Project Design Phase-II

Technology Stack (Architecture & Stack)

Date: June 28, 2025

Team ID: LTVIP2025TMID50318

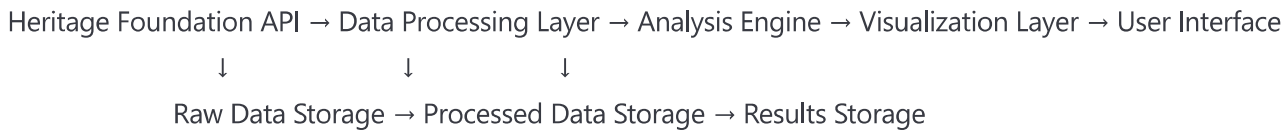
Project Name: Measuring the Pulse of Prosperity: An Index of Economic Freedom Analysis

Maximum Marks: 4 Marks

Technical Architecture

The deliverable includes the architectural diagram for the Economic Freedom Analysis system, showing the flow from data collection through analysis to visualization and reporting.

System Architecture Overview



Architecture Description

The Economic Freedom Analysis system follows a **layered architecture** approach with clear separation of concerns:

1. **Data Layer:** Heritage Foundation API integration and data storage
2. **Processing Layer:** Data normalization, validation, and transformation
3. **Analysis Layer:** Statistical computations, PCA, and index construction
4. **Visualization Layer:** Chart generation and dashboard creation
5. **Presentation Layer:** Web-based user interface and reporting

---

Table-1: Components & Technologies

S.No	Component	Description	Technology
1.	**User Interface**	Web-based dashboard for researchers, policymakers, and academics to interact with economic freedom data and analysis results	HTML5, CSS3, JavaScript, Bootstrap for responsive design
2.	**Data Collection Module**	Automated data retrieval from Heritage Foundation Economic Freedom Index API with error handling and validation	Python with requests library, pandas for data manipulation
3.	**Data Processing Engine**	Core logic for data normalization, cleaning, and transformation to 0-100 scale with missing value handling	Python with NumPy, pandas, scikit-learn
4.	**Statistical Analysis Module**	Implementation of Principal Component Analysis (PCA), correlation analysis, and regression modeling	Python with scikit-learn, scipy.stats, statsmodels
5.	**Index Construction Engine**	Calculation of Economic Freedom Composite Index (EFCI) using PCA-based weighting and aggregation algorithms	Python with NumPy, pandas
6.	**Database**	Storage of raw economic indicators, processed data, analysis results, and metadata	SQLite for development, PostgreSQL for production
7.	**File Storage**	Storage of generated reports, exported data files, and visualization assets	Local filesystem with organized directory structure
8.	**External API**	Heritage Foundation Economic Freedom Index API for accessing latest economic freedom indicators and country data	Heritage Foundation REST API
9.	**Visualization Engine**	Generation of interactive charts, scatter plots, heat maps, and statistical visualizations	Tableau Public, Python matplotlib, seaborn, plotly
10.	**Export Module**	Data export functionality supporting multiple formats for research and academic use	Python with pandas (CSV, Excel), reportlab (PDF)
11.	**Infrastructure**	**Local Development:** Windows/Linux system with Python 3.8+ **Cloud Deployment:** GitHub Pages for documentation, Tableau Public for dashboards	Local: Python development environment Cloud: GitHub, Tableau Public

