

CERTIFICATE

This is to certify that the project entitled

“Alumni Portal”

Has been satisfactorily
completed by

1) Utkarsh Patil

2) Sujit Gade

3) Harshada Patil

4) Manisha Loni

5) Shivnath Desale

6) Akshay Patil

Under supervision and guidance for partial fulfilment of the

**PG Diploma in Advanced Computing (DAC) (24 weeks
FullTime) Course**

of

**Centre for Development of Advanced Computing (C-DAC),
Pune.**

at

Academy of Information

Technology (YCP) Nariman Point,

Mumbai –400 021.

Faculty

Course Co-ordinator

Batch: PG DAC Sep 2023 BatchDate: 23rd Feb 2024.

PREFACE

This project report presents the development of an Alumni Portal, a web application designed to connect and engage alumni with their alma mater and each other. The Alumni Portal serves as a bridge between the institution and its alumni, fostering a sense of community and providing a platform for networking, event management, and information sharing.

The primary goal of this project was to create an intuitive and user-friendly interface that allows alumni to easily access and interact with various features such as viewing events, exploring batches, and managing their profiles. The portal also facilitates administrators in managing events and maintaining up-to-date alumni records.

Developed using React JS, this project incorporates modern web development practices and technologies to ensure a responsive and dynamic user experience. The database design is carefully structured to store and manage alumni data efficiently, while the backend provides secure access and robust functionality.

The project is a culmination of the knowledge and skills acquired during the coursework, applying concepts from front-end development, database management, and user interface design. It demonstrates the practical implementation of these concepts in a real-world scenario, aiming to meet the needs of both alumni and the institution.

This report details the design, development, and testing phases of the project, highlighting the challenges encountered and the solutions implemented. It also provides insights into the project's scope, objectives, and future enhancements.

ACKNOWLEDGMENT

We, the project team, would like to express our heartfelt gratitude to everyone who contributed to the successful completion of our Alumni Portal project.

First and foremost, we would like to thank our project guide, Gaurav Talale, for their continuous support, expert guidance, and valuable insights. Their encouragement and constructive feedback were pivotal in shaping the direction of our project.

We are also grateful to AIT-YCP, I for providing us with the resources and a conducive environment to work on this project. The knowledge and skills we acquired during our coursework were essential in the successful development of this application.

We would like to extend our appreciation to our fellow classmates and friends who offered their support, shared their ideas, and provided constructive criticism, helping us refine our project.

A special thanks goes to each member of our project team—Sujit Gade, Harshada Patil, Shivnath Deale, MANisha Loni, and Akshay Patil—for their hard work, dedication, and collaboration. Each member brought unique strengths to the project, and it was our collective effort that enabled us to achieve our goals.

Lastly, we are deeply thankful to our families and loved ones for their constant encouragement, patience, and understanding throughout this journey. Their support was crucial in helping us stay focused and motivated.

This project is the result of the combined efforts and contributions of everyone involved, and we are sincerely thankful for all the support we received.

Alumni Portal

To everyone who has played a role, big or small, in making this project a reality
– thank you from the bottom of our hearts.

Team Members;

- 1) Utkarsh Patil
- 2) Sujit Gade
- 3) Harshada Patil
- 4) Manisha Loni
- 5) Shivnath Desale
- 6) Akshay Patil

INDEX

SR.NO	TITLE	PAGE NO
1.	Certificate	1
2.	Preface	2
3.	Acknowledgement	3
4.	Introduction	6
5.	Software Requirement	7
6.	Hardware Requirement	7
7.	ER Diagram	8
8.	Data Flow Diagram	9
9.	Use Case Diagram	10
10.	Backend Details	11
11.	Frontend Details	12
12.	Interface Home Page: Login Page: Alumni Data: Registration Page: News Page: Event Page: Social Page: Gallery Page:	13-16
13.	Coding	17-25
14.	Future Scope	26
15.	Conclusion	27
16.	Bibliography	28

INTRODUCTION

In today's digital age, the importance of maintaining strong connections between educational institutions and their alumni cannot be overstated. Alumni are not only ambassadors of their alma mater but also valuable resources for networking, mentoring, and support. To facilitate these connections and foster a thriving community, we have developed an Alumni Portal—a web-based platform designed to bring alumni and their alma mater closer together.

The Alumni Portal is a comprehensive solution that enables alumni to stay informed about institutional events, connect with fellow graduates, and contribute to the growth and development of the institution. The portal offers a user-friendly interface that allows alumni to easily navigate through various sections, including event listings, batch information, and personalized profiles. Additionally, it provides administrators with tools to manage events, update alumni records, and communicate effectively with the alumni community.

This project was undertaken as part of our group coursework, with the aim of applying theoretical knowledge to a real-world scenario. Utilizing React JS for front-end development, we focused on creating a responsive and dynamic user experience. The back-end was meticulously designed to ensure data security and efficient management of the alumni database.

The development of this Alumni Portal was a collaborative effort, with each team member contributing their expertise in different areas such as front-end design, back-end development, database management, and user interface design. The project not only allowed us to enhance our technical skills but also provided us with an opportunity to work as a cohesive team, overcoming challenges and finding innovative solutions.

