

# SkinSavvy: AI-Powered Skincare Analysis & Recommendation System

## 1. Core User Features & Functionality

Feature Category	Specific Features
User Authentication & Management	<ul style="list-style-type: none"><li>User Registration &amp; Login (JWT)</li><li>Role-Based Access Control (Free, Premium, Admin)</li><li>User Profile Management</li></ul>
Skin Analysis	<ul style="list-style-type: none"><li>AI-Powered Image Analysis (Acne, Dark Spots, Wrinkles, Normal)</li><li>Camera Capture Interface with Guidance</li><li>Image Upload</li><li>Analysis with Confidence Scoring</li></ul>
Recommendations & Insights	<ul style="list-style-type: none"><li>Personalized Product Recommendations</li><li>LLM-Powered Friendly Explanations &amp; Tips (Ollama)</li><li>Product "Buy Now" Affiliate Links</li></ul>
User History & Data	<ul style="list-style-type: none"><li>Personal Analysis History Timeline</li><li>Premium Feature: Detailed PDF Reports</li><li>Premium Feature: Personalized Skincare Routines (AM/PM)</li></ul>
Administration	<ul style="list-style-type: none"><li>Admin Dashboard</li><li>CRUD Operations on Product Database</li><li>User Management (View, Edit, Delete)</li><li>System Analytics &amp; Moderation</li></ul>

## 2. Technical Stack & Technologies Used

Layer	Technology	Version / Specifics	Purpose
Frontend	React.js	18+	Building the user interface
	React Native	0.72+	(Optional) Mobile app development
	Chakra UI / Material-UI	Latest	UI component library & styling
	Axios	Latest	HTTP client for API calls
Backend	Node.js	18+	Runtime environment
	Express.js	4.18+	Web framework for API
	JWT (jsonwebtoken)	Latest	Authentication & authorization
	Bcrypt	Latest	Password hashing
	Multer	Latest	File upload handling
AI/ML Service	Python	3.10+	Core ML language
	TensorFlow & Keras	2.15.0	Deep learning framework
	MobileNetV2	-	Pre-trained model for transfer learning
	OpenCV	4.8+	Image preprocessing
	NumPy, Pandas	Latest	Data manipulation
	Ollama	Latest	To run local LLMs (Llama 3.2)
Database	PostgreSQL	15+	Primary relational database
	pg (Node.js driver)	Latest	Database connector

Layer	Technology	Version / Specifics	Purpose
File Storage	Amazon S3 / MinIO	-	Cloud object storage for images & models
DevOps & Deployment	Docker	Latest	Containerization
	Docker Compose	Latest	Multi-container orchestration
	NGINX	Latest	Reverse proxy & load balancer
	(Potential) AWS EC2-		Deployment server

### 3. AI & Machine Learning Models

Model	Type	Purpose	Source
Custom Skin Condition Classifier	Convolutional Neural Network (CNN)	Classifying skin images into conditions (acne, etc.)	<b>Fine-tuned from MobileNetV2</b> on a custom dataset
Llama 3.2:1B	Large Language Model (LLM)	Generating friendly, explanatory text for users	Downloaded and run locally via <b>Ollama</b>

### 4. Development & Project Management Tools

Category	Tool	Purpose
Version Control	Git	Source code management
	GitHub / GitLab	Code hosting, collaboration, and CI/CD
Methodology	<b>Agile SDLC</b>	Project management framework (Sprints, Backlog)

Category	Tool	Purpose
Design & Prototyping	Figma / Adobe XD	UI/UX design and prototyping
API Testing	Postman / Thunder Client	Testing backend API endpoints
IDE	VS Code	Primary code editor with extensions
Documentation	Markdown	Writing <code>README.md</code> and documentation
	Draw.io / Mermaid	Creating architecture diagrams and flowcharts

## 5. Key System Characteristics

Characteristic	Description
Architecture	<b>Microservices-based</b> , containerized with Docker. Clear separation between Frontend, Backend API, AI Service, and Database.
Data Flow	RESTful API communication between all services.
Security	JWT-based authentication, password hashing, role-based access control.
Scalability	Designed with microservices, making it easy to scale individual components (e.g., AI service) independently.
Portability	<b>Fully Dockerized.</b> The entire system can run on any machine that has Docker installed, eliminating "it works on my machine" problems.
Privacy	All user data and image processing happen on your own server; no reliance on external APIs for core analysis.

## 6. Documentation & Deliverables

Document	Description
<b>Project Report/Thesis</b>	Detailed document following SDLC phases: Introduction, Analysis, Design, Implementation, Testing, Conclusion.
<b>System Architecture Diagrams</b>	Diagrams showing component interaction (e.g., DFD Level 0-2, Container Diagram).
<b>API Documentation</b>	Swagger/OpenAPI specs or detailed endpoint lists for the backend API.
<b>UI/UX Prototypes</b>	Figma mockups and wireframes of all major screens.
<b>Source Code</b>	Well-commented code hosted on GitHub.
<b>Setup Guide</b>	Comprehensive <b>README.md</b> with instructions to build and run the project.
<b>Demo Video</b>	A video showcasing the functionality of the final application.