

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

Team ID	LTVIP2026TMIDS87645
Project Name	Prosperity Prognosticator
Maximum Marks	4 Marks

**Technical Architecture:**

The Prosperity Prognosticator system follows a layered architecture where user input is collected through a web interface, processed by the backend server, and analyzed using a machine learning model to generate prediction results. The architecture ensures modularity, scalability, and ease of maintenance.

The system supports real-time prediction by integrating a trained machine learning model with a Flask-based backend application.

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

Component	Description	Technology
User Interface	Provides interaction with the application to input startup details	HTML, CSS, JavaScript
Application Logic – 1	Handles user requests and form submission	Flask (Python)
Application Logic – 2	Processes input data and prepares it for prediction	Python
Application Logic – 3	Executes machine learning prediction logic	Scikit-learn
Database	Stores startup dataset and related data	CSV / Local Storage
ML Model	Predicts startup outcome (Acquired / Closed)	Random Forest Classifier
External API (Optional)	Can be used for future enhancements	Not Applicable
Infrastructure	Hosts application and model	Local System / Cloud Deployment

**Table-2: Application Characteristics:**

Characteristics	Description	Technology
<b>Open Source Frameworks</b>	Uses open-source frameworks for development	Flask, Scikit-learn
<b>Security Implementations</b>	Secures user inputs and prevents invalid access	Flask validation, Input Sanitization
<b>Scalable Architecture</b>	Can be extended to support more users and data	Modular Flask Architecture

**Table-3: Non-Functional Characteristics:**

Characteristics	Description	Technology
<b>Availability</b>	The application is accessible whenever required by users	Localhost / Cloud
<b>Performance</b>	Generates prediction results within seconds	Optimized ML Model
<b>Reliability</b>	Provides consistent prediction results	Trained & Tested ML Model

**References:**

- <https://flask.palletsprojects.com>
- <https://scikit-learn.org>
- <https://www.python.org>
- <https://developer.mozilla.org>