



Innovation Center for Education



# YENEPOYA UNIVERSITY



**YENEPOYA**

(DEEMED TO BE UNIVERSITY)

Recognized under Sec 3(A) of the UGC Act 1956

Accredited by NAAC with 'A' Grade

## Build a Scalable File Processing System

### PROJECT SYNOPSIS

### BACHELOR OF COMPUTER APPLICATION

#### SUBMITTED BY

**Name:** Mohammed Ayad

**Class Roll No.:** 22BDACC158

**Present Address:** Mangalore

**Email:** ayadmohammed403@gmail.com

**Phone:** [your phone number here]

**Branch:** BCA (Cloud Computing Specialization)

**Batch:** 2022–2025

#### GUIDED BY

Ms. Manjula



Innovation Center for Education



## Table of Contents

1. <b>Cover Page</b> .....	1
2. <b>Introduction</b> .....	3
3. <b>Methodology / Planning of Work</b> .....	4



## INTRODUCTION

In the modern digital landscape, managing and processing large volumes of files securely and efficiently is a growing challenge. Traditional server-based systems often face limitations related to scalability, maintenance, and cost. To address these challenges, this project titled **“Build a Scalable File Processing System”** leverages a **serverless architecture** using cloud technologies.

The system is built using **Amazon Web Services (AWS)** — specifically **AWS Lambda**, **Amazon S3**, and **DynamoDB** — to enable event-driven, fully automated file processing. Users can upload files through a responsive frontend interface developed with **React.js** and **Tailwind CSS**. Once a file is uploaded, AWS Lambda automatically processes it, and metadata is stored in DynamoDB for easy retrieval.

This project eliminates the need for manual server management and demonstrates how serverless computing can be used to build scalable, cost-effective, and highly available cloud-native applications.

## METHODOLOGY / PLANNING OF WORK

The development of this project was carried out in a structured and phased manner to ensure systematic progress and completion. The key stages involved are:

### 1. Requirement Analysis

Identified the core functionality needed — including secure file upload, automatic processing, and metadata storage — and selected appropriate AWS services to implement them.

### 2. System Design

Designed architecture diagrams including Data Flow Diagrams (DFD) and Entity-Relationship Diagram (ERD). Planned user flow, database structure, and API routes.

### 3. Frontend Development

Created a responsive user interface using React.js and Tailwind CSS. Implemented routing, authentication forms, and file upload/dashboard components.

### 4. Backend Development & Cloud Integration

Developed APIs using Node.js and Express.js. Integrated AWS services (Lambda, S3, DynamoDB) to handle file storage, processing, and metadata management.

### 5. Testing & Deployment

Conducted functional, integration, and security testing using Postman and browser dev tools. Finalized the system with bug fixes and documented the complete workflow.