SHREE L. R. TIWARI COLLEGE OF ENGINEERING,

(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)

MIRA ROAD (E) – 401 107

Academic Year 2024–2025

"CAMPUS COLLAB"

Project Synopsis

Submitted in Partial Fulfillment of the Requirement for the Degree of

BACHELOR OF ENGINEERING

(Electronics and Telecommunication Engineering)

MUMBAI UNIVERSITY

By

Piyush Jha

Sahil Wadhwana

Nitish Choudhary

Satyam Jha

Under the Guidance of

Dr. Sheetal Mahadik

Department of Electronics and Telecommunication Engineering

SHREE L. R. TIWARI COLLEGE OF ENGINEERING

Department of Electronics and Telecommunication Engineering

CERTIFICATE OF APPROVAL

for

Project Synopsis

This to certify that

Name 1 Piyush Jha

Name 2 Sahil Wadhwana

Name 3 Nitish Choudhary

Name 4 Satyam Jha

have satisfactorily carried out the Project work entitled **"CAMPUS COLLAB"** in partial fulfillment of Bachelor of Engineering in Electronics and Telecommunication Engineering as laid down by University of Mumbai during the academic year 2022-2023.

Internal Guide:
Dr. Sheetal Madik

External Examiner:

Head of Department:
Mrs. Aboli Moharil

Principal:
Dr. Deven Shah

ACKNOWLEDGMENTS

Special thanks to our Guide ***Dr.Sheetal Mahadik*** for assisting us to partially complete our project on ***"Campus Collab"***.

Her expertise and talent in front-end and back-end web development, along with her ability to troubleshoot complex issues and optimize performance, played a crucial role in helping us partially complete this project.

We would also like to thank our Head of Department ***Mrs. Aboli Mohariل*** madam for providing us facilities and labs, which helped us constantly in increasing our technical knowledge, and for writing this report.

We are also thankful to our Principal ***Dr. Deven Shah*** sir for his continuous encouragement throughout the process.

Now, last but not the least, special thanks to all the staff of **Electronics & Telecommunication Engineering Department** for their technical support and constant motivation, without which this work would not have become successful.

TABLE OF CONTENTS

List of Figures

List of Tables

ABBREVIATIONS

API : Application Programming Interface

UI : User Interface

DB : Database

TCP/IP : Transmission Control Protocol/Internet Protocol

IoT : Internet of Things

FTP : File Transfer Protocol

URL: User Referal Link

NPM: Node Package Manager

ABSTRACT

This project proposes the development of a web-based platform designed for college students to upload, share, and view both academic and personal projects. The platform will feature an intuitive, user-friendly interface that facilitates effective project management and public accessibility. Additionally, it will be optimized for responsiveness across various devices to enhance user experience. By fostering an academic community and promoting resource sharing among students, this platform aims to support collaborative learning and enhance the overall educational experience.

1. INTRODUCTION

We propose a web-based platform designed for college students to upload, share, and view academic and personal projects. This platform will facilitate collaboration and resource sharing, enhancing creativity and learning experiences.

Key features will include project management tools, options for public and private sharing, user profiles, and interactive elements like comments and ratings. With a responsive and user-friendly interface, the platform aims to create a vibrant community where students can inspire and learn from each other, enriching their educational journey.

1.1. Motivation and Background

The main motivation behind Campus Collab is to streamline communication and collaboration on a university campus. Currently, students and faculty rely on a variety of tools like emails, apps, and cloud platforms to share files, work on group projects, and keep track of campus events. This can become overwhelming and inefficient. Campus Collab aims to solve this by bringing all these activities into one platform, making it easier and more efficient for everyone involved. Technologically, we are able to build this platform thanks to modern advancements in web development. Mobile integration makes it possible for users to access the platform seamlessly from their phones or laptops, providing convenience and flexibility whether they are on campus or offsite.

1.2. Objective and Scope of the Software Schema

With Campus Collab, students will have a dedicated space to upload and manage their academic projects. They can create personal accounts where they can upload multiple projects, including key details like demo videos, project codes, and software or hardware requirements. This helps students stay organized, but it also provides a centralized platform where faculty can easily access and review these projects. Professors can view all student projects, provide feedback, and track progress in one place, without the need for multiple systems.