In this report, we present a detailed account of the project, including the design and development process, the challenges faced, and the solutions implemented. We also discuss the potential future enhancements that could further improve the functionality and user experience of the Alumni Portal.

Software Requirements :

- Eclipse IDE
- MySQL
- React JS
- Browser
- Spring Boot Tool suite

Hardware Requirements :

- PC / Laptop
- Intel core i5
- 8 GB Ram
- 500 GB Storage

Tools Requirements :

- Postman
- Git
- GitHub

ER Diagram

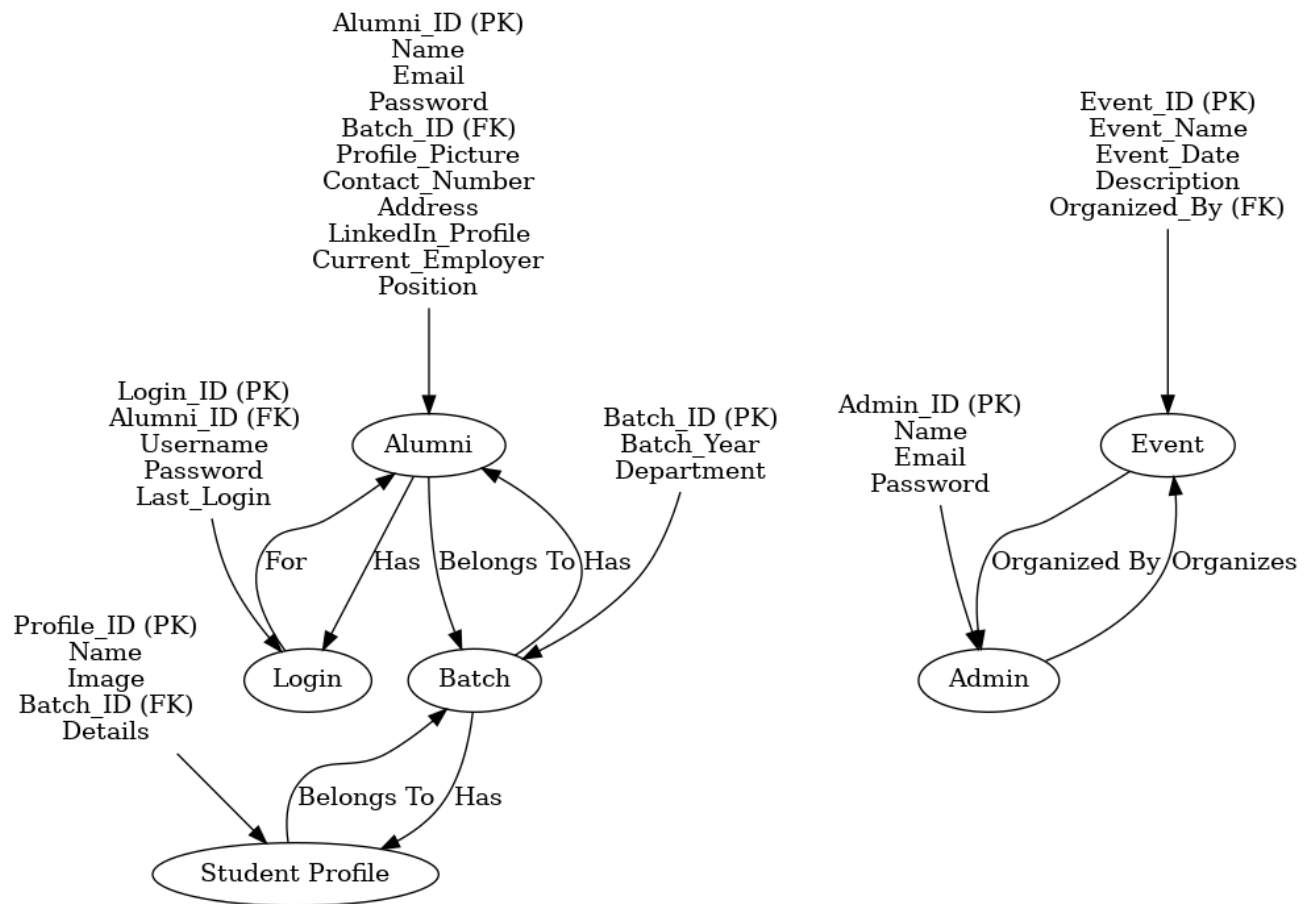


Diagram 1- ER Diagram

Data Flow Diagram

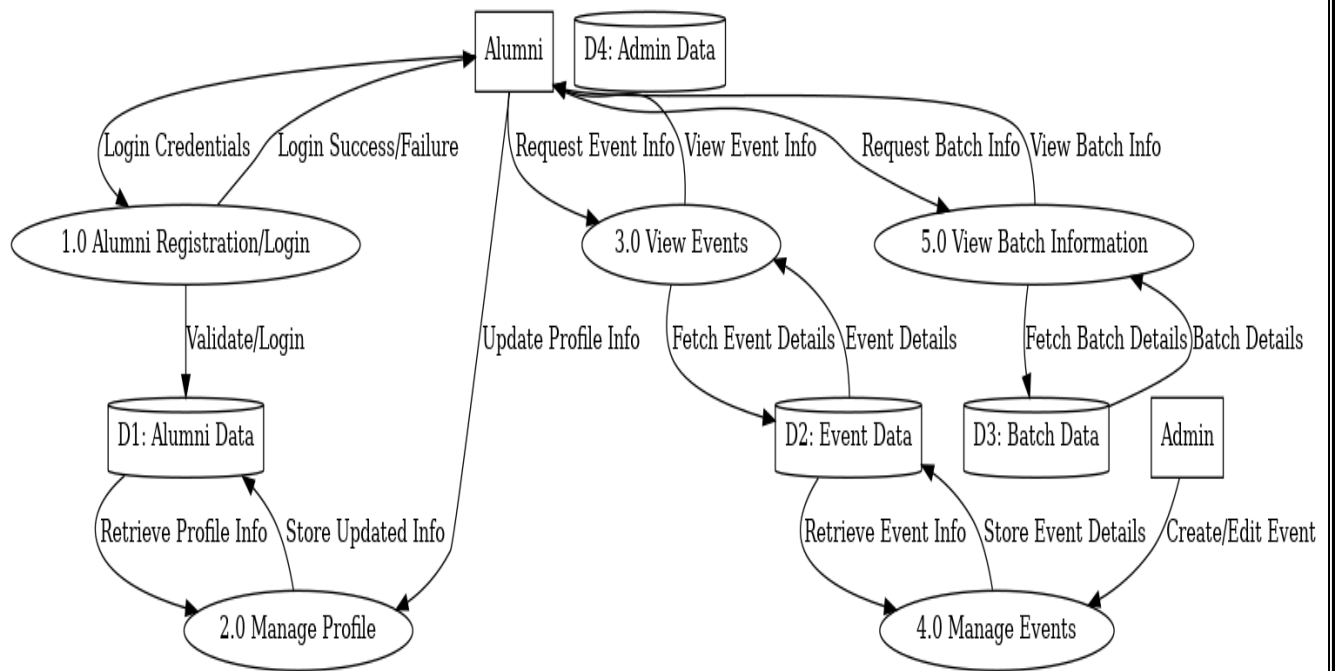


Diagram 2- Data Flow Diagram

Use case Diagram

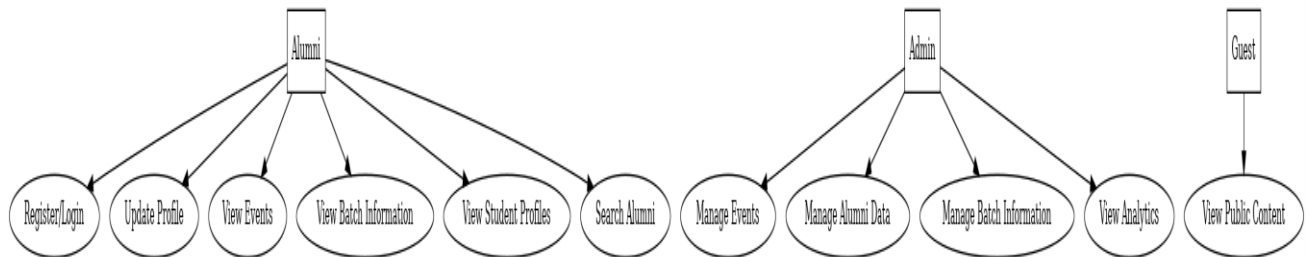


Diagram 3- Use Case Diagram

BACKEND DETAILS

What technology used?

- Spring Boot (Framework)
- MySQL (Database)

Dependencies:

- Spring web
- Spring-data-Jpa
- MySQL-connector
- Spring-security-core
- Thymeleaf

API End Points:

- Alumni Login Functionality
- User Login Functionality
- Alumni Registration Functionality
- User Registration Functionality

FRONTEND DETAILS

The alumni portal will have a responsive Home Page with a navbar for Events, Batches, and Login, featuring an image and student profile boxes. The Events Page will include an admin panel for managing events and a list of events with "More Details" links. On the Batches Page, users can click on batch years to see student profiles with "Details" buttons. The Alumni Profile Page will display detailed profiles in a card layout. The Login/Signup Page will feature validation forms. Use responsive design with React JS and styling libraries like Bootstrap or Material UI.

What libraries are used in React?

