

**PROJECT REPORT**

## On

A college website

Submitted in partial fulfilment of the requirement for the Course BEE (22CS026) of

**COMPUTER SCIENCE AND ENGINEERING**

**B.E. Batch-2022 in**

**Dec -2024**

|  |  |
| --- | --- |
| **Under the Guidance of:** - | **Submitted By:** - |
| Rahul Singh Rajput | Pranzal yadav |
| Project Supervisor | Roll No.: 2210992067 |
|  | Prajwal |
|  | Roll No.: 2210992057 |
|  |  |
|  |  |
|  |  |
|  |  |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CHITKARA UNIVERSITY**

**PUNJAB**

# CERTIFICATE

This is to be certified that the project entitled “A College website” has been submitted for the Bachelor of Computer Science Engineering at Chitkara University, Punjab during the academic semester July 2024 – December 2024 is a bona fide piece of project work carried out by “pranzal yadav (2210992067), Prajwal (2210992057),” towards the fulfilment for the award of the course Integrated Project (CS 203) under the guidance of “Rahul Singh Rajput” and supervision.

**Signature:**

Rahul Singh Rajput Project Supervisor (BE-CSE)



# CANDIDATE’S DECLARATION

We, Pranzal yadav (2210992067), Prajwal (2210992057), B.E.-2022 of the Chitkara University, Punjab hereby declare that the Integrated Project Report entitled “A College Website ” is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other course.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature:** | **Signature:** |  |  |
| Pranzal yadav | Prajwal |  |  |
| 2210992067 | 2210992057 |  |  |

Place:

Date:

# ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing me an opportunity to undergo Integrated Project as the part of the curriculum.

We are thankful to “Rahul Sir” for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improve our quality of work.

|  |  |  |
| --- | --- | --- |
| Pranzal yadav | Prajwal |  |
| 2210992067 | 2210992057 |  |



# Table of Contents

|  |  |  |
| --- | --- | --- |
| **S. L. No.** | **Topics** | **Page No.** |
| 1 | Abstract | 6 |
| 2 | Introduction | 7 |
| 2.1 | Background | 7 |
| 2.2 | Problem Statement | 7 |
| 3 | Software and Hardware Requirement Specification | 8 |
| 3.1 | Methods | 8 |
| 3.2 | Programming/Working Environment | 8 |
| 3.3 | Requirements to Run the Application | 8 |
| 4 | Database Analysing, Design and Implementation | 9 |
| 5 | Program’s Structure Analysing and GUI Constructing | 9-11 |
| 6 | Code-Implementation and Database Connections | 12-15 |
| 7 | System Testing | 16 |
| 8 | Limitations | 16-17 |
| 9 | Conclusion | 17 |
| 10 | Future Scope | 17 |
| 11 | Bibliography/References | 17 |



1. **Abstract**

The **College Website** is a comprehensive backend engineering project designed to streamline and enhance the academic and administrative processes of a college. This website serves as a centralized platform, offering features such as categorized course listings, detailed subject information, and personalized student and faculty portals.

Built on a robust backend architecture, the website facilitates efficient data management, secure user interactions, and seamless navigation. Key functionalities include dynamic subject and course categorization, a centralized system for news and announcements, and interactive tools for enhanced engagement.

This project showcases the practical implementation of backend technologies, including database design, API integration, and server-side programming. The **College Website** demonstrates how backend engineering can create scalable and user-friendly solutions tailored to the needs of the educational sector.



# Introduction

## Background

1. The **College Website** is a feature-rich project developed as part of a backend engineering initiative, showcasing the practical application of backend development skills. This website is designed to cater to the diverse needs of a college community, providing an intuitive and efficient platform for academic and administrative management.

## Problem Statement

. In today’s digital era, educational institutions often face challenges in managing and disseminating information efficiently. Colleges require a centralized platform to handle a wide range of tasks, including organizing course categories, providing detailed subject information, and facilitating communication between students and faculty.

Many existing solutions lack personalization, scalability, and seamless integration of academic resources, leading to inefficient workflows, fragmented communication, and a poor user experience. Additionally, there is a need for a secure system to manage sensitive data, such as student records and faculty resources, while ensuring accessibility and ease of use.

This project addresses these issues by developing a **College Website** that serves as an all-in-one platform, leveraging backend engineering principles to create a scalable, secure, and user-friendly solution for modern educational institutions.



# Software and Hardware Requirement Specification

## Methods

**A College Website** employs a cloud-based architecture utilizing the MERN stack to enable scalable and secure real-time communication. The application is designed to be accessible through web portals and mobile applications, ensuring user-friendly interaction across multiple devices.

## Programming/Working Environment

The development environment for **A college website** includes:

* **Frontend**: HTML, CSS, JavaScript, and React.js
* **Backend**: Node.js with Express framework
* **Database**: MongoDB for data storage
* **Real-Time Communication**: Socket.io for real-time messaging
* **Cloud Services**: AWS or Azure for hosting and storage
* **IDE**: Visual Studio Code
* **Version Control**: Git for source code management

## Requirements to Run the Application

**Hardware Requirements**:

* A computer or server with at least 8GB RAM and 4 CPU cores
* Stable internet connection

**Software Requirements**:

* Node.js and npm installed
* MongoDB database server
* Web browser (latest version)
* Mobile platforms (iOS and Android) for mobile app deployment



# Database Analysing, Design and Implementation

**A College Website U**tilizes MongoDB for its database, which allows flexible storage of user data and messages. The database schema includes collections for users, conversations, messages, and notifications. Each document contains fields such as:

* **Users Collection**:
  + User ID
  + Username
  + Password (hashed)
  + Profile information (e.g., avatar, status)
  + Contact list
* **comments Collection**:
  + Message ID
  + Sender ID
  + Conversation ID
  + Content (text, images, files)
  + Timestamp
  + Read receipts

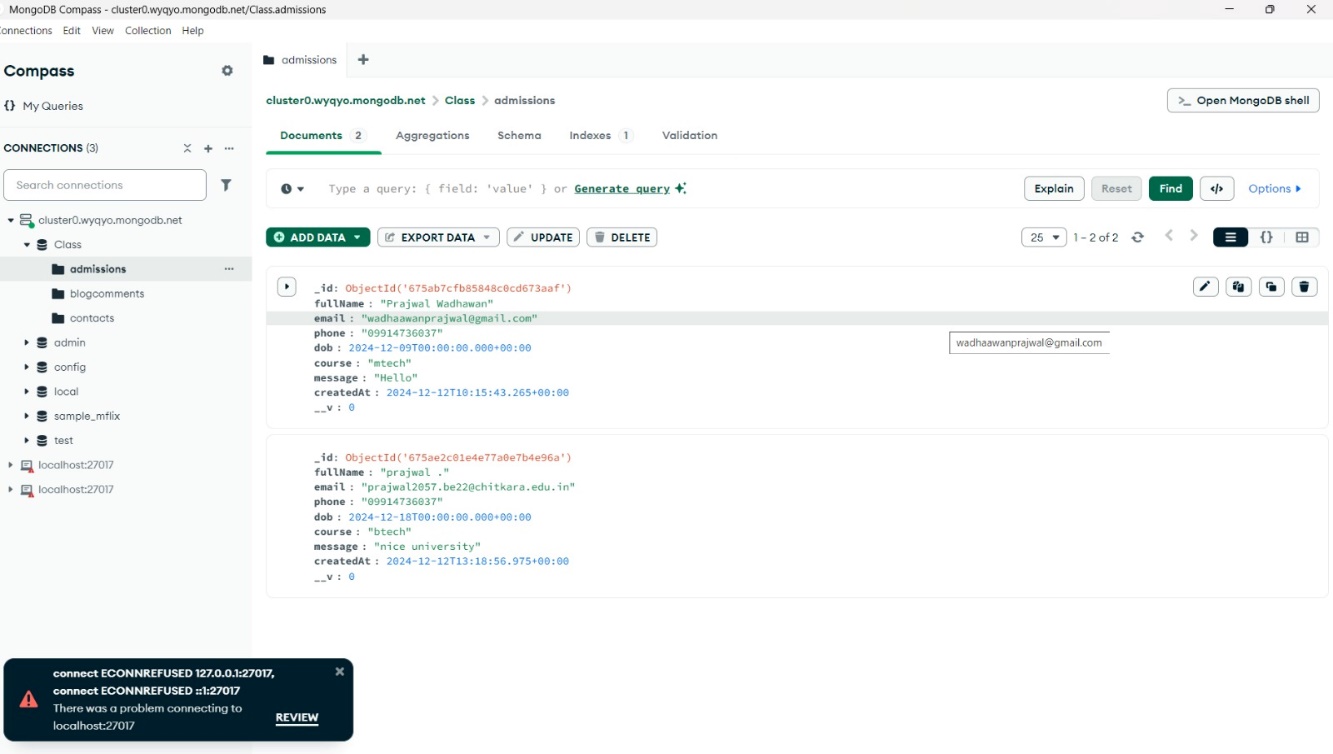
# Program’s Structure Analysing and GUI Constructing

The application is structured into several modules:

* **User Authentication**: Handles login, registration, and authentication for users.
* **Comment Management**: Allows users to create, and manage comments .
* **User Profile**: Provides an overview of user information and allows customization of profile settings.

# Code-Implementation and Database Connections

The backend uses Node.js and Express.js to handle API requests and manage real-time communication with Socket.io. MongoDB is accessed using the Mongoose library, which simplifies interactions with the database.

* 



# System Testing

Testing of **A College Website** included three main types: unit testing, integration testing, and user acceptance testing (UAT).

* **Unit Testing** focused on verifying individual components of the application, such as user authentication, message handling, and database operations, to ensure each part functions correctly.
* **Integration Testing** assessed how well different components work together within the overall system, ensuring seamless interaction between the frontend, backend, and real-time communication modules.
* **User Acceptance Testing (UAT)** involved gathering feedback from real users to ensure the application meets their needs and expectations. This phase included testing the user interface, ease of use, responsiveness, and overall user experience.

This comprehensive testing process helped identify and resolve issues, ensuring a smooth and effective user experience. Tools such as Jest for unit testing, Postman for API testing, and manual testing for UAT were utilized to ensure the application's reliability and performance.

# Conclusion

The **College Website** project is a robust and scalable solution designed to address the academic and administrative challenges faced by educational institutions. By integrating features like course categorization, subject management, and personalized user portals, the platform enhances efficiency, accessibility, and user engagement. With its strong backend foundation and scope for future expansion, this project demonstrates the potential to evolve into a comprehensive digital hub, transforming the way colleges operate and interact with their communities.

# Future Scope

#  Advanced Personalization:

# Implement AI-driven recommendation systems to suggest courses, resources, and career paths based on user interests and academic performance.

#  Integration with Learning Management Systems (LMS):

# Seamlessly connect with popular LMS platforms to facilitate online learning, assignment submissions, and grading.

#  Mobile Application:

# Develop a companion mobile application to provide on-the-go access to the platform’s features, enhancing accessibility and user engagement.

#  Data Analytics and Reporting:

# Integrate advanced analytics tools to provide insights into student performance, course popularity, and faculty engagement.

# Generate automated reports for administrative decision-making.

#  Real-Time Communication Tools:

# Add chat functionalities, discussion forums, and video conferencing tools to enhance collaboration between students and faculty.

#  Dynamic Event Management:

# Create modules for event registration, calendar integration, and automated reminders for college activities.

#  Multi-Language Support:

# Incorporate multi-language functionality to make the platform accessible to users from diverse linguistic backgrounds.

#  Enhanced Security Features:

# Strengthen the security architecture with features like multi-factor authentication, role-based access control, and encryption for sensitive data.

#  Integration with External APIs:

# Enable integration with third-party services such as payment gateways for fee management and external libraries for additional resources.

#  Cloud-Based Scalability:

# Transition to cloud-based infrastructure to support scalability, ensuring high performance even with increased user loads.

# References

 **W3Schools**: <https://www.w3schools.com>

 **GeeksforGeeks**: <https://www.geeksforgeeks.org>

 **GitHub**: <https://github.com>

 **MERN Stack Documentation**: <https://mern.io>

 **Socket.io Documentation**: https://socket.io/docs/v4

 **React.js Documentation**: https://reactjs.org/docs/getting-started.html

 **Node.js Documentation**: https://nodejs.org/en/docs/

 **Express.js Documentation**: https://expressjs.com/en/4x/api.html

 **MongoDB Documentation**: <https://docs.mongodb.com>