Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story Points)

Date	30 January 2026
Team ID	LTVIP2026TMIDS76348
Project Name	Prosperity Prognosticator – Machine Learning for Startup Success Prediction
Maximum Marks	5 Marks

Product Backlog & Sprint Schedule

Sprint	Functional Requirement (Epic)	User Story No.	User Story / Task	Story Points	Priority	Team Members
Sprint-1	UI Development	USN-1	As an investor or entrepreneur, I can enter startup metrics (funding, milestones, relationships) in a clean web form.	3	**High**	Frontend Team
	Data Validation	USN-2	As a system, I validate all 9 numeric inputs before passing them to the model to prevent errors.	2	**High**	Backend Team
	Model Integration	USN-3	As a user, I receive a startup success prediction (Acquired or Closed) from the Random Forest model instantly.	5	**High**	ML Team
Sprint-2	Preprocessing Pipeline	USN-4	As a system, I preprocess input features — remove negatives, handle missing values, and reshape the array — before model inference.	5	**High**	ML Team
	EDA & Visualization	USN-5	As a data analyst, I perform univariate, bivariate and multivariate analysis with distplots, heatmaps, countplots, and box plots.	5	**High**	ML Team
	Result Page Design	USN-6	As a user, I see a visually distinct result page — green for Acquired, red for Closed — with clear prediction label and navigation buttons.	3	**High**	Frontend Team
Sprint-3	Performance Optimization	USN-7	As a user, I get my startup prediction result within 2 seconds of submitting the form.	2	**Medium**	Backend Team
	Model Training & Saving	USN-8	As a system, I train the Random Forest classifier on startup1.csv and save it as random_forest_model.pkl using joblib for reuse.	5	**High**	ML Team
	Deployment Setup	USN-9	As a team, we deploy the Flask application locally and verify all routes (/, /predict GET, /predict POST) work correctly.	3	**Medium**	DevOps Team

Sprint	Functional Requirement (Epic)	User Story No.	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Model Improvement	USN-10	As a system, I improve prediction accuracy by evaluating ROC-AUC, Precision-Recall curves, and confusion matrix on the test set.	5	Medium	ML Team
	Documentation & Testing	USN-11	As a team, we complete end-to-end testing of all application pages, validate predictions with sample inputs, and finalise project documentation.	3	High	All Members

Story Points Summary

Sprint	Story Points	User Stories
Sprint-1	10	USN-1, USN-2, USN-3
Sprint-2	13	USN-4, USN-5, USN-6
Sprint-3	10	USN-7, USN-8, USN-9
Sprint-4	8	USN-10, USN-11
Total	**41**	**11 User Stories**

Sprint Planning Summary

Sprint-1: Develop the frontend input form, implement input validation, and integrate the trained Random Forest model to return an Acquired/Closed prediction.

Sprint-2: Implement the full data preprocessing pipeline (null handling, negative removal, feature selection), complete EDA (distplots, heatmaps, box plots), and design the result page with visual indicators.

Sprint-3: Optimise prediction response time to under 2 seconds, train and save the Random Forest model as a .pkl file using joblib, and deploy the Flask application on localhost.

Sprint-4: Improve model performance by analysing confusion matrix, ROC-AUC, and Precision-Recall scores. Complete end-to-end application testing and finalise all project documentation.

Estimation Approach

Story points were estimated using relative complexity and effort required for the Prosperity Prognosticator project. Tasks involving ML model training, preprocessing pipelines, and EDA were assigned higher points (5) due to technical depth. Frontend UI design and result page tasks were assigned moderate points (3), while input validation and performance optimisation tasks were assigned lower points (2) reflecting their comparatively straightforward implementation.