**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 15 February 2025 |
| Team ID | LTVIP2025TMID50498 |
| Project Name | Cosmetic Insights |
| Maximum Marks | 4 Marks |

**Solution Architecture – Cosmetic Insights Web App**

**Goals of the Architecture**

* ✅ Bridge the gap between cosmetic industry business needs and accessible data visualization tools
* ✅ Deliver a lightweight, scalable, and user-friendly solution for market insights
* ✅ Define a roadmap of how technology will serve various project stakeholders
* ✅ Provide a clear structure for implementation, deployment, and further development

[ User (Browser) ]

|

v

[ Flask Web Server (app.py) ]

|

+--> [ Tableau Dashboards Embedded via iframe ]

|

+--> [ Feedback Form --> Flask POST Route ]

|

v

[ Static/Template Files (HTML, CSS) ]

|

v

[ Deployment via Render.com ]

**⚙️ Component Breakdown**

| **Component** | **Role & Functionality** |
| --- | --- |
| **Frontend (HTML/CSS)** | Renders a clean, responsive interface for users to view dashboards and submit feedback |
| **Flask (Python)** | Acts as the core server handling routing, form submission, and template rendering |
| **Tableau Public** | Provides the visual analytics dashboards via embeddable iframes |
| **Feedback Handling** | Receives user feedback via POST requests and logs/sends for further action |
| **Render.com** | Cloud deployment platform hosting the app with CI/CD integration |

**🧠 Features & Development Phases**

| **Phase** | **Feature** | **Status** |
| --- | --- | --- |
| Phase 1 | Flask setup, static pages, dashboard embed | ✅ Completed |
| Phase 2 | Add feedback form + POST processing | ✅ Completed |
| Phase 3 | Responsive UI design | ✅ Completed |
| Phase 4 | Deployment on Render.com | ✅ Completed |
| Phase 5 | Optional user account system, dynamic dashboards | 🚧 Planned |
| Phase 6 | Advanced analytics (AI/ML integration) | 🔜 Future Scope |

**📐 Architecture Characteristics**

| **Attribute** | **Value** |
| --- | --- |
| **Scalability** | Can scale horizontally by hosting more dashboards or using a CDN |
| **Maintainability** | Simple file structure and separation of concerns (Flask MVC pattern) |
| **Flexibility** | Easily extendable for other domains like fashion, fitness, etc. |
| **Accessibility** | No login required, device/browser agnostic interface |
| **Security** | No sensitive data stored; can add reCAPTCHA or validation on feedback |

**📄 Solution Specifications**

* **Port Configuration**: Uses default Flask port (or custom port 10000 if required by Render)
* **Deployment Environment**: Python 3.x, requirements.txt for dependencies
* **Monitoring**: Basic logging through server terminal or deployment dashboard (Render logs)
* **Version Control**: Managed via GitHub