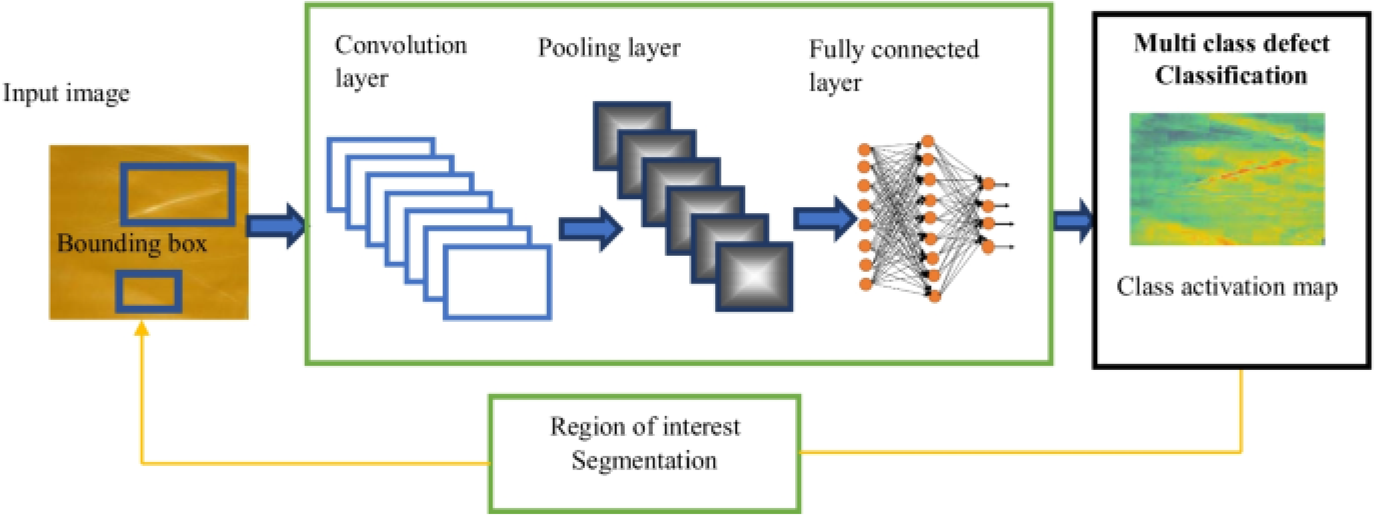
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2025TMID34512
Project Name	**Pattern Sense: Classifying Fabric Patterns using Deep Learning**
Maximum Marks	4 Marks

Technical Architecture:

The technical architecture for *Pattern Sense* is designed to provide an end-to-end, automated solution for classifying fabric patterns using deep learning models. The architecture ensures scalability, reliability, security, and real-time results to assist users in accurately identifying fabric patterns via a web or mobile platform.The solution follows a layered architecture that integrates frontend applications, backend services, deep learning infrastructure, and cloud hosting.



Architecture Flow

1. **User Access**
   Users interact with the system via a web application (ReactJS) or mobile app (Flutter).
2. **User Registration & Authentication**
   Users register and log in through Firebase Authentication or a custom Email/OTP system.
3. **Pattern Upload**
   Users upload fabric images via gallery selection or live camera capture.
4. **Backend Processing**
   The backend (Node.js or Python FastAPI) handles API requests, processes images, and communicates with the deep learning model.
5. **Deep Learning Model**
   A trained model (TensorFlow or PyTorch) receives the image, performs pattern classification, and returns results along with a confidence score.
6. **Data Storage**
   MongoDB or PostgreSQL stores user data, uploaded images, and classification history.
7. **Result Display**
   Classification results are displayed to the user in real-time with an option to review or download reports.
8. **Cloud Deployment**
   The system is hosted on scalable cloud infrastructure (AWS, Azure, or Google Cloud).

Table-1 : Components & Technologies:

S.NO	Component Name	Description	Responsibility

1	**User Interface (UI)**	Web or Mobile Interface for users	User registration, pattern upload, results display
2	**Authentication Module**	Handles user verification via Email/OTP	Secure Login and Registration
3	**Pattern Upload Service**	Facilitates image uploads via gallery or live camera	Receives pattern images for classification
4	**Deep Learning Model**	Trained model for fabric pattern classification	Detects and categorizes fabric patterns
5	**Database**	Stores user details, uploaded patterns, and classification history	Data persistence and retrieval
6	**Notification Service**	Sends confirmation emails or OTP to users	User verification and updates

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	**Frontend**	The user interface will be developed using **ReactJS** for web-based access and **Flutter** for cross-platform mobile access. This ensures responsive, interactive user experience.	ReactJS / Flutter (Mobile option)

S.No	Characteristics	Description	Technology
2.	**Backend**	he backend will be powered by **Node.js** (JavaScript) or **Python (FastAPI)** for handling API requests, managing business logic, and integrating with the Deep Learning model.	Node.js / Python (FastAPI)
3.	**Deep Learning Model**	**TensorFlow** or **PyTorch** will be used to build, train, and deploy the deep learning model for automated fabric pattern classification with high accuracy.	TensorFlow / PyTorch
4.	**Database**	**MongoDB** (for flexible document-based storage) or **PostgreSQL** (for structured relational data) will store user information, uploaded images, and classification results.	MongoDB / PostgreSQL
1.	**Authentication**	**Firebase Authentication** or a custom Email and OTP system will secure user access and registration, ensuring data privacy and system security.	Firebase Auth / Custom Email & OTP System
1.	**Cloud/Hosting**	The system will be deployed on cloud platforms like **AWS**, **Microsoft Azure**, or **Google Cloud**, offering high availability, scalability, and global access.	AWS / Azure / Google Cloud
1.	**Version Control**	**GitHub** or **GitLab** will be used for source code management, version control, and collaborative development across teams.	GitHub / GitLab