



**COLLEGE CODE: 9623**

**COLLEGE NAME: AMRITA COLLEGE OF ENGINEERING AND  
TECHNOLOGY**

**DEPARTMENT: COMPUTER SCIENCE AND ENGINEERING**

**STUDENT NM-ID: 12A034CC55E85E1BD8C491A91B2BC928**

**ROLL NO: 962323104044**

**DATE: 6-10-2025**

**Completed the project named as Phase 5 TECHNOLOGY**

**PROJECT NAME : Employee Directory with Search**

**SUBMITTED BY,**

**NAME: K. JEGAN**

**MOBILE NO: 8428217644**

# Final Demo Walkthrough

## 1. LandingPage:

Displays all existing employees in a tabular format.

Search bar filters records dynamically.

## 2. LoginPage:

Only admins can access CRUD functions. Implemented using Flask-Login.

## 3. AddEmployeePage:

Form collects employee ID, name, department, designation, email, phone. Validations ensure data accuracy.

## 4. Edit/DeleteEmployee:

Admin can update existing records or remove entries.

## 5. SearchBar:

Real-time filtering using AJAX requests.

## 6. LogoutFeature:

Ends session securely and returns to login page. System Architecture

Architecture Type: MVC (Model–View–Controller)

Layer	Component	Description
-------	-----------	-------------

Model	Database (MySQL)	Stores employee records.
-------	------------------	--------------------------

View	HTML / CSS / JS	Displays dynamic pages to user.
------	-----------------	---------------------------------

Controller	Flask routes & API	Handles logic & communication.
------------	--------------------	--------------------------------

## Data Flow:

1. User requests an action via UI.

2. Flask routes process the request.

3. MySQL

4. Responses sent back as JSON → rendered on page.

Backend Implementation

## Flask App Structure:

/employee\_directory

```
├── app.py
├── static/
├── templates/
├── models/
├── routes/
├── database.py
└── config.py
```

### **Key APIs:**

GET /employees – List employees

POST/employees–Addemployee

PUT /employees/<id> – Update employee

DELETE/employees/<id>–Deleteemployee GET

/search?name=... – Search employee

LibrariesUsed:Flask,Flask-SQLAlchemy,Flask-Login,Jinja2

---

### **FrontendImplementation**

HomePage:Displayemployeetablewithsearchandpagination. Form

Pages: Bootstrap forms for adding/editing records.

### **JavaScriptFeatures:**

Fetch API calls to Flask endpoints.

DOMmanipulationforliveupdates.

Validation before submission.

CSSDesign:Clean,minimal,responsivelayout.

Icons/UI:FontAwesome+Bootstrapiconsforclarity.

### **ScreenshotsDescription:**

(Insertrealscreenshotslater–textplaceholdersbelow)

1. HomePage:Employeeelisttablewithsearchbar.

2. AddEmployee:Formforaddingnewemployee.

3. EditEmployee:Formpre-filledwithcurrentdata.

4. DeleteConfirmation:Popuptoconfirmdeletion.

5. LoginPage:Adminloginform.

6. Dashboard: Overview of total employees & departments.

7. Search Results: Filtered list appearing in real-time.

---

## Testing and Validation

### Unit Tests:

Verified CRUD endpoints (POST, GET, PUT, DELETE).

Tested authentication and session flow.

### Integration Tests:

Checked communication between frontend and backend APIs.

### UI Testing:

Manual testing on desktop + mobile browsers.

### Performance Check:

Search function tested on 1000+ records.

### Result:

All modules performed within acceptable limits.

---

## Challenges & Solutions

Challenge	Description	Solution
Database Connection Errors	Deployment caused environment issues	Used environment variables + SQLAlchemy pooling
Slow Search	Large dataset queries delayed results	Added indexes and query optimization
UI Scaling	Interface broke on mobile devices	Used Bootstrap grid system
Authentication Bugs	Session timeout & redirect issues	Integrated Flask-Login session handler
API Integration	Inconsistent JSON formats	Standardized API responses with schemas

---

## GitHub README & Setup Guide

### Setup Steps:

1. Git clone <https://github.com/jegankj864-create/EMPLOYEE-DIRECTORY-WITH-SEARCH->
2. `cd employee-directory`
3. `python -m venv venv`

4. venv\Scripts\activateorsourcevenv/bin/activate

5. pipinstall-r requirements.txt

6. ConfigureDBin.envor config.py

7. pythonsetup\_db.py

8. pythonapp.py

Deployment:

Push to GitHub → Deploy on Render / Heroku.

Setenvironmentvariables(DB\_URI,SECRET\_KEY).

Test live link.

---

### **FinalSubmissionandConclusion**

RepositoryLink: <https://github.com/jegankj864-create/EMPLOYEE-DIRECTORY-WITH-SEARCH->

Deployed URL:

<https://employee-directory-with-search-nine.vercel.app/>

Conclusion:

The project successfully meets the objective Of creating a searchable,scalable,and user-friendly employee directory.

Through modular design and FlaskAPIintegration,it demonstrates efficient data handling and real-time interaction.

Future improvements include advanced filters,role-based access,and integration with external HR API.

GITHUB: <https://github.com/jegankj864-create/EMPLOYEE-DIRECTORY-WITH-SEARCH->  
SCREENSHOT:

The screenshot shows a GitHub repository page for 'EMPLOYEE-DIRECTORY-WITH-SEARCH-' by user 'jegankj864-create'. The repository is public and has 20 commits. The file list includes CODE, PHASE 1, PHASE 2, PHASE 3, PHASE 4, index.html, script.js, and style.css. The README section is currently empty, with a prompt to 'Add a README'. The right sidebar shows repository statistics: 0 stars, 0 watching, 0 forks, and 4 deployments. The languages section shows CSS (46.3%), JavaScript (39.8%), and HTML (13.9%).

File	Commit Message	Time Ago
CODE	Delete CODE/TM	15 minutes ago
PHASE 1	Delete PHASE 1/TEMP	45 minutes ago
PHASE 2	Delete PHASE 2/TM	44 minutes ago
PHASE 3	Delete PHASE 3/TM	43 minutes ago
PHASE 4	Delete PHASE 4/TM	42 minutes ago
index.html	Add files via upload	16 minutes ago
script.js	Add files via upload	16 minutes ago
style.css	Add files via upload	16 minutes ago

**README**

**Add a README**

Help people interested in this repository understand your project by adding a README.

**Add a README**

**About**

[employee-directory-with-search-nine.ver...](#)

Activity

0 stars

0 watching

0 forks

**Releases**

No releases published

[Create a new release](#)

**Packages**

No packages published

[Publish your first package](#)

**Deployments** 4

Production 1 minute ago

[+ 3 deployments](#)

**Languages**

CSS 46.3% JavaScript 39.8% HTML 13.9%

The screenshot shows a VS Code editor with the 'index.html' file open. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6   <title>Employee Directory</title>
7   <link rel="stylesheet" href="style.css" />
8 </head>
9 <body>
10  <h1>Employee Directory</h1>
11
12  <div class="search-box">
13    <input type="text" id="searchInput" placeholder="Search employees by name or role..." />
14  </div>
15
16  <div class="directory" id="employeeDirectory">
17    <!-- Employee cards will be added dynamically -->
18  </div>
19
20  <script src="script.js"></script>
21 </body>
22 </html>
23
```

A notification at the bottom right states: 'Server is Started at port : 5500' with a 'Don't show again' button.

## WEBSITE:

