



COLLEGE CODE : 9623

COLLEGE NAME : Amrita College Of Engineering And Technology

DEPARTMENT : Computer Science And Engineering

STUDENT NM-ID : 12A034CC55E85E1BD8C491A91B2BC928

ROLL NO : 962323104010

DATE : 22-09-2025

Completed the project named as Phase-3 TECHNOLOGY



PROJECT NAME : Employee Directory Search


SUBMITTED BY,

NAME : R.N. Ahilesh Raja

MOBILE NO : 9003560915



Project Setup

Feature DesignProject SetupSelect a tech stack for the directory (e.g., React for frontend, Node.js for backend, and MongoDB/MySQL for data storage)  



Initialize the application repository using a version control system such as GitHub to track changes and manage collaboration 


Create a project directory structure, including separate folders for source code, assets, configuration, and tests.

Core FeaturesEmployee Profiles:

Each entry should contain name, job title, department, email, phone number, location, photo, and skills  

Search Functionality:

Implement text-based search that allows filtering by name, department, or other fields  


UI/UX ConsiderationsDesign a clean, intuitive interface using a component-based architecture.Prioritize usability—group employees by department, provide alphabetical filtering, and support mobile responsiveness 

Add list views and detail pages for individual employees with easy navigation.Page


Data Storage and Core Implementation

Data Storage

Local State:


Use React's state management for temporary data during development 


Database Integration:

Set up a database (MongoDB or MySQL). Define a schema for employee data with required fields like name, department, contact info, and profile picture 

API Layer:

Create RESTful API endpoints for Create, Read, Update, and Delete (CRUD) operations on employee data. Implementing Search Use API queries to enable searching by keywords and filters (e.g., department, location).

Optimize the search algorithm for speed and relevance, showing only relevant results and supporting partial matches. Consider adding advanced search features such as filtering by skills, designation, or location 

Security and Permissions Restrict data modification rights to authorized users (HR/admin) while allowing all employees to view directory data 

Implement authentication (email/password, SSO, or social login as needed) to protect sensitive employee information

Testing and Version Control
Testing Core Features
Write unit tests for UI components (employee cards, search bar, filters) using frameworks like Jest or Mocha ❖

Develop integration tests for API endpoints and data flows to ensure CRUD operations work smoothly ❖

Test edge cases, such as incorrect search queries, missing profile entries, and unauthorized user access.

Use manual and automated tests to validate the user experience and detect bugs before launch.
Version Control & Deployment
Maintain all code in a GitHub repository, using branches and pull requests for feature development ❖

Document all changes in the repository using commit messages and README files.
Set up CI/CD pipelines (e.g., GitHub Actions) for automated build, test, and deployment cycles.
Deploy the MVP on a cloud platform (e.g., Heroku, AWS, Azure) or within the company's intranet ❖❖

Future Enhancements
Add features such as org charts, digital business cards, and event notifications for a richer directory experience ❖

Enable bulk data import/export and advanced analytics reporting for HR ❖