- React-router-Dom
- Axios

Alumni Login:

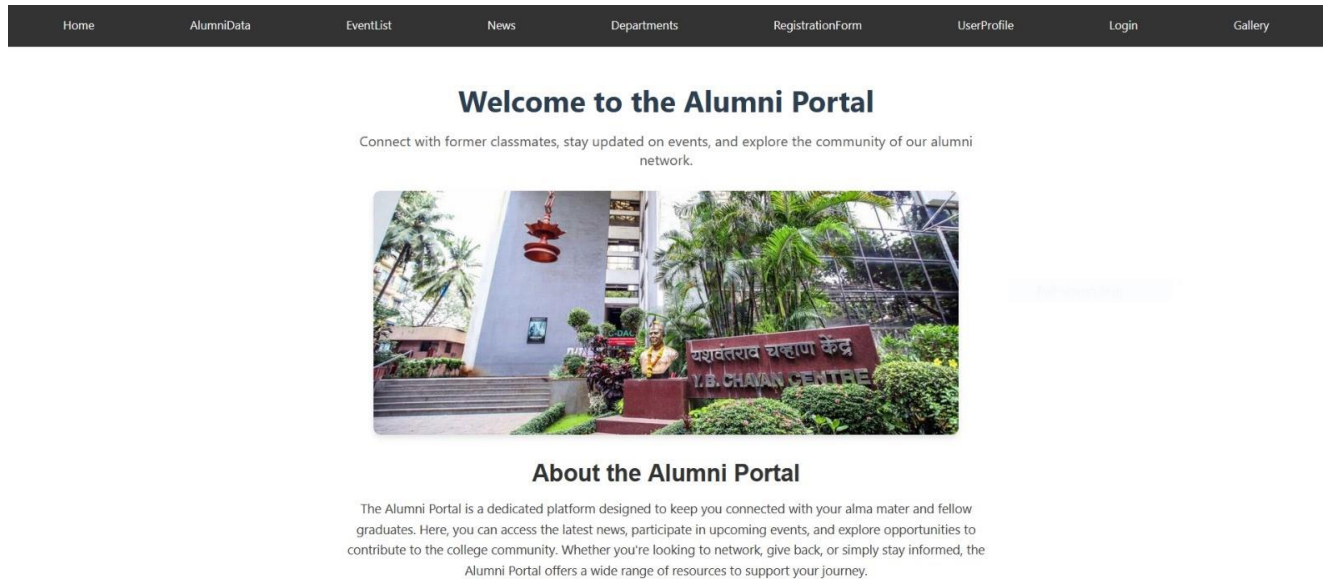
The Alumni Login React component is a crucial part of the authentication system for administrators in the web application. It handles login attempts, Super Password, and navigation.

User Login:

The User Login React component is a crucial part of the authentication system for administrators in the web application. It handles login attempts, Username and password.

Interface :

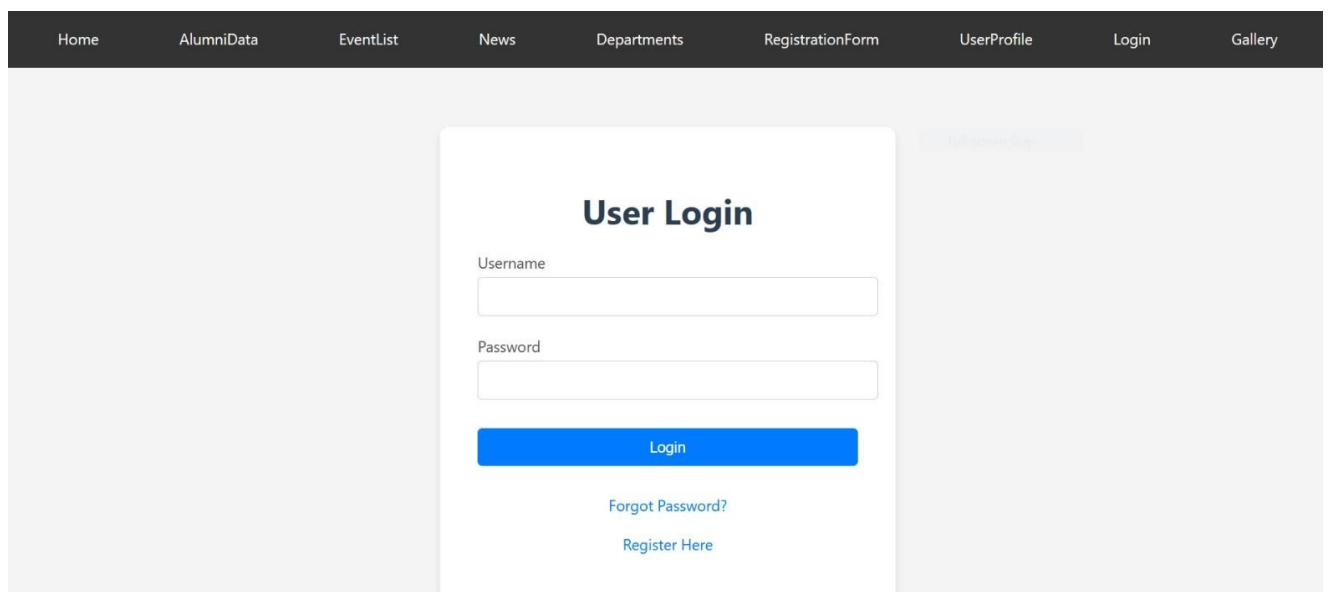
Home Page :



Screenshot-1

This is our Home Page you can see there is option for selection Home, Alumni Data, Event List, News, User Profile, Login, Gallery When you click on given navbar you can redirect to the Particular page.

