

# Project Documentation: Employee Management System

## Project Overview

This project is a full-stack Employee Management System designed using Angular (Frontend) and Node.js with Express (Backend), MongoDB for data storage, AWS S3 for image storage, and Nodemailer for sending emails. It allows users to manage employee data including image uploads, search, and messaging functionality

## Tech Stack

Layer	Technology
Frontend	Angular
Backend	Node.js, Express.js
Database	MongoDB (Mongoose)
Cloud	AWS S3 (Image Storage)
Email	Nodemailer (SMTP - Gmail)

## Backend Details

### ► API Endpoints

Method	Route	Description
POST	/api/employee	Create employee (with image upload)
GET	/api/employee	Get all employees
GET	/api/employee/:id	Get employee by ID
PUT	/api/employee/:id	Update employee
DELETE	/api/employee/:id	Delete employee
POST	/api/message	Send an email

### ► MongoDB Schema

```
{
  name: String,
  phone: String,
```

```
email: String,  
skills: [String],  
image: String, // S3 URL  
bio: String,  
position: String  
}
```

### ► *AWS S3 Integration*

- Uses @aws-sdk/client-s3 (v3) and multer-s3 for image uploads.
- Images are saved under: employee-images/{timestamp}\_{filename}

### ► *Nodemailer Integration*

- Uses Gmail SMTP to send emails.
- Requires EMAIL\_USER and EMAIL\_PASSWORD from .env.

## Frontend Details (Angular)

### ► *Components Overview*

Component	Description
CardComponent	Displays all employees as cards with name, email, position, and image and also has search functionality for filtering employees.
DetailsComponent	Shows detailed view of an employee. Options to edit, delete, or send email.
EditComponent	Allows editing employee info using form.
MessageComponent	Input form to compose and send email to an employee.
CrudComponent	Full CRUD operations on a dedicated page (list, add, update, delete).

### ► *Flow*

1. CardComponent → Employee click → DetailsComponent
2. DetailsComponent → Edit → EditComponent
3. DetailsComponent → Send Email → MessageComponent
4. CrudComponent is a standalone view for bulk operations.

## Environment Variables

Create a .env file in your backend root with:

```
PORT=5000
URI=mongodb://localhost:27017/employees
AWS_REGION=your-region
AWS_ACCESS_KEY_ID=your-access-key
AWS_SECRET_ACCESS_KEY=your-secret-key
AWS_S3_BUCKET=your-bucket-name
EMAIL_USER=your@gmail.com
EMAIL_PASSWORD=yourpassword
```

## Installation & Setup

### ***Backend:***

```
cd backend
npm install
npm run start
```

### ***Frontend:***

```
cd frontend
npm install
ng serve
```

### **Team Members:**

1. Shyam Kumar
2. Sudarshana M G