostgreSQL with PostGIS extension for geographic data |
| **7.** | **Cache Layer** | Performance optimization for frequently accessed data | Redis for caching, Tableau Server cache |
| **8.** | **File Storage** | Storage for exported reports, images, and backup data | Amazon S3 or Azure Blob Storage |
| **9.** | **External API** | UNESCO World Heritage API for real-time data updates | UNESCO API, REST services |
| **10.** | **Authentication** | User authentication and authorization system | OAuth 2.0, JWT tokens, Active Directory integration |
| **11.** | **Infrastructure** | Cloud-based hosting and deployment platform | AWS EC2/Azure VMs, Docker containers, Kubernetes |
| **12.** | **Monitoring** | System monitoring and performance tracking | Tableau Server monitoring, CloudWatch, Grafana |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| **1.** | **Open-Source Frameworks** | Utilizes open-source components for cost-effectiveness and flexibility | Python (pandas, numpy), PostgreSQL, Redis, Apache Airflow, Node.js, React.js |
| **2.** | **Security Implementations** | Comprehensive security measures for data protection and access control | - SSL/TLS encryption for data in transit<br>- AES-256 encryption for data at rest<br>- OAuth 2.0 and SAML for authentication<br>- Role-based access control (RBAC)<br>- API rate limiting and token-based security<br>- Regular security audits and penetration testing |
| **3.** | **Scalable Architecture** | Three-tier architecture supporting horizontal and vertical scaling | - Presentation Layer: Tableau Server cluster<br>- Application Layer: Microservices with Docker containers<br>- Data Layer: PostgreSQL with read replicas<br>- Load balancing with NGINX<br>- Auto-scaling groups in AWS/Azure |
| **4.** | **Availability** | High availability design with redundancy and failover capabilities | - Multi-zone deployment across AWS/Azure regions<br>- Load balancers with health checks<br>- Database clustering with automatic failover<br>- CDN integration (CloudFront/Azure CDN)<br>- 99.5% uptime SLA with disaster recovery |
| **5.** | **Performance** | Optimized performance for handling large datasets and concurrent users | - In-memory caching with Redis<br>- Database indexing and query optimization<br>- Tableau Server performance tuning<br>- CDN for static content delivery<br>- Asynchronous processing for data updates<br>- Support for 100+ concurrent users |

**Detailed Technology Stack:**

**Frontend Layer:**

* Tableau Desktop/Server: Primary visualization platform
* HTML5/CSS3/JavaScript: Custom web interfaces
* React.js: Admin portal and user management
* Bootstrap: Responsive UI framework

**Backend Layer:**

* Python: Data processing and ETL operations
* Node.js: Web application backend
* Express.js: RESTful API framework
* Apache Airflow: Workflow orchestration

**Database Layer:**

* PostgreSQL: Primary relational database
* PostGIS: Geographic data extension
* Redis: Caching and session storage

**Integration Layer:**

* REST APIs: Service communication
* UNESCO API: External data source
* OAuth 2.0: Authentication protocol

**Infrastructure Layer:**

* AWS/Azure: Cloud hosting platform
* Docker: Containerization
* Kubernetes: Container orchestration
* NGINX: Load balancing and reverse proxy

**Monitoring & DevOps:**

* CloudWatch/Azure Monitor: Infrastructure monitoring
* Grafana: Dashboard monitoring
* Git: Version control
* Jenkins/GitHub Actions: CI/CD pipeline

**Deployment Architecture:**

**Production Environment:**

**├── Load Balancer (NGINX)**

**├── Tableau Server Cluster (3 nodes)**

**├── Application Servers (Auto-scaling group)**

**├── Database Cluster (Master + 2 Replicas)**

**├── Redis Cluster (3 nodes)**

**└── File Storage (S3/Azure Blob)**

**Development Environment:**

**├── Single Tableau Server instance**

**├── Application Server (Docker)**

**├── PostgreSQL Database**

**├── Redis Cache**

**└── Local file storage**

**References:**

* [**C4 Model for Architecture**](https://c4model.com/)
* [**Tableau Architecture Guide**](https://help.tableau.com/current/server/en-us/server_architecture.htm)
* [**AWS Well-Architected Framework**](https://aws.amazon.com/architecture/well-architected/)
* [**Azure Architecture Center**](https://docs.microsoft.com/en-us/azure/architecture/)
* [**PostgreSQL High Availability**](https://www.postgresql.org/docs/current/high-availability.html)