Login Page :



Screenshot-2


Alumni Portal

This is user Login in page if user is already have an account then he/she can simply Sign in otherwise he/she can Create a new account"


Alumni Data :

[Home](#) [AlumniData](#) [EventList](#) [News](#) [Departments](#) [RegistrationForm](#) [UserProfile](#) [Login](#) [Gallery](#)


Alumni Data



Tejas Patil
Current Designation: Software Engineer at Google
Achievements: Developed scalable microservices, led the development team for major projects.
Additional Info: Graduated in 2015, B.Tech in Computer Science.



Dipakshi Deshmukh
Current Designation: Product Manager at Apple
Achievements: Led the design of the latest iPhone UI, managed a team of 20 designers.
Additional Info: Graduated in 2014, B.Tech in Information Technology.



Pramod Joshi
Current Designation: Data Scientist at Facebook
Achievements: Implemented machine learning algorithms, published papers on AI.
Additional Info: Graduated in 2016, M.Tech in Data Science.

Screenshot-3

This is the page on which Alumni Data you can see all alumnis and also you can check their current status and more details about him/her.

Registration Page :

[Home](#) [AlumniData](#) [EventList](#) [News](#) [Departments](#) [RegistrationForm](#) [UserProfile](#) [Login](#) [Gallery](#)

Registration Form

PRN (Permanent Registration Number):

First Name:

Last Name:

Email:

Password:

Confirm Password:

[Register](#)


Screenshot-4

New alumni can register first here then they are able to login as alumni.


News Page :

[Home](#) [AlumniData](#) [EventList](#) [News](#) [Departments](#) [RegistrationForm](#) [UserProfile](#) [Login](#) [Gallery](#)


YCP Mumbai News



YCP Mumbai Alumni Meet 2024
Saturday, August 10, 2024 - 10:00 AM
Join us for the YCP Mumbai Alumni Meet 2024. Reconnect with classmates, network with industry leaders, and celebrate the achievements of our alumni.



Launch of New Research Center
Thursday, July 25, 2024 - 2:00 PM
YCP Mumbai is proud to announce the inauguration of our new research center. This facility aims to foster innovation and support cutting-edge research initiatives.



YCP Mumbai Annual Sports Day
Friday, June 15, 2024 - 9:00 AM
The Annual Sports Day at YCP Mumbai was a tremendous success. Students participated in various sports events, showcasing their athletic skills and team spirit.

Screenshot-5


In news Page, you can check daily news about institute, So that you can aware of what is going in the institute.

Event Page :


[Home](#) [AlumniData](#) [EventList](#) [News](#) [Departments](#) [RegistrationForm](#) [UserProfile](#) [Login](#) [Gallery](#)

College Events


Stay updated with the latest events happening at our college.




Alumni Meet 2024
Date: September 25, 2024
Time: 10:00 AM - 2:00 PM
An annual gathering for alumni to reconnect with the college and network with fellow graduates.



Tech Innovation Workshop
Date: October 10, 2024
Time: 9:00 AM - 4:00 PM
A hands-on workshop focused on the latest trends and innovations in technology.



Cultural Fest 2024
Date: November 15, 2024
Time: 5:00 PM - 10:00 PM
A vibrant cultural festival showcasing the talents of students through music, dance, and art.

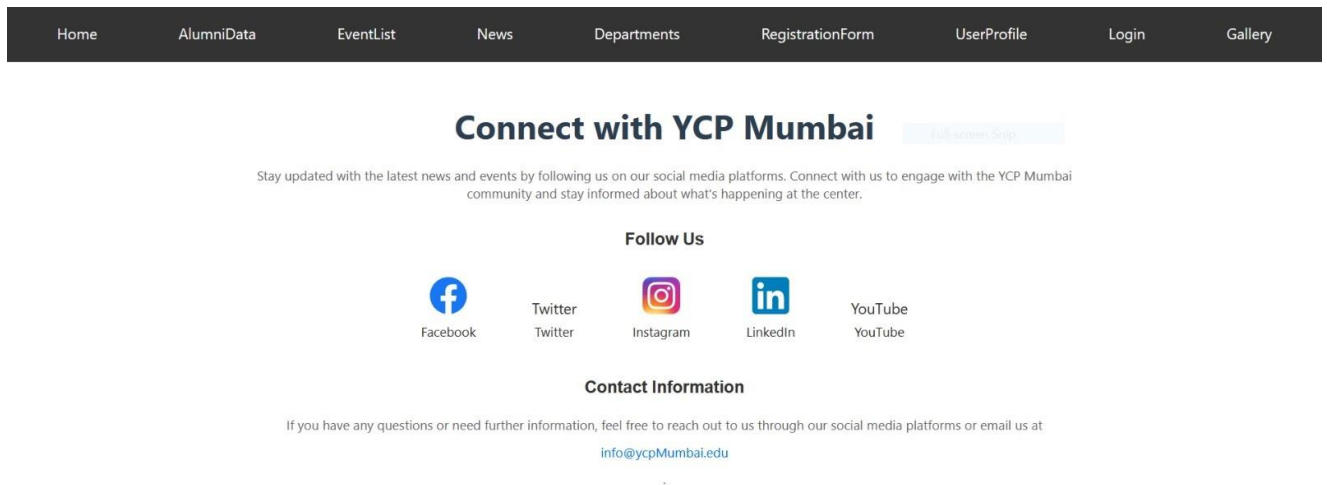


Guest Lecture Series: Leadership in the Modern World
Date: December 5, 2024
Time: 11:00 AM - 1:00 PM
A special guest lecture on leadership and its role in today's fast-paced world.

