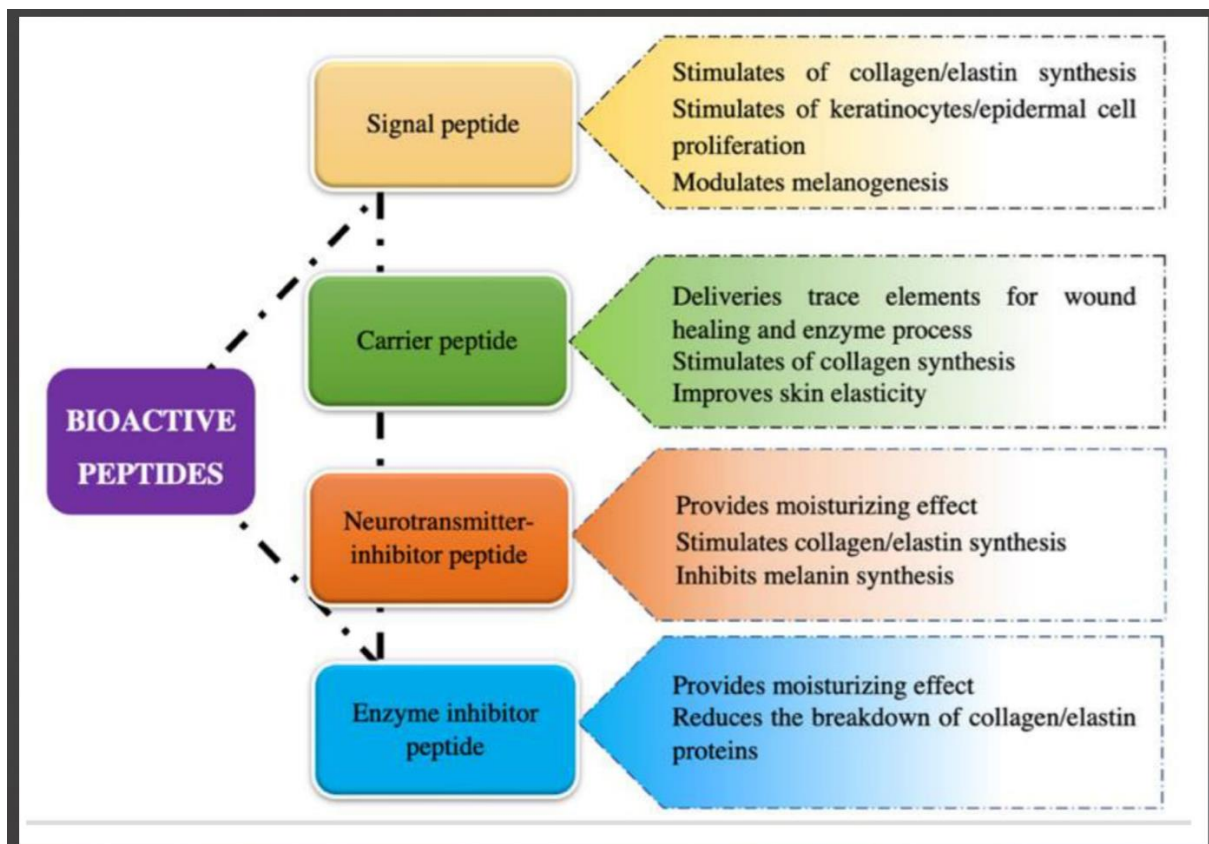


Project Design Phase-II

Solution Requirements (Functional & Non-functional)

Date	June 2025
Team ID	LTVIP2025TMID50930
Project Name	Cosmetic Insights _ Navigating Cosmetics Trends and Consumer Insights with Tableau
Maximum Marks	4 Marks



Functional Requirements:

Following are the functional requirements of the proposed solution.

S.NO.	Component	Description	Technology
1.	User Authentication	Register, login, and manage user sessions securely	Firebase Auth / OAuth / JWT
2.	Admin Dashboard	Admin panel to manage products, users, and view analytics	React / Angular + Node.js
3.	Product CatLog	Displays a list of cosmetics with filters and search features	React / MongoDB
4.	Review Sentiment Analysis	Analyses customer reviews for sentiment and trends	Python (NLTK / Text Blob)
5.	Trend Visualization	Visualizes trending products across time and location	Chart.js / D3.js / Power BI
6.	Data Collection Module	Gathers product reviews and ratings from web or internal sources	Web Scraping / APIs / ETL scripts
7.	Notification System	Sends alerts or insights to admins or users	Firebase Cloud Messaging / Twilio
8.	Search Functionality	Allows users to search cosmetics by name, brand, or category	Elasticsearch / Alogia
9.	User Profile Management	Users can view and edit their own information	Node.js / Express + MongoDB
10.	Product Comparison Feature	Compares multiple cosmetic products by ratings and features	React + Backend API
11.	Reporting Engine	Generates downloadable reports with insights and charts	Python (Pandas, Report Lab)

Table-2: Application Characteristics:

S.NO.	Characteristics	Description	Technology
1.	Web-Based Access	Platform accessible through web browsers without needing installation	HTML, CSS, JavaScript, React.js
2.	Data Analytics	Capable of analysing trends, sentiments, and user reviews	Python (Pandas, NLTK), Power BI
3.	Scalability	Can support increasing user traffic and data volume	AWS / Azure Cloud, Docker, Kubernetes
4.	Real-time Updates	Displays trends and data updates in near real-time	Firebase / WebSocket / Node.js