Project Overview

The Alumni Network Management System is a web-based application designed to help educational institutions maintain long-term relationships with their alumni. After graduation, colleges often lose structured communication with former students, leading to missed opportunities in networking, mentorship, career guidance, and institutional growth.

This system provides a centralized digital platform where alumni, current students, and college administrators can interact securely. Alumni can maintain professional profiles, share career opportunities, participate in events, and mentor students. Administrators can manage users, verify alumni authenticity, and analyze alumni engagement.

The application is developed using a modern full-stack architecture with React for the frontend, Spring Boot for the backend, and MySQL as the relational database.

Problem Statement

Most colleges lack an efficient and structured system to track and engage their alumni after graduation. Existing methods such as spreadsheets, social media groups, or manual records are unorganized, insecure, and difficult to maintain. As a result:

- Alumni data becomes outdated or inaccurate
- Students miss networking and mentorship opportunities
- Colleges struggle to organize alumni events and job postings
- There is no centralized platform for alumni–college interaction

The problem is to design and implement a secure, scalable, and user-friendly Alumni Network Management System that enables effective communication and engagement between alumni, students, and the institution.

Objectives

The main objectives of the project are:

- To provide a centralized platform for managing alumni information
- To enable secure registration and authentication for different user roles

- To allow alumni to share job opportunities and participate in events
- To facilitate mentorship interactions between alumni and students
- To provide administrators with control over user verification and system data
- To improve alumni engagement and long-term institutional connections

---

Tech Stack

The project uses the following technologies:

Frontend

- React.js
- HTML5, CSS3, JavaScript
- Axios for API communication

Backend

- Spring Boot (Java)
- Spring Security with JWT for authentication
- RESTful API architecture

Database

- MySQL
- JPA / Hibernate for ORM

Tools

- Git & GitHub for version control
- Postman for API testing
- Google Docs for documentation

System Architecture

The system follows a three-tier architecture:

1. Presentation Layer (Frontend)
    - Developed using React
    - Handles user interaction and UI rendering
2. Application Layer (Backend)
    - Developed using Spring Boot
    - Handles business logic, authentication, and authorization
    - Exposes REST APIs
3. Data Layer (Database)
    - MySQL relational database
    - Stores user data, profiles, jobs, events, and mentorship records

Communication between the frontend and backend is done using HTTP requests with JSON data format.

Functional Requirements

User Authentication

- Users can register and log in securely
- Role-based access for Admin, Alumni, and Student
- JWT-based authentication

Profile Management

- Alumni can create and update professional profiles
- Students can maintain academic profiles

Alumni Directory

- Users can search alumni by department, graduation year, skills, or company

Job and Internship Management

- Alumni can post job and internship opportunities
- Students can view and apply for jobs

Event Management

- Admin or alumni can create events
- Users can view and register for events

Mentorship Management

- Students can send mentorship requests to alumni
- Alumni can accept or reject requests

Admin Management

- Admin can verify alumni accounts
- Admin can manage users and view system statistics

Development Plan (8 Phases)

Phase 1: Requirement Analysis and Planning

- Understand system requirements
- Define user roles and modules
- Create wireframes

Phase 2: Frontend Authentication UI

- Design login and registration pages
- Implement role-based navigation

Phase 3: Frontend Core Modules

- Profile pages
- Alumni directory UI
- Job and event UI

Phase 4: Frontend API Integration Setup

- Axios configuration
- Protected routes
- Error handling

Phase 5: Backend Authentication and Security

- User registration and login APIs
- JWT implementation
- Role-based authorization

Phase 6: Backend Core Functionalities

- Profile management APIs
- Job, event, and mentorship APIs

Phase 7: Database Integration and Optimization

- Entity relationships
- Query optimization
- Data validation

Phase 8: Testing and Documentation

- API testing using Postman
- UI testing
- Final documentation and deployment preparation

Conclusion

The Alumni Network Management System provides an effective solution for maintaining long-term engagement between educational institutions and their alumni. By offering features such as alumni directories, mentorship programs, job postings, and event management, the system benefits students, alumni, and administrators alike.

The use of modern web technologies ensures scalability, security, and maintainability. This project demonstrates the practical application of full-stack development concepts, RESTful architecture, and database design, making it suitable as a comprehensive college-level software engineering project.

Future Enhancements

- Email and notification system
- Real-time chat between alumni and students
- Advanced analytics dashboard
- Mobile application support