Screenshot-6

This is page for uploading events by admin and alumnis and users can check upcoming events and take participant in that event.

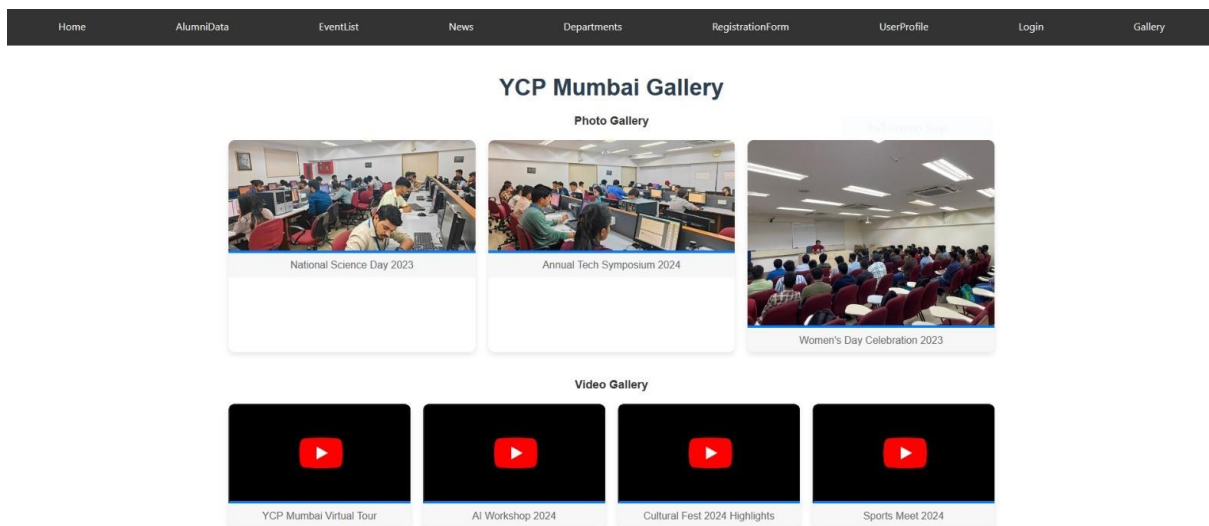
Social Page :



Screenshot-7

This page is dedicated to all social media site of institute. So anyone can reach their Social platform and take details.

Gallery Page :



Screenshot-8

This is a gallery page where everyone can see all events photos and videos in this page.

Coding :

Functions:

The alumni portal will include essential functions such as user authentication for login and signup, along with profile management to create, update, and view alumni profiles. Admins will use event management functions to add, edit, and delete events, while users can view event details. Additionally, search and filter functions will help users find specific alumni, events, or batches. These functions will streamline user interaction and data management across the platform. The system architecture consists of three interconnected layers:

1. Model:

- Manages alumni and admin data.
- Manages event data.
- Manages batch data and student lists.
- CRUD operations for handling data storage and retrieval..

2. View:

- Shows navbar, carousel, and profile boxes.
- Displays event list and admin controls.
- Shows batch years and student profiles.
- Displays detailed alumni profiles.
- Forms for user authentication.

3. Controller:

- Handles navigation and data for the home page.
- Manages event creation, editing, and retrieval.
- Handles batch selection and student profile display.
- Manages profile data and updates.
- Handles user login, signup, and authentication

Main Component of Backend:

Entry Point :

Entity :

```
package com.alumniportal.entity;

import java.time.Year;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.FetchType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.OneToOne;
import jakarta.persistence.Table;
import lombok.Getter;
import lombok.Setter;

@Entity
@Getter
@Setter
@Table(name = "Alumni_Details")
public class AlumniDetails {

    @Id
    @Column(name = "alumni_id")
    private Long alumniId;

    // UI Side restriction to check valid mobile no
    @Column(nullable = false, unique = true)
    private String email;

    @Column(name = "year_of_graduation", nullable = false)
    private Year yearOfGraduation;

    @Column(name = "prn_no", nullable = false, unique = true, length = 50)
    private String prnNo;

    @Column(nullable = false, length = 255)
    private String department;

    @Column(name = "current_location", length = 255)
    private String currentLocation;
```

Alumni Portal

```
@Column(length = 255)
private String companyName;

@Column(name = "profile_picture", length = 255)
private String profilePicture;

@Column(name = "current_designation", length = 255)
private String currentDesignation;

@Column(columnDefinition = "TEXT")
private String achievements;

@OneToOne(cascade = CascadeType.MERGE)
@JoinColumn(name = "alumni_id", referencedColumnName = "user_id")
private UsersDetails user;

}
```

Gateway Properties File :

```
spring.application.name=Project-On-Alumniportal

spring.datasource.url=jdbc:mysql://localhost:3306/aluminiPortaltwo?useSSL=false
spring.datasource.username=root
spring.datasource.password=DatabasePass@
#Sds@26969161

#spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
#logging.level.org.hibernate=DEBUG
```

User Controller:

```
package com.alumniportal.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.alumniportal.entity.UsersDetails;
import com.alumniportal.service.UserDetailsService;
```

Alumni Portal

```
@RestController
@RequestMapping("/user")
public class LoginController {

    @Autowired
    private UserDetailsService userServiceRepo;

    @PostMapping("/login")
    public List<UsersDetails> getUserByEmailAndPassword(@RequestBody
UsersDetails user) {

        return userServiceRepo.getUserByEmailAndPassword(user.getEmail(),
user.getPassword());
    }

    @PostMapping("/register")
    public UsersDetails insertUser(@RequestBody UsersDetails user) {
        return userServiceRepo.insertUser(user);
    }

    @PostMapping("/forget")
    public UsersDetails forgetPassword(@RequestBody UsersDetails user) {
        return userServiceRepo.forgetPassword(user);
    }

    @PostMapping("/update")
    public UsersDetails updateDetails(@RequestBody UsersDetails user) {
        return userServiceRepo.updateDetails(user);
    }
}
```

Alumni Controller :

```
package com.alumniportal.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
```

Alumni Portal

```
import org.springframework.web.bind.annotation.RestController;

import com.alumniportal.entity.AlumniDetails;

import com.alumniportal.entity.UsersDetails;

import com.alumniportal.service.AlumniDetailsService;

@RestController

@RequestMapping("/user")

public class AluminiController {

    @Autowired

    private AlumniDetailsService alumniDetailsService;

    //viewbyName

    @PostMapping("/getAllAlumni")

    public List<AlumniDetails> getUserByName(@RequestBody UsersDetails alumni) {

return alumniDetailsService.findAlumni(alumni);

    }

    @PostMapping("/AddAlumni")

    public AlumniDetails addAlumni(@RequestBody AlumniDetails alumni) {

        return alumniDetailsService.addAlumni(alumni);

    }

    @PostMapping("/UpdateAlumni")

    public AlumniDetails updateAlumni(@RequestBody AlumniDetails alumni) {

        return alumniDetailsService.updateAlumni(alumni);

    }

    @PostMapping("/DeleteAlumni")

    public String deleteAlumni1(@RequestBody AlumniDetails alumni) {

        return alumniDetailsService.deleteAlumni(alumni);

    }

}
```

Alumni Service:

```
package com.alumniportal.service;

import java.util.List;

import org.springframework.stereotype.Service;

import com.alumniportal.entity.AlumniDetails;
import com.alumniportal.entity.UsersDetails;

@Service
public interface AlumniDetailsService {

    // List<AlumniDetails> getAluminByEmailAndPassword(String email, String
    password);
    List<AlumniDetails> findAlumni(UsersDetails Alumni);

    AlumniDetails addAlumni(AlumniDetails alumni);

    AlumniDetails updateAlumni(AlumniDetails alumni);

    String deleteAlumni(AlumniDetails alumni);
}
```

User Service:

```
package com.alumniportal.service;

import java.util.List;

import com.alumniportal.entity.UsersDetails;

public interface UserDetailsService {

    List<UsersDetails> getUserByEmailAndPassword(String email, String password);

    UsersDetails insertUser(UsersDetails user);

    UsersDetails forgetPassword(UsersDetails user);

    UsersDetails updateDetails(UsersDetails user);

}
```

Alumni Service Implementation:

```
package com.alumniportal.service.impl;

import java.util.ArrayList;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.util.ObjectUtils;

import com.alumniportal.entity.AlumniDetails;
import com.alumniportal.entity.UsersDetails;
import com.alumniportal.repositories.AlumniDetailsRepo;
import com.alumniportal.repositories.UserDetailsRepo;
import com.alumniportal.service.AlumniDetailsService;

@Service
public class AlumniDetailsServiceimpl implements AlumniDetailsService {

    @Autowired
    public AlumniDetailsRepo AlumniDetailsRepo;

    @Autowired
    public UserDetailsRepo userDetailsRepo;

    // @Override
    // public List<AlumniDetails> getAluminByEmailAndPassword(String email, String
    // password) {
    //     return AlumniDetailsRepo.findByEmailAndPassword(email, password);
    // }

    @Override
    public List<AlumniDetails> findAlumni(UsersDetails alumni) {
        List<AlumniDetails> alumniList = new ArrayList<>();
        try {
            List<UsersDetails> userDetailsMap =
userDetailsRepo.findByFirstNameOrLastNameLike(alumni.getFirstName(),
alumni.getLastName());
            if (!ObjectUtils.isEmpty(userDetailsMap)) {
                alumniList = AlumniDetailsRepo
.findAllById(userDetailsMap.stream().map(UsersDetails::getUserId).toList());
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return alumniList;
    }
}
```

Alumni Portal

```
@Override
public AlumniDetails addAlumni(AlumniDetails alumni) {
    try {
        UsersDetails userDetails =
userDetailsRepo.findByEmail(alumni.getEmail());
        if (!ObjectUtils.isEmpty(userDetails)) {
            alumni.setAlumniId(userDetails.getUserId());
            AlumniDetailsRepo.save(alumni);
        }
        return alumni;
    } catch (Exception e) {
        e.printStackTrace();
        return new AlumniDetails();
    }
}

@Override
public AlumniDetails updateAlumni(AlumniDetails alumni) {
    return addAlumni(alumni);
}

@Override
public String deleteAlumni(AlumniDetails alumni) {
    String msg = "";
    try {
        if (AlumniDetailsRepo.existsById(alumni.getAlumniId())) {
            AlumniDetailsRepo.delete(alumni);
            msg = "Record Is Successfully Deleted";
        }
        msg = "No Record Found";
    } catch (Exception e) {
        msg = "Falid to delete Record";
    }
    return msg;
}
}
```

User Service Implementation:

```
package com.alumniportal.service.impl;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.util.ObjectUtils;
import com.alumniportal.entity.UsersDetails;
import com.alumniportal.repositories.UserDetailsRepo;
import com.alumniportal.service.UserDetailsService;
```



```
@Service
public class UserDetailsServiceImpl implements UserDetailsService {

    @Autowired
    public UserDetailsRepo userDetailsRepo;

    @Override
    public UsersDetails insertUser(UsersDetails user) {
        try {
            return userDetailsRepo.save(user);
        } catch (Exception e) {
            return null;
        }
    }

    @Override
    public List<UsersDetails> getUserByEmailAndPassword(String email, String
password) {
        return userDetailsRepo.findByEmailAndPassword(email, password);
    }

    @Override
    public UsersDetails forgetPassword(UsersDetails user) {
        UsersDetails findUser = null;
        try {
            findUser =
userDetailsRepo.findByEmailAndSecurityQuesAndSecurityAns(user.getEmail(),
user.getSecurityQues(),
                        user.getSecurityAns());
            if (!ObjectUtils.isEmpty(findUser)) {
                findUser.setPassword(user.getPassword());
                findUser = insertUser(user);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return findUser;
    }

    // update userdetails set user="";
    @Override
    public UsersDetails updateDetails(UsersDetails user) {
        return insertUser(user);
    }
}
```

Future Scope

The future scope for the alumni portal project is vast and can significantly enhance the platform's utility and engagement. Introducing networking features where alumni can connect, communicate, and create groups based on common interests or industries will foster a stronger sense of community. Adding a job board will enable alumni to post and apply for job opportunities, internships, and mentorship programs, further strengthening professional connections. Enhancements in event management, such as online event registrations, reminders, virtual event integration, and post-event feedback, will streamline event participation. The portal can also benefit from advanced search filters and analytics tools that allow users to find alumni based on specific criteria like skills or current job roles and track engagement metrics. Developing a mobile app will ensure easy access and a better user experience on smartphones and tablets. Additionally, integrating alumni contributions through content sharing, success stories, or donation features can enhance community involvement. Social media integration, particularly with platforms like LinkedIn, will simplify networking and profile updates. Finally, AI-powered recommendations can personalize the user experience by suggesting relevant events, job postings, and connections, making the portal a comprehensive tool for alumni engagement and professional development.

Conclusion

In conclusion, the alumni portal project represents a vital tool for maintaining and strengthening the connections between alumni and their alma mater. It is designed to serve as a central hub for alumni engagement, providing a seamless and user-friendly experience that encourages ongoing interaction within the community. The portal offers core functionalities such as user authentication, allowing secure access to personalized profiles where alumni can manage and update their information. The profile management system enables alumni to showcase their career progress, achievements, and personal updates, fostering a sense of pride and continuity within the community.

The event management feature is another cornerstone of the portal, facilitating the organization, promotion, and participation in various alumni events. This feature not only keeps alumni informed about upcoming events but also allows for easy registration and interaction, whether the events are held virtually or in person. The batch-specific features provide a targeted approach to engagement, allowing alumni to reconnect with peers from their specific graduating classes, which is essential for maintaining close-knit relationships over time.

Looking to the future, the portal has significant potential for expansion. Networking features can be introduced to allow alumni to connect based on shared professional interests, industries, or geographic locations. This could include direct messaging, group formation, and forums for discussion and collaboration. The addition of a job board would be a highly valuable feature, enabling alumni to post job opportunities, seek mentorship, and support each other in their professional journeys. This could also be expanded to include internships, volunteer opportunities, and collaborations on projects.

Advanced search capabilities and analytics tools would empower users to find specific alumni based on criteria like skills, job titles, or locations, making it easier to form connections and foster professional growth. A mobile app would ensure that the portal is accessible at all times, providing a convenient and optimized experience on smartphones and tablets, which is increasingly important in today's mobile-first world.

Furthermore, the integration of social media platforms like LinkedIn would simplify the process of keeping profiles up-to-date and expanding one's network, while AI-driven recommendations could personalize the user experience by suggesting relevant connections, events, and opportunities based on individual interests and activity. The potential for alumni contributions, such as sharing success stories, providing mentorship, or making donations, would further enrich the community and enhance the institution's reputation.

Bibliography:

We kept window open for further additional development.

- <https://react.dev/>
- <https://spring.io/projects/spring-boot>
- <https://www.w3schools.com/>