

**Project Design Phase-II**  
**Technology Stack (Architecture & Stack)**

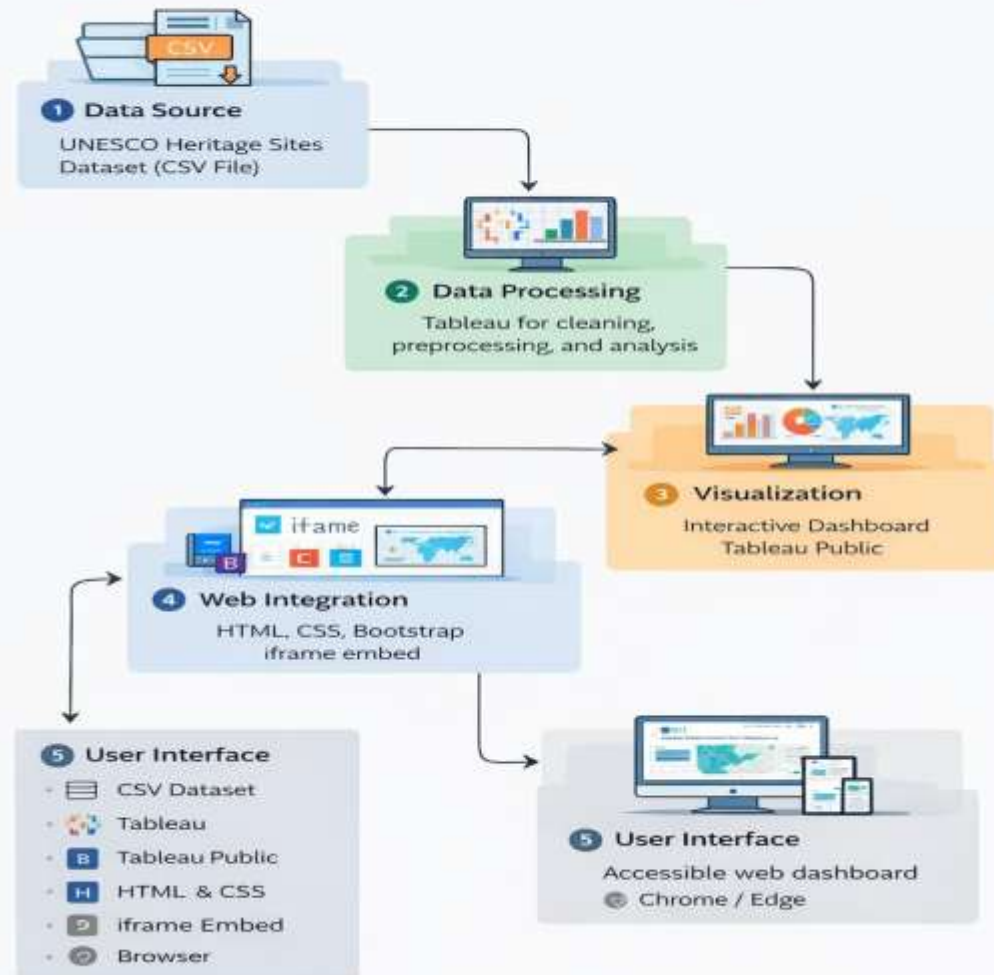
Date	10 February 2026
Team ID	LTVIP2026TMIDS91063
Project Name	Heritage Treasures: An In-Depth Analysis of UNESCO World Heritage Sites in Tableau
Maximum Marks	4 Marks

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

# Technical Architecture

## UNESCO World Heritage Sites Dashboard



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

S.No	Component	Description	Technology
1.	User Interface	Web interface through which users view dashboards, stories, maps, and insights	HTML, CSS, JavaScript, Bootstrap
2.	Frontend Logic	Handles page navigation, UI interactions, responsiveness, and animations	JavaScript, Bootstrap JS
3.	Web Framework	Connects frontend with backend and manages routing and page rendering	Flask (Python)
4.	Application Logic	Controls application flow and serves embedded dashboards and static content	Python (Flask)
5.	Data Visualization Engine	Creates interactive charts, maps, dashboards, and stories	Tableau Public
6.	Data Processing Layer	Performs data cleaning, aggregation, filtering, and calculations	Tableau Calculated Fields
7.	Dataset	Stores UNESCO World Heritage data such as country, region, category, year, and danger status	CSV / Excel Dataset
8.	Embedding Layer	Embeds Tableau dashboards and stories into web pages	Tableau Embed
9.	Static Resource Management	Stores CSS, JS, images, icons, and template assets	Flask Static Folder
10.	Infrastructure (Server / Cloud)	Hosts and runs the Flask web application	Local System / Cloud Hosting

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Uses open-source frameworks and libraries for building the web interface and backend server	Flask (Python), Bootstrap, JavaScript
2.	Security Implementations	No user authentication is required. The application follows basic web security practices such as secure embedding and safe routing	HTTPS, Secure Embedding
3.	Scalable Architecture	Follows a layered architecture where frontend, backend, and visualization layers are loosely coupled, allowing easy scalability	Flask Backend, Tableau Public
4.	Availability	Application is accessible anytime as dashboards are hosted on Tableau Public and served through a Flask web server	Tableau Public Hosting, Flask Server
5.	Performance	Optimized by using pre-aggregated data in Tableau dashboards and lightweight frontend rendering	Tableau Optimized Views, Bootstrap