Campus Vault

for
Mini Project-II (K24MCA18P)
Session (2024-25)

Submitted by
Harsh Maheshwari
202410116100089
Harsh Sharma
202410116100092
Harshit Singh
20241011610092
Imran Ahmad
202410116100092

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Divya Singhal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206 (MARCH- 2025)

Declaration

We, the undersigned, hereby declare that the Mini Project titled "**Campus Vault**" is an original work completed by us as part of the curriculum requirement for the course under the Master of Computer Applications (MCA) program at **KIET Group of Institutions**.

We affirm that we have undertaken this during the academic year 2024-25 under the guidance of **Ms. Divya Singhal**. All the content and ideas presented in this report are the result of our own efforts, except where explicitly stated otherwise. Proper citations have been provided wherever references to external sources have been made.

We further declare that this project has not been submitted, either in part or in full, to any other university or institution for any degree or diploma**.**

Harsh Maheshwari

Harsh Sharma

Harshit Singh

Imran Ahmad

Certificate

This is to certify that Harsh Maheshwari (202410116100083), Harsh Sharma (202410116100084),Harshit Singh (202410116100089) and Imran Ahmad (202410116100092), have successfully completed their Mini Project - "Campus Vault" as part of the curriculum requirement for the Master of Computer Applications (MCA) program at KIET Group of Institutions, affiliated with Dr. A.P.J. Abdul Kalam Technical University (AKTU), Lucknow.

This project was carried out during the academic year 2024-25 under the esteemed guidance of Ms. Divya Singhal, Assistant Professor, Department of Computer Applications. The project embodies original work and studies conducted by the students, and its contents do not form the basis for the award of any other degree or diploma at this or any other institution.

We certify that the project meets the academic and technical requirements and is a valuable contribution in the field of certificate management and digital record-keeping.

Ms. Divya Singhal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad
Ghaziabad

Dr. Arun Kr. Tripathi
Dean
Department of Computer Applications
KIET Group of Institutions,

Campus Vault

Abstract

Campus Vault is a web-based platform designed to provide students with seamless access to their academic and event-related certificates. The system allows students to retrieve mock certificates, course certifications, and event or hackathon certificates while also accessing previous placement drive questions for interview preparation. The admin end manages certificate distribution by generating secure access links and enhancing security measures for stored documents. This project aims to replace traditional, inefficient certificate storage methods with a secure, scalable, and user-friendly solution.

Key features of the system include:

- A user-friendly interface built using React.js.
- Secure storage and retrieval of certificates via cloud services.
- Role-based access control to prevent unauthorized access.
- A repository of placement drive questions to assist students in interview preparation.
- Scalable architecture using Java, Spring Boot, and MongoDB for efficient performance.
- Students can upload their certificates to the college's Google Drive through a secure link.
- Students can review their certificates and placement questions for better preparation.

Keywords: Certificate Management, Web Application, React.js, Java, Spring Boot, MongoDB, Cloud Services

Acknowledgments

Success in life is never attained single-handedly. I extend my deepest gratitude to my project supervisor, Ms. Divya Singhal, for her invaluable guidance, encouragement, and insightful feedback throughout the development of this project. Her expertise and suggestions played a crucial role in shaping this work.

I am also immensely grateful to Dr. Arun Kumar Tripathi, Professor and Dean, Department of Computer Applications, for his continuous support, insightful comments, and administrative assistance, which have significantly contributed to the successful completion of this project.

Finally, I extend my heartfelt thanks to my family for their unwavering encouragement, moral support, and constant belief in my abilities. Their love and motivation have been the driving force behind my dedication and perseverance. Without their support, the successful completion of this project would not have been possible.

This project is a result of collective efforts, and I am truly grateful to everyone who has contributed to its success.

Name - Harsh Maheshwari

Roll No - 202410116100083

Name - Harsh Sharma

Roll No-202410116100084

Name - Harshit Singh

Roll No-20241011610089

Name- Imran Ahmad

Roll No - 202410116100092

TABLE OF CONTENTS

5.Introduction

5.1 Overview

In today’s digital era, managing academic and extracurricular certifications efficiently has become a necessity for students and institutions alike. Traditional paper-based systems are cumbersome, prone to loss or damage, and often lack accessibility when needed. Campus Vault aims to address these challenges by providing a secure, cloud-based platform where students can store and retrieve their certificates conveniently.

Campus Vault serves as a centralized repository that not only helps students access their academic and event-related certificates but also provides a valuable resource for placement preparation. By integrating modern web technologies such as React.js, Java, Spring Boot, MongoDB, and cloud services, the system ensures a scalable, secure, and efficient solution for certification management. The platform is designed to benefit students by enabling easy certificate retrieval, assisting in interview preparation, and allowing them to upload certificates directly to the college's Google Drive. Additionally, students can review their certificates and placement questions for better preparation. The admin panel facilitates seamless document management and security enforcement.

This project aims to revolutionize the way academic certifications are handled, offering a user-friendly interface, role-based access control, and enhanced security features to prevent unauthorized access. Campus Vault provides a modern and efficient approach to certificate management, helping institutions and students streamline administrative workflows and focus on academic and professional growth.

6.Literature Review

6.1 Evolution in Digital Certification

In the past, educational institutions relied on physical certificates to validate student achievements. However, with advancements in digital technology, institutions have begun shifting towards digital certification systems. Digital certificates offer greater accessibility, security, and ease of verification, eliminating the risks associated with traditional paper-based certificates, such as damage, forgery, or loss.

The adoption of digital certificates has been further propelled by the rise of blockchain technology, which provides tamper-proof and verifiable credentials. Universities and organizations worldwide are increasingly using digital credentialing platforms to issue, manage, and verify certificates, ensuring authenticity and long-term accessibility.

6.2 Role of Modern Web Technologies

Modern web technologies have significantly improved the way digital certificates are managed and distributed. Technologies like cloud storage, encryption, blockchain, and web-based platforms have made it possible to store, access, and verify certificates securely. Frameworks like React.js, Java, and Spring Boot allow for efficient, scalable, and user-friendly platforms, ensuring seamless interaction between students and administrators.

Cloud-based solutions provide centralized storage, allowing students to access their certificates from anywhere, at any time. Moreover, encryption and security protocols safeguard the authenticity of digital documents, preventing fraud and unauthorized modifications. These advancements have made certificate management systems more reliable and efficient.

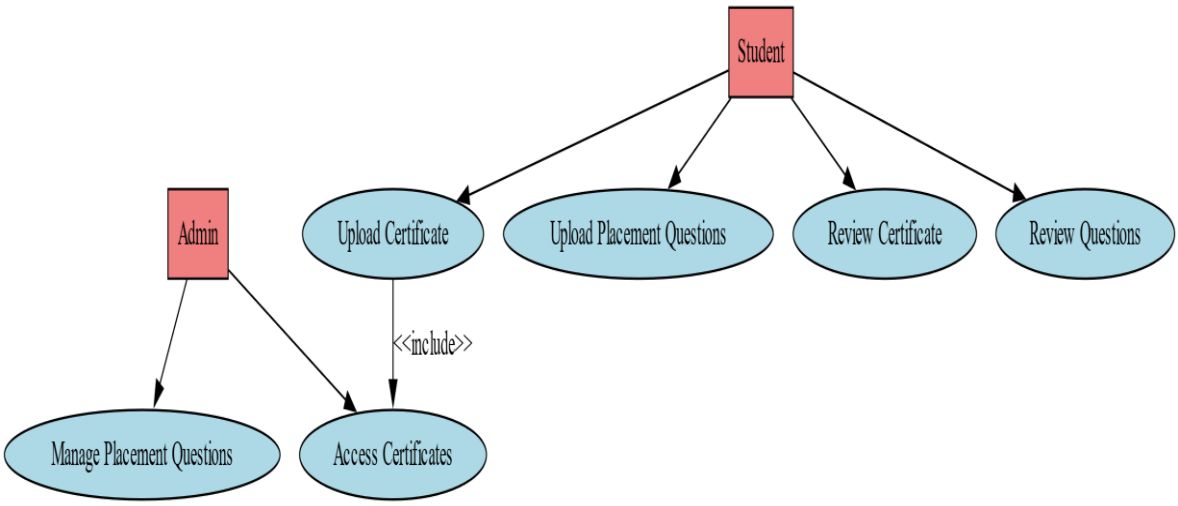
6.3 Existing System

Many institutions still rely on outdated certificate management systems, making it difficult for students to retrieve their documents when needed. Existing systems often involve manual handling, which can lead to inefficiencies, security risks, and accessibility issues. Institutions that lack a digital certification platform often require students to physically visit offices, fill out paperwork, and wait for processing, leading to delays and inconvenience.

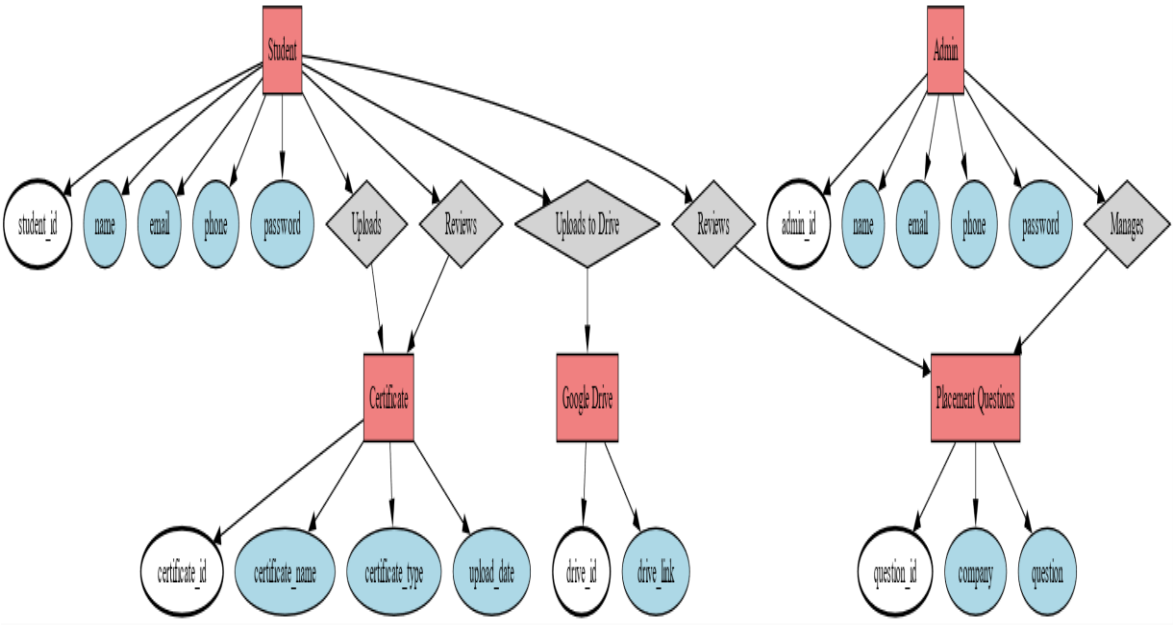
With Campus Vault, we aim to address these limitations by providing a streamlined, automated, and secure digital certification solution that enhances accessibility, security, and usability. By leveraging cloud computing and modern web technologies, Campus Vault ensures that students and administrators can efficiently manage and access certificates with minimal effort.

7. Project Flow

7.1 Use Case Diagram



7.2 ER Diagram

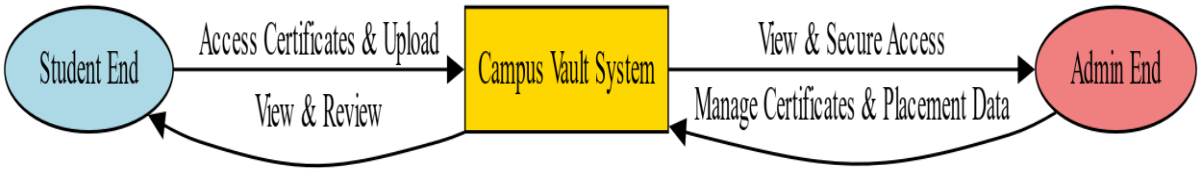


7.3 Data Flow Diagram

7.3.1 Level 0 (Context Diagram):

Entities in Level 0 DFD:

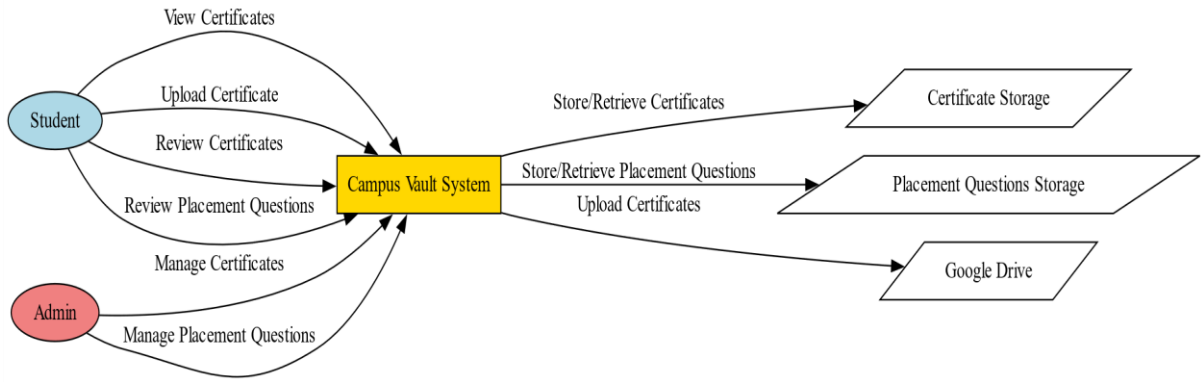
1. Student (External Entity)
    - Interacts with the system to access and upload certificates, and review placement questions.
2. Admin (External Entity)
    - Manages certificates, placement questions, and provides Google Drive links for students to upload their documents.
3. Campus Vault (System - Central Processing Unit)
    - Centralized system for processing and managing certificates, placement questions, and user authentication.



7.3.2 Level 1(Context Diagram):

Entities in Level 1 DFD:

1. Student End:
    1. View Certificates
    2. Upload Certificates (via Google Drive Link)
    3. Review Certificates
    4. Review Placement Questions
2. Campus Vault (System):
    1. Stores Certificates
    2. Manages Placement Questions
    3. Provides Secure Access
3. Admin End:
    1. Access Certificates
    2. Manage Placement Questions



7.4 System Architecture:

Homepage (Login Page)

- Users: Student & Admin
- Features: Login, Password Reset, Signup

Student Dashboard

- Features: View & Download Certificates, Upload Certificates, Review Certificates, Review Placement Questions

Upload Certificate Page

- Feature: Upload certificates via Google Drive link

Admin Dashboard

- Features: Manage Certificates, Manage Placement Questions, Provide Google Drive Link

View Certificates Page

- Features: Display and verify stored certificates

7.5 Database Schema

1.Users Table

Stores user information.

Column Name	Data Type	Constraints
User id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	UNIQUE, NOT NULL
password	VARCHAR(255)	NOT NULL
role	ENUM	('student', 'faculty', 'admin') NOT NULL

2.Documents Table

Stores uploaded documents.

Column Name	Data Type	Constraints
Document id	INT	PRIMARY KEY, AUTO_INCREMENT
User id	INT	FOREIGN KEY REFERENCES Users(user_id) ON DELETE CASCADE
title	VARCHAR(255)	NOT NULL
File path	VARCHAR(255)	NOT NULL
Uploaded at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

3.Permissions Table

Manages document access.

Column Name	Data Type	Constraints
Permission id	INT	PRIMARY KEY, AUTO_INCREMENT
Document id	INT	FOREIGN KEY REFERENCES Documents(document id) ON DELETE CASCADE
User id	INT	FOREIGN KEY REFERENCES Users(user id) ON DELETE CASCADE
Access type	ENUM	('view', 'edit', 'download') NOT NULL

4. Categories Table

Stores document categories.

Column Name	Data Type	Constraints
Category id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(255)	UNIQUE, NOT NULL

5. Document Categories Table

Associates documents with categories.

Column Name	Data Type	Constraints
Document id	INT	FOREIGN KEY REFERENCES Documents(document id) ON DELETE CASCADE
Category id	INT	FOREIGN KEY REFERENCES Categories(category id) ON DELETE CASCADE

6.Notifications Table

Stores notifications for users.

Column Name	Data Type	Constraints
Notification id	INT	PRIMARY KEY, AUTO_INCREMENT
User id	INT	FOREIGN KEY REFERENCES Users(user id) ON DELETE CASCADE
message	TEXT	NOT NULL
status	ENUM	('unread', 'read') DEFAULT 'unread'

7.6 System Architecture

Frontend: Developed using React.js, providing an intuitive and interactive user interface for students and admins.

Backend: Powered by Java and Spring Boot, handling business logic, authentication, and data processing through RESTful APIs.

Database: MongoDB serves as the database for storing certificates, user credentials, placement questions, and review data.

Cloud Storage: Google Drive API is integrated to allow students to upload and retrieve certificates securely.

Implementation

8.1 Technology Stack

The Campus Vault system leverages modern technologies to ensure efficiency, security, and scalability:

- Frontend: React.js (for a responsive and interactive UI)
- Backend: Java with Spring Boot (for robust backend processing)
- Database: MongoDB (for efficient document-based storage)
- Cloud Storage: Google Drive API (for secure certificate storage and retrieval)
- Authentication: JWT (JSON Web Token) for role-based access control
- Deployment & Hosting**:** Cloud services such as AWS/GCP/Azure

8.2 Features

Campus Vault provides the following key features:

Student Features:

- View & Download Certificates: Students can access and retrieve their certificates anytime.
- Upload Certificates via Google Drive Link: Secure integration with Google Drive for uploading documents.
- Review Certificates: Students can verify their certificates for correctness.

- Review Placement Questions: A repository of previous placement questions for preparation.

Admin Features:

- Manage Certificates: Upload, approve, and distribute certificates securely.
- Manage Placement Questions: Maintain and update placement-related questions.
- Provide Google Drive Link: Securely share upload links for certificate submission.

8.3 Challenges (23-26)

During the development of Campus Vault, several challenges were encountered:

1. Secure Storage & Retrieval
    - Ensuring the confidentiality and integrity of certificates stored in the cloud.
    - Implementing proper encryption for document safety.
2. Role-Based Access Control
    - Differentiating between student and admin access while maintaining security.
    - Using JWT for authentication without compromising usability.
3. Data Consistency & Performance
    - Ensuring MongoDB handles large amounts of student data efficiently.
    - Optimizing database queries to avoid performance bottlenecks.
4. User Experience & Interface
    - Designing a simple, user-friendly interface with React.js.
    - Providing a smooth and intuitive experience for students and admins.
5. Integration with Google Drive
    - Setting up API authentication and handling file uploads securely.
    - Managing access permissions dynamically for different users.

9. Project Outcome

The Campus Vault project successfully addresses challenges in certificate management and placement preparation:

- Efficiency: Students can easily access and upload certificates without paperwork delays.
- Security: Cloud-based storage ensures data integrity, with proper authentication controls.
- Scalability: The system can be expanded to accommodate more features in the future.
- Improved Placement Readiness: Students have access to a growing repository of placement questions.
- Simplified Admin Workflow**:** Admins can efficiently manage certificates and student queries.

10. Conclusion

Campus Vault is a secure, scalable, and efficient platform that modernizes certificate management for educational institutions. By integrating React.js, Java Spring Boot, MongoDB, and Google Drive APIs, the system ensures seamless access, security, and usability. The project successfully replaces traditional manual processes with a digital, automated solution, improving both student experience and administrative efficiency.

Future enhancements may include:

- Blockchain-based certificate verification for enhanced security.
- AI-powered resume builder integrated with certification records.
- More cloud storage options beyond Google Drive.

11.Refereneces

React.js Documentation: https://react.dev

Spring Boot Documentation: https://spring.io/proiects/spring-boot

MongoDB Documentation: https://www.mongodb.com

Google Cloud: https://cloud.google.com