Project Design Phase
Solution Architecture

Team ID	LTVIP2026TMIDS87645
Project Name	Prosperity Prognosticator
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a structured approach that defines how a software system is designed to bridge the gap between business problems and technology solutions. It describes the overall structure of the system, its components, interactions, and data flow to ensure that the proposed solution effectively meets stakeholder requirements.

The solution architecture of **Prosperity Prognosticator** focuses on providing a scalable, reliable, and efficient system that predicts startup success using machine learning techniques.

Goals of the Solution Architecture

The primary goals of the solution architecture are to:

- Identify the most suitable technological solution to address the startup success prediction problem.
- Describe the structure, behavior, and interaction of system components.
- Define the key features, development phases, and solution requirements.
- Provide clear specifications for development, deployment, and maintenance of the system.

Solution Architecture Description

The Prosperity Prognosticator follows a **layered architecture**, where each layer has a specific responsibility. This design improves modularity, scalability, and maintainability.

1. **Presentation Layer (User Interface)**
   This layer provides a web-based interface that allows investors and startup founders to input startup details. It is designed to be user-friendly and responsive.
2. **Application Layer (Backend Server)**
   The backend layer is implemented using Flask. It handles user requests, validates input data, and manages communication between the frontend and the machine learning model.
3. **Machine Learning Layer**
   This layer contains the trained Random Forest model. It processes the validated input data and predicts the startup outcome as *Acquired* or *Closed*.
4. **Data Layer**
   The data layer stores the startup dataset and model-related files. It supports training, testing, and prediction processes.

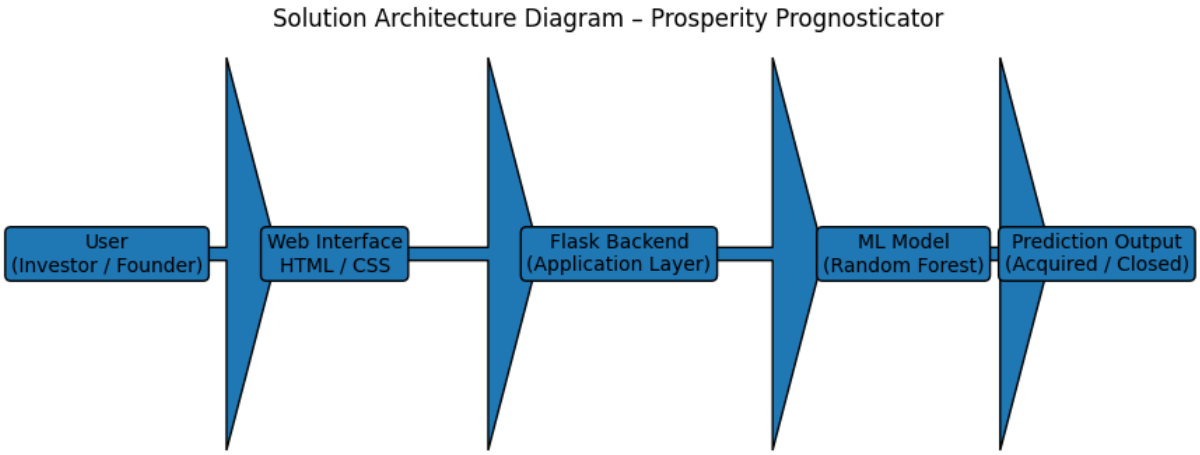
5. **Output Layer**
   The prediction result is sent back to the user interface and displayed clearly on the dashboard.

Solution Architecture Flow

1. The user enters startup details through the web interface.
2. The input data is sent to the Flask backend.
3. The backend preprocesses and validates the data.
4. The machine learning model analyzes the input.
5. The prediction result is generated.
6. The result is displayed to the user.

Advantages of the Proposed Architecture

- Modular and easy to maintain
- Supports scalability and future enhancements
- Enables fast and accurate predictions
- Separates business logic and machine learning logic



Reference

https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-architecture-considerations/