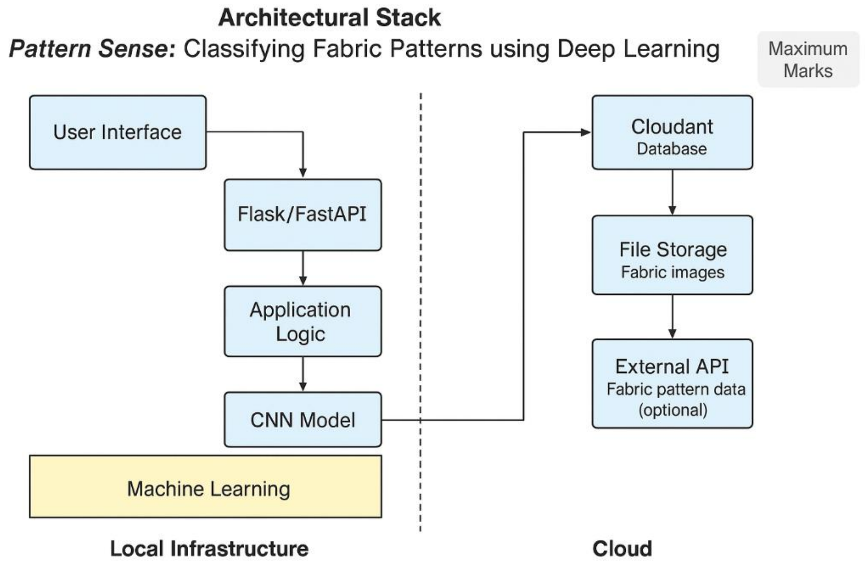
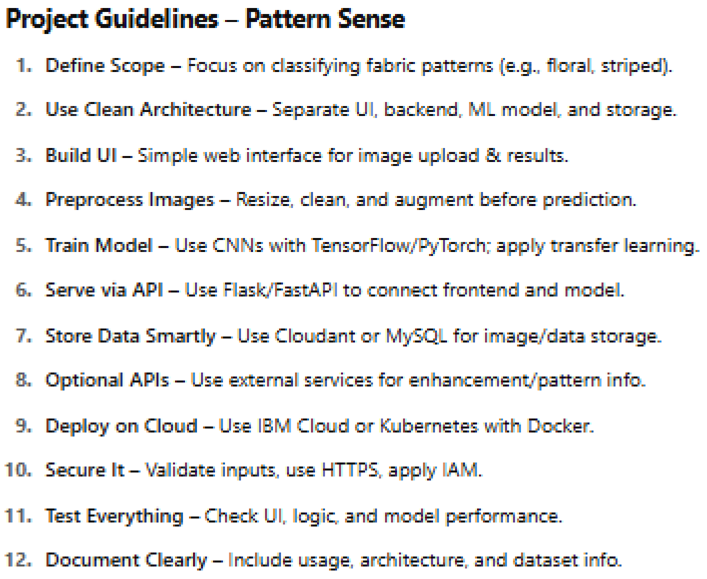
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2025TMID33870
Project Name	Pattern Sense: Classifying Fabric Patterns using Deep Learning
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2





Project Guidelines – Pattern Sense

1. **Define Scope** – Focus on classifying fabric patterns (e.g., floral, striped).
2. **Use Clean Architecture** – Separate UI, backend, ML model, and storage.
3. **Build UI** – Simple web interface for image upload & results.
4. **Preprocess Images** – Resize, clean, and augment before prediction.
5. **Train Model** – Use CNNs with TensorFlow/PyTorch; apply transfer learning.
6. **Serve via API** – Use Flask/FastAPI to connect frontend and model.
7. **Store Data Smartly** – Use Cloudant or MySQL for image/data storage.
8. **Optional APIs** – Use external services for enhancement/pattern info.
9. **Deploy on Cloud** – Use IBM Cloud or Kubernetes with Docker.
10. **Secure It** – Validate inputs, use HTTPS, apply IAM.
11. **Test Everything** – Check UI, logic, and model performance.
12. **Document Clearly** – Include usage, architecture, and dataset info.

(Diagram placeholder: You can add an architectural diagram showing user interface, backend logic, ML model interface, cloud/local storage, and external APIs.)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web interface for uploading images & viewing results	HTML, CSS, React.js
2.	Application Logic-1	Preprocessing and handling image uploads	Python
3.	Application Logic-2	Pattern detection using CNN model	TensorFlow / PyTorch
4.	Application Logic-3	Model inference and result output	Flask / FastAPI
5.	Database	Metadata storage	MySQL
6.	Cloud Database	Cloud storage for training images and logs	IBM Cloudant
7.	File Storage	Fabric images for training and prediction	IBM Block Storage
8.	External API-1	Fabric pattern database access (if applicable)	Custom Fabric Pattern API
9.	External API-2	Image enhancement via third-party service (optional)	Adobe API or DeepAI
10.	Machine Learning Model	Classifies fabric patterns (e.g., floral, geometric)	CNN model trained in TensorFlow
11.	Infrastructure	Deployment and hosting	IBM Cloud Kubernetes or Cloud Foundry

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frontend and ML libraries	React, TensorFlow, Flask
2.	Security Implementations	Image upload validation, IAM, encryption	SHA-256, IAM, OWASP Secure Headers
3.	Scalable Architecture	Microservices with REST APIs	Docker, Kubernetes
4.	Availability	High availability using distributed servers and load balancing	IBM Load Balancer, Kubernetes
5.	Performance	Caching, optimized ML inference, CDN usage for static assets	Redis, Cloudflare CDN

References:

1. **C4 Model for Architecture Design**
   https://c4model.com/

2. **AI-Powered Order Processing Example**
   https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

3. **IBM Cloud Architecture Center**
   https://www.ibm.com/cloud/architecture

4. **AWS Architecture Center**
   https://aws.amazon.com/architecture

5. **How to Draw Useful Architecture Diagrams**
   https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d

6. **TensorFlow: Image Classification Guide**
   https://www.tensorflow.org/tutorials/images/classification

7. **PyTorch: Transfer Learning for Image Classification**
   https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

8. **IBM Cloudant NoSQL DB**
   https://www.ibm.com/cloud/cloudant

9. **Flask Documentation**
   https://flask.palletsprojects.com/

10. **FastAPI Documentation**
    https://fastapi.tiangolo.com/