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1.	**Open-Source Frameworks**	Utilizes multiple open-source libraries for data processing, statistical analysis, and visualization to ensure transparency and reproducibility	**Python:** pandas, NumPy, scikit-learn, matplotlib, seaborn **Web:** Bootstrap, jQuery **Database:** SQLite, PostgreSQL
2.	**Security Implementations**	Implements secure data handling practices including API key management, data validation, and secure file operations	**Data Security:** Input validation, SQL injection prevention **API Security:** Secure key storage, HTTPS connections **Access Control:** Read-only database access for public dashboards
3.	**Scalable Architecture**	Modular design allows for easy addition of new economic indicators, countries, or analytical methods without system redesign	**Modular Design:** Separate modules for data collection, processing, analysis **Component-based:** Independent processing pipelines **Configuration-driven:** External config files for easy modifications
4.	**Availability**	System designed for high availability with automated backup mechanisms and error recovery procedures	**Backup Strategy:** Daily automated backups of databases and results **Error Handling:** Graceful failure handling with logging **Recovery:** Quick restart capabilities and data validation
5.	**Performance**	Optimized for processing large datasets (180+ countries) with efficient algorithms and caching mechanisms	**Data Processing:** Vectorized operations with NumPy/pandas **Caching:** Result caching for repeated queries **Optimization:** Efficient PCA implementation, optimized SQL queries
6.	**Maintainability**	Clean code architecture with comprehensive documentation, version control, and standardized coding practices	**Version Control:** Git with GitHub repository **Documentation:** Comprehensive code comments and README files **Standards:** PEP 8 Python style guide compliance
7.	**Interoperability**	Supports multiple data formats and integration capabilities for academic research and policy analysis	**Data Formats:** CSV, Excel, JSON, PDF export capabilities **API Integration:** RESTful API design principles **Standards:** ISO country codes, standard economic indicators

S.No	Characteristics	Description	Technology
8.	**Reliability**	Implements robust error handling, data validation, and quality assurance mechanisms	**Data Validation:** Automated data quality checks **Error Logging:** Comprehensive logging system **Testing:** Unit tests for critical functions

---

Technology Stack Details

Frontend Technologies

- **HTML5/CSS3:** Modern web standards for user interface
- **JavaScript:** Interactive dashboard functionality
- **Bootstrap 5:** Responsive design framework
- **Tableau Public:** Professional data visualization platform

Backend Technologies

- **Python 3.8+:** Primary programming language
- **Flask/Django:** Web framework for API development
- **pandas:** Data manipulation and analysis
- **NumPy:** Numerical computing
- **scikit-learn:** Machine learning and statistical analysis

Database Technologies

- **SQLite:** Development and testing database
- **PostgreSQL:** Production database (if scaled)
- **File-based storage:** CSV/Excel for data exchange

Development Tools

- **Git/GitHub:** Version control and code repository
- **Jupyter Notebooks:** Exploratory data analysis and prototyping
- **Visual Studio Code:** Integrated development environment
- **Python virtual environments:** Dependency management

Deployment and Hosting

- **GitHub Pages:** Documentation and project presentation
- **Tableau Public:** Dashboard hosting and sharing

- **Local deployment:** Development and testing environment

System Requirements

Hardware Requirements

- **Minimum:** 8GB RAM, 4-core processor, 50GB storage
- **Recommended:** 16GB RAM, 8-core processor, 100GB SSD storage
- **Network:** Stable internet connection for API access

Software Requirements

- **Operating System:** Windows 10+, macOS 10.15+, or Linux Ubuntu 18+
- **Python:** Version 3.8 or higher
- **Web Browser:** Chrome 90+, Firefox 88+, Safari 14+, Edge 90+

Development Environment

- **IDE:** Visual Studio Code, PyCharm, or Jupyter Lab
- **Version Control:** Git 2.30+
- **Package Manager:** pip or conda for Python dependencies

Integration Architecture

Data Flow Architecture

1. **Data Ingestion:** Heritage Foundation API → Raw Data Storage
2. **Data Processing:** Raw Data → Cleaned/Normalized Data
3. **Analysis Pipeline:** Processed Data → Statistical Analysis → EFCI Calculation
4. **Visualization Pipeline:** Analysis Results → Charts/Dashboards
5. **Output Generation:** Results → Reports/Exports

API Integration Strategy

- **Heritage Foundation API:** Primary data source with scheduled updates
- **Error Handling:** Graceful degradation with cached data fallback
- **Rate Limiting:** Respect API limits with appropriate delays
- **Data Validation:** Automated checks for data completeness and accuracy

---

References

- Heritage Foundation API Documentation: https://www.heritage.org/index/
- Tableau Public Gallery: https://public.tableau.com/gallery/
- GitHub Repository: https://github.com/anvitharegidi/Measuring-the-Pulse-of-Prosperity-An-Index-of-Economic-Freedom-Analysis-