Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	8 February 2026
Team ID	LTVIP2026TMIDS76029
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**View Home Page**	Display project title and description
		Provide navigation buttons (Home, Predict)
		Provide “Start Prediction” button
FR-2	**Enter Transaction Details**	Input Step (Time)
		Select Transaction Type (CASH_OUT, PAYMENT, TRANSFER, etc.)
		Enter Transaction Amount
		Enter Old Balance (Sender)
		Enter New Balance (Sender)
		Enter Old Balance (Receiver)
		Enter New Balance (Receiver)
FR-3	**Fraud Prediction Processing**	Send input data to trained ML model
		Load payments.pkl model
		Process transaction using best trained algorithm
		Generate Fraud / Not Fraud prediction
FR-4	**Display Prediction Result**	Show prediction result on screen
		Display color-coded output (Red = Fraud, Green = Not Fraud)

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
		Provide “Predict Another” button
		Provide “Back to Home” button

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system should provide a simple and user-friendly web interface where users can easily enter transaction details and view prediction results without technical knowledge.
NFR-2	**Security**	The system should securely handle user input data and prevent unauthorized access. The trained model file (payments.pkl) must be protected from modification.
NFR-3	**Reliability**	The system should consistently generate accurate fraud predictions based on the trained machine learning model without crashing or producing incorrect outputs.
NFR-4	**Performance**	The system should generate fraud prediction results in real-time with minimal delay after submitting transaction details.
NFR-5	**Availability**	The web application should be accessible whenever the Flask server is running and should function properly across modern web browsers.
NFR-6	**Scalability**	The system should be designed in a way that allows future improvements such as adding more features, integrating new ML models, or handling larger datasets.