Video Upload & Processing System

A full-stack web application that allows users to upload videos, track upload progress, and store video metadata using **Node.js, Express, MongoDB, and HTML/CSS**.

Project Overview

This project demonstrates a **complete video upload and processing pipeline** with real-time progress tracking and database integration.
It is designed to simulate a real-world system where users upload media, which is stored securely and managed through a backend service.

Features

- 📤 Video upload with real-time progress bar
- 💾 Persistent storage using MongoDB
- 📁 Secure file handling using Multer
- 📊 Upload status tracking
- 🎥 Video preview after upload
- 🔄 RESTful API architecture
- 💻 Simple and responsive UI

Technology Stack

Layer	Technology
Frontend	HTML, CSS, JavaScript
Backend	Node.js, Express.js
Database	MongoDB
File Handling	Multer
Server	Node.js

Layer	Technology
Runtime	JavaScript

Project Structure

Pulse Assignment 2/

|

├── backend/

| ├── server.js

| ├── uploads/

| └── public/

| └── index.html

Installation & Setup

1. Install Node.js

Download and install Node.js from:
👉 https://nodejs.org

2. Install MongoDB

Download MongoDB Community Edition:
👉 https://www.mongodb.com/try/download/community

During installation:

- Enable **MongoDB Compass**
- Keep default settings

3. Start MongoDB Server

mongod

(Leave this window open)

4. Start the Application

Open a **new terminal** and run:

cd "C:\Users\ASUS\Downloads\Jasleen Documents\Pulse Assingment 2\Pulse Assingment 2\backend"

node server.js

You should see:

MongoDB Connected

Server running at http://localhost:5000

5. Open the App

Open your browser and go to:

http://localhost:5000

How It Works

1. User selects a video file.
2. File uploads via multipart form-data.
3. Backend stores the file in /uploads.
4. MongoDB stores metadata (filename, status).
5. Frontend displays upload progress.
6. Uploaded videos are listed and playable.

Database Structure

Collection: videos


```
{
  "_id": "ObjectId",
  "name": "sample.mp4",
  "status": "Safe",
  "uploadedAt": "2025-01-01T10:30:00Z"
}
```

Sample API Endpoints

Method	Endpoint	Description
POST	/upload	Upload video
GET	/videos	Fetch uploaded videos

Dependencies

express

multer

mongoose

Install with:

npm install express multer mongoose

Security Notes

- File uploads restricted to server directory
- MongoDB handles structured storage
- Can be extended with authentication (JWT)