1.3. Outline of Dissertion

The structure of Campus Collab is carefully designed to ensure a robust, scalable, and user-friendly platform for facilitating collaboration among students and faculty. This section outlines the key architectural components, technologies utilized, and the implementation process.

This dissertation focuses on designing a web-based platform tailored for college students to upload, manage, and share both academic and personal projects. It addresses the lack of existing systems that combine these functionalities while promoting student interaction through features like comments and feedback.

The introduction outlines the problem, research objectives, and scope, emphasizing the need for a specialized platform. The literature review examines existing solutions, usability principles, and gaps in current systems.

In the methodology, the system design is explained, covering tools, user-centered design, and testing. The system implementation highlights key features such as project categorization, privacy settings, and responsive design.

Results and discussion focus on platform evaluation, performance, and user engagement, followed by reflections on challenges encountered. The conclusion summarizes achievements, limitations, and recommendations for future improvements, such as adding real-time collaboration tools.

2. Problem Statement

Design and develop a web-based platform for college students to upload, share, and view academic and personal projects. The platform should facilitate easy project management, public viewing, and interaction among students, while providing a responsive and user-friendly experience.

Many students work on academic and personal projects, but there is no central place to showcase their work or find potential collaborators. Platforms like LinkedIn and GitHub exist, but they are either too broad (LinkedIn) or too specific to technical fields (GitHub). Other platforms lack the focus on college-level academic projects across various disciplines.

3. Literature Survey:

3.1. **Janvhi.V Ahire , Rutuja.U Bidgar , Sayali.D Patil Sanajana S Surywansh, Prof. Amit.P Bhuse. Research on Online Integrated Platform For Student The Journal of Computational Science and Engineering. ISSN: 2583-9055**

INTRODUCTION:

The Online Integrated Platform for Students is designed to assist diploma-level computer science and IT students with their final-year projects by providing tools for project ideation, planning, execution, and collaboration. Key features include project repositories, mentorship, career development resources, and secure data management. It integrates a user-friendly interface with frontend technologies like React.js, and backend technologies like Python or Node.js, and MySQL or MongoDB for data storage.

The platform offers modules for project ideation, task management, mentorship, and document uploads. However, challenges include scalability, content updates, user adoption, and potential security risks. The platform’s goal is to improve students' project outcomes and prepare them for the workforce, though future iterations must address its limitations.

3.2. **Cheng Xiaojin. (2014). Research on file upload based on HTML5. 2014 11th International Conference on Service Systems and Service Management (ICSSSM). doi:10.1109/icsssm.2014.687410210.1109/ICSSSM.2014.6874102**

INTRODUCTION:

The paper explores modern file upload mechanisms in web applications using HTML5 and Ajax, contrasting them with traditional FTP and HTTP-based methods. It highlights the advantages of the HTML5 File API, including components like FileList, Blob, File, and FileReader, and demonstrates how Ajax enables asynchronous file uploads, improving user experience.

Key components include the FileReader interface for reading file contents and XMLHttpRequest for sending data to the server in chunks. This approach enhances performance, particularly for large files. Limitations such as the lack of folder uploads, the need for buffer size optimization, and browser compatibility issues are noted.

The research offers a practical solution for efficient file uploads in modern web applications, emphasizing its ease of adoption and improvements in user experience and performance. The method allows for simultaneous file uploads, but further refinement is suggested for optimal results.

3.3. **Sagar, Y. (2015). Web indexing using HTML priority system. 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). doi:10.1109/ablaze.2015.7154929**

INTRODUCTION:

The paper explores web indexing techniques, introducing the HTML Priority System to improve page ranking and search efficiency. It discusses the use of inverted indexes for keyword-based searches and Distributed Hash Tables (DHTs)

for scalable indexing. The HTML Priority System ranks pages by analyzing HTML components like titles, headings, and URLs.

Key findings include the effectiveness of DHTs for distributed indexing and the system's vulnerability to content spam. The methodology involves web crawling, using Bloom Filters to avoid redundancy, and organizing data through inverted indexing. The study highlights challenges in spam detection and the need for scalability improvements in DHTs.

Overall, the HTML Priority System enhances search precision but faces limitations in spam management and optimization for large-scale data.

3.4. **Palanisamy, S., & SuvithaVani, P. (2020). A survey on RDBMS and NoSQL Databases MySQL vs MongoDB. 2020 International Conference on Computer Communication and Informatics (ICCCI). doi:10.1109/iccci48352.2020.9104.**

INTRODUCTION:

The paper compares Relational Database Management Systems (RDBMS), specifically MySQL, with NoSQL databases, focusing on MongoDB. It examines key properties like MySQL's ACID compliance and MongoDB's adherence to the CAP theorem, discussing the types of databases and their query processing methods.

Key highlights include MySQL's strength in handling structured data with fixed schemas, while MongoDB excels in unstructured data, scalability, and cloud-based applications. MySQL relies on vertical scaling, whereas MongoDB uses horizontal scaling for large-scale data. The study also explores the limitations of both systems, such as MySQL's difficulty with horizontal scaling and MongoDB's lack of standardization and stored procedures.

The study's significance lies in helping organizations choose the right database system based on their data structure and scalability needs. It concludes that MySQL is ideal for structured data, while MongoDB is suited for large, dynamic, and unstructured data in distributed environments.

4. Hardware and Software Requirement

Hardware Requirements:

- Desktop Computer / Laptop with InternetConnection
- 2 GB RAM
- 32/64 Bit CPU

Software Specifications:

Software	Version	License Type	Purpose
HTML	HTML5	Open Source	Framework
CSS	CSS3	Open Source	Styling & Designing
JavaScript	ECMAScript 2024	Open Source	Logic
React.js	17.0	Open Source	Front-End development for user interface
Node.js	16.0	Open Source	Back-end for collaboration platform
Express.js	5.0	Open Source	simplifies Node.js web development.
MongoDB	MongoDB8.0	Open Source	DBMS

Here's a brief description of each:

HTML5:

HTML5 is the latest version of the markup language used to create web pages. It introduces new elements such as `<header>`, `<footer>`, and `<video>`, enhancing the structure and multimedia capabilities of websites. HTML5 also supports modern APIs like Geolocation and Web Storage. It improves browser compatibility and provides better semantics. It is widely adopted for building dynamic and interactive web applications.

CSS3:

CSS3 is the latest evolution of Cascading Style Sheets, the language used to style HTML documents. It introduces new features like transitions, animations, and media queries, which allow for responsive design. CSS3 also includes flexible box layouts (Flexbox) and grid systems, providing better control over web layouts. It enhances the visual presentation of web content without relying heavily on JavaScript. Its modular nature allows developers to implement specific features independently.

JavaScript (ECMAScript):

JavaScript, often standardized as ECMAScript, is a high-level, dynamic programming language primarily used for adding interactive behavior to web pages. It enables client side scripting, allowing developers to create dynamic content, form validations, and multimedia control. Modern JavaScript supports advanced features like asynchronous programming (promises, async/await), modules, and object-oriented programming. It’s an essential part of the web development stack alongside HTML and CSS. The latest ECMAScript versions have brought significant improvements in performance and syntax.

React.js (ver.17.0):

React.js is a popular JavaScript library developed by Facebook for building user interfaces, particularly single-page applications. React 17.0 introduced several internal changes for better compatibility with future versions without breaking existing apps. It uses a component-based architecture and a virtual DOM for efficient rendering. React enables developers to build reusable UI components and manage state effectively with hooks. It's widely used for building fast, scalable, and maintainable web applications.

Node.js (ver.16.0):

Node.js is a server-side runtime environment that allows developers to run JavaScript code outside the browser. Node.js 16.0 includes support for the latest ECMAScript features, native Apple Silicon binaries, and better performance optimizations. It's built on Google Chrome's V8 engine, making it fast and scalable for backend services. Node.js is widely used for creating APIs, real-time applications, and handling

concurrent network connections. It offers a large ecosystem of libraries and tools via npm (Node Package Manager).

Express.js (ver.5.0):

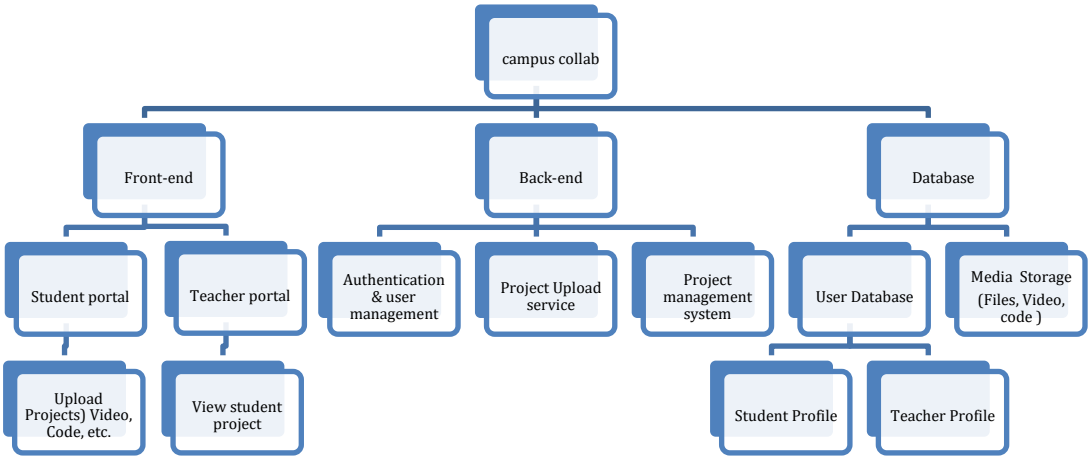
Express.js is a minimalist web framework for Node.js, designed to simplify the creation of web servers and APIs. Express 5.0 introduces improved routing, middleware management, and async error handling, allowing for cleaner and more flexible code. It offers a robust set of features for web and mobile applications, including handling HTTP requests and managing sessions. Express is highly customizable and integrates well with other tools, making it a popular choice for building RESTful APIs and web applications.

MongoDB (ver.8.0):

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, making it ideal for handling large volumes of unstructured or semi-structured data. MongoDB 8.0 introduces new features for performance improvements, cloud-native features, and better query optimization. It supports horizontal scaling through sharding and provides high availability with replica sets. MongoDB is widely used in modern web applications due to its flexibility, schema-less nature, and ability to handle complex queries. It's highly suitable for distributed systems and big data applications.

5. Block Diagram with Description

Block Diagram: Overall Layout of Project



(Fig 5.1)

Description :

- Users (Students & Teachers):
    - Students interact with the system through the Student Portal, where they upload their projects, including demo videos, code, and requirements. They can also manage their profiles and view previously submitted projects.
    - Teachers access the system through the Teacher Portal, where they can view all student projects, provide feedback, and assess the submissions.

- Front-End Interface:
    - The Student Portal allows students to log in, upload their projects, and organize their submissions.
    - The Teacher Portal provides teachers with a centralized interface to browse, search, and access the uploaded projects by students.

- Backend Services:
    - Authentication & User Management : Handles the login and registration of students and teachers, along with role-based access (i.e., students can upload, teachers can view).
    - Project Upload Service : Manages the uploading process, ensuring that project files (videos, code, etc.) are correctly stored.
    - Project Management System : This service stores and organizes all project-related data, such as project titles, descriptions, and timestamps for easy retrieval by teachers.
- User Database & Media Storage:
    - User Database : Stores user credentials and roles for both students and teachers.
    - Project Database : Stores project metadata such as titles, descriptions, submission dates, and tags for easy indexing and retrieval.
    - Media Storage : Stores the actual project content, such as demo videos, project code, and documentation.

Key Interactions:

- A student logs in to the Student Portal, creates a project, and uploads the necessary files (code, video, etc.). This data is processed by the Project Upload Service, which stores metadata in the Project Database and large files in Media Storage.
- A teacher logs into the Teacher Portal, views the list of uploaded projects, selects a project to review, and can view or download files.

This block diagram captures the high-level flow of data and interactions within the Campus Collab system. It highlights the essential components needed to achieve the goal of allowing students to upload projects and teachers to centrally access and review them.

