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

Date	June 2025
Team ID	LTVIP2025TMID33624
Project Name	Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables
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	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Image Prediction	Upload image via web/mobile - Classify fruit/vegetable as fresh or rotten - Show prediction result with confidence score
FR-4	Tips and Feedback	Display food storage tips after prediction - Show warning if food is predicted as rotten - Allow user to rate prediction accuracy
FR-5	User Dashboard	- View prediction history - Re-upload failed images - View tips previously accessed

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application must have a clean, intuitive, and mobile-friendly user interface. Users should be able to register, upload images, and get predictions easily.

NFR-2	**Security**	User data, uploaded images, and login credentials must be encrypted. Google Sign-In and secure login flows must be implemented to protect user accounts.
NFR-3	**Reliability**	The model should consistently provide accurate predictions, and the application should handle common edge cases and failures gracefully without crashing.
NFR-4	**Performance**	Image upload and prediction should be processed within **3–5 seconds** for a smooth user experience. Backend APIs should respond efficiently under load.
NFR-5	**Availability**	The application must be available at least 99% of the time, including during peak usage hours. Deployment platforms like Render or Heroku should ensure uptime.
NFR-6	Scalability	The solution should scale to support multiple users uploading images simultaneously. The backend and model should be deployable on scalable cloud infrastructure.