Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	28th June 2025
Team ID	LTVIP2025TMID34819
Project Name	GrainPalette A Deep Learning Odyssey In Rice Type Classification Through Transfer Learning
Maximum Marks	4 Marks

Technical Architecture:

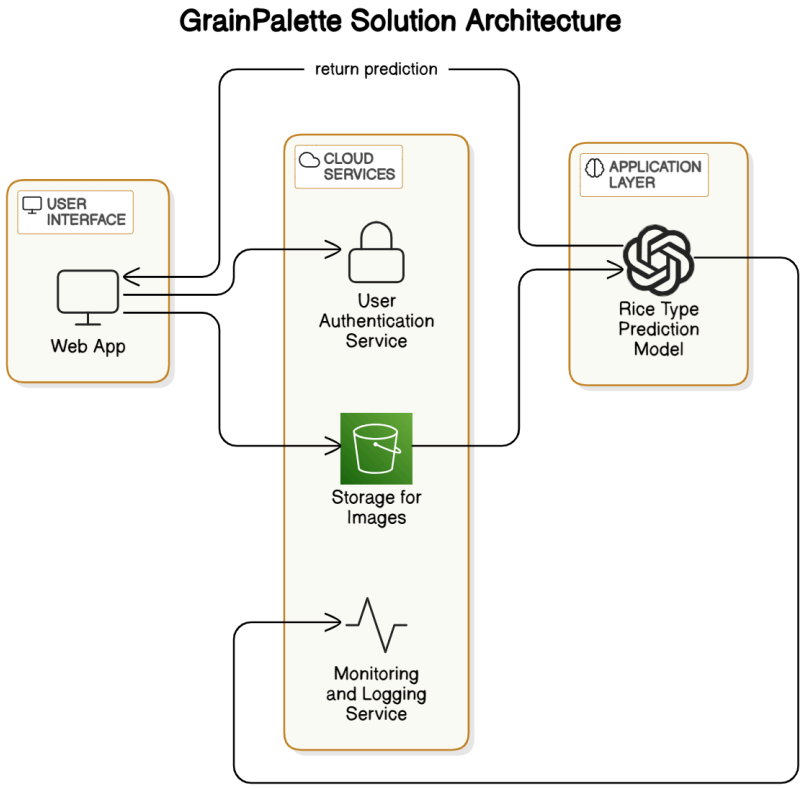

Table-1 : Components & Technologies:

S.No.	Component	Description	Technology/Stack (Example Choices)
1	User Interface (UI)	Client-side user interaction; Image upload & result display.	HTML, CSS, JavaScript, React/Angular/Vue.js
2	Backend API (App Logic)	Application processing logic; Image pre-processing, ML model calls, result handling.	Python (Flask/Django), Node.js, Java Spring Boot
3	AI/ML Model Layer	Rice Type prediction; Trained MobileNetv4 for image classification.	TensorFlow, PyTorch, MobileNetv4, Transfer Learning
4	Database (Optional)	User Data & Prediction History storage; For user accounts & data persistence (optional).	Cloud SQL (GCP, AWS, Azure), MongoDB, Firebase Firestore
5	Storage (Cloud)	File storage for uploaded rice grain images.	AWS S3, Google Cloud Storage, Azure Blob Storage
6	Monitoring & Logging (Cloud)	Error tracking & Performance monitoring; Application health & usage tracking.	Prometheus, Grafana, ELK Stack, Cloud Monitoring/Logging (AWS, GCP, Azure)
7	Cloud Infrastructure (Server)	Application hosting; Platform for deployment & scalability.	Google Cloud Platform (GCP), Amazon Web Services (AWS), Azure, Kubernetes, Docker
8	Interface APIs (External)	Potential future integrations for data enrichment from external sources.	REST APIs, Webhooks

Table-2: Application Characteristics:

S.No.	Characteristics	Description	Technology Justification (Example)
1	Open Source Frameworks	Utilize open-source tools to reduce costs and leverage community support.	Python, TensorFlow/PyTorch, React/Vue.js, Kubernetes, Prometheus, Grafana, ELK Stack
2	Reusability	Design for reusable components to facilitate future enhancements/modules.	Microservices architecture (if applicable), Modular design of API & UI components
3	Security Requirements	Protect user data and ensure secure API access.	HTTPS, OAuth 2.0 (for Authentication), Input validation, Secure cloud service configurations
4	Availability	Ensure high availability of the service for consistent user access.	Cloud-based infrastructure, Load balancing, Redundancy, Container Orchestration (Kubernetes)

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d