

## Question 1 :-Secure File Upload & Processing Service with Policy Enforcement

**Primary Tech: Python (FastAPI) or Node.js (Express), Docker**

**Problem Statement**-Build a system that:

- Allows users to upload files,
- Processes them (e.g., extracts metadata)
- Applies security and compliance policies
- Allows admins to review file status.

### System Components

#### Backend (Python or Node.js)

- POST /upload → Upload a file.
- GET /files → List uploaded files with status.
- POST /policies → Define file acceptance rules.
- Store files locally or in object storage (e.g., local FS).

#### Frontend (React or simple JS UI)

- Upload form.
- Table showing uploaded files and policy status.

#### Data Flow

- User uploads file.
- Backend extracts metadata (type, size, hash, etc.).
- Policy engine evaluates file against rules.
- File is accepted/rejected and stored.
- Frontend displays file status.

**NB - Use a policy-as-code engine (e.g., Open Policy Agent or similar lightweight policy evaluation library).not simple if-statements to enforce rules like:**

- Max file size,
- Allowed MIME types,
- Hash blacklist,
- File naming conventions.

+251912662951

Info@fineto.fi

www.fineto.fi

Old Airport Munteha Bldg 2nd floor  
Addis Ababa ET



**Deliverables**

- Complete, running source code
- README.md with setup & execution instructions.
- Brief design explanation & trade-offs.
- Simple architecture/data-flow diagram.

**Question 2 :- Deployment Approval Workflow System**

**Primary Tech:** Node.js (Express) or Python (FastAPI)

**Problem Statement-Build a system that:**

- Manages deployment requests,
- Requires multi-step approval before execution,
- Tracks deployment state transitions,
- Exposes APIs and UI for operators.

**System Components****Backend**

- POST /deployments → Create deployment request.
- POST /deployments/{id}/approve → Approve stage.
- POST /deployments/{id}/reject → Reject stage.
- GET /deployments → List deployments and status.

**Frontend- HTML + JavaScript + CSS**

- UI to Submit deployment request, View approval status, Approve/reject deployments

**Data Flow**

- User submits deployment request
- Backend initializes state machine.
- Approvals move deployment through states.
- Frontend displays state transitions.

**NB - Implement a state machine using a library (e.g., transitions in Python or xstate in JS), not manual status flags, to manage deployment lifecycle.**

- Complete, running source code.
- README.md with setup instructions.
- Design explanation and trade-offs.
- Architecture/data-flow diagram.