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

Date	13 April 2025
Team ID	SWTID1743509015
Project Name	iMovies – Online Movie Ticket Booking System
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 Facebook (optional)
FR-2	User Confirmation	Confirmation via Email
		Confirmation via OTP (future scope)
FR-3	User Login	Login with Email and Password
		Login with Gmail (future enhancement)
FR-4	Movie Browsing	View Trending Movies
		Browse by Categories (Action, Horror, etc.)
FR-5	Offers Display	View Latest Movie Offers
FR-6	Favorites	Add/Remove Movies to/from Favorites
		View Favorite Movies
FR-7	Movie Recommendations	Recommend movies based on favorites
FR-8	Admin Panel	Admin Login
		Manage Movies (Add/Edit/Delete)
		Manage Offers

Non-functional Requirements:

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

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application provides a clean, modern, and responsive user interface for both desktop and mobile users. Navigation is intuitive, and features are easily accessible.
NFR-2	**Security**	User passwords are hashed using bcrypt. JWT-based authentication is implemented for secure login. Role-based access control is used to separate user and admin functionalities.
NFR-3	**Reliability**	The application is stable under typical usage scenarios and handles API failures gracefully using proper error handling and fallback responses.
NFR-4	**Performance**	The system loads trending movies and categories efficiently using optimized MongoDB queries and lazy-loaded components.
NFR-5	**Availability**	The service is expected to have high availability with minimal downtime when hosted on platforms like Vercel/Render (frontend) and Railway (backend).
NFR-6	**Scalability**	The backend is built using RESTful APIs with a modular code structure, allowing for horizontal scaling. The MongoDB database supports high-volume data storage for future growth.