COLLEGE CODE : 9623

COLLEGE NAME : Amrita College Of Engineering And Technology

DEPARTMENT : Computer Science And Engineering

STUDENT NM-ID: 945280E2CD17FCFA8D042728D29B5ADB

ROLL NO : 962323104060

DATE : 15-09-2025

Completed the project named as Phase-2 TECHNOLOGY

PROJECT NAME : Employee Directory Search

SUBMITTED BY,

NAME : G.S. Mobin Rex
MOBILE NO: 7598779851

Employee Directory with Search

Phase 2 — Solution Design & Architecture

1. Tech Stack Selection

Layer	Technology
Frontend	React.js, Tailwind CSS
Backend	Node.js, Express.js
Database	MongoDB Atlas
Version Control	GitHub
Hosting	Netlify/GitHub Pages (Frontend), Render/Heroku (Backend)
Testing Tools	Postman, Jest, React Testing Library

2. UI Structure / API Schema Design

UI Structure:
- Navbar: Navigation across the site.
- Dashboard: Displays employee list in card/table format.
- Search Bar: Allows filtering employees by name, ID, or department.
- Employee Details Page: Shows detailed profile of an employee.
- Footer: Static information.

API Schema Design:

Endpoint	Method	Description
/api/employees	GET	Fetch all employee records
/api/employees?name=John	GET	Search employees by name
/api/employees/:id	GET	Fetch employee by ID

3. Data Handling Approach

Frontend (React):
- Uses useState and useEffect for state management.
- Search functionality implemented by filtering employee data dynamically.

Backend (Node.js/Express):
- Provides REST APIs for fetching employees.
- Implements query parameters for search (name, department, ID).

Database (MongoDB):
- Stores employee records with fields (id, name, department, designation, email, phone).
- Indexed fields for faster searches.

4. Component / Module Diagram

Frontend Components:
- Navbar → Navigation
- EmployeeList → Displays employees
- SearchBar → Filters employees
- EmployeeCard → Displays individual employee info
- EmployeeDetails → Detailed profile (optional)
- Footer → Static info

Backend Modules:
- server.js → Express setup
- routes/employees.js → Employee APIs
- models/employee.js → MongoDB schema

5. Basic Flow Diagram

System Flow:

User → Frontend (React) → Backend API (Node.js + Express) → Database (MongoDB)
↑ | ↓
| ■– Returns Employee Data ———■
←——— Filtered / Complete Employee List ————————————

Flow Description:
1. User loads app → Employee list fetched from backend API.
2. User searches employee by name/ID/department.
3. Search handled in frontend (filtering) or backend (API query).
4. Backend queries MongoDB and returns results.
5. Results displayed dynamically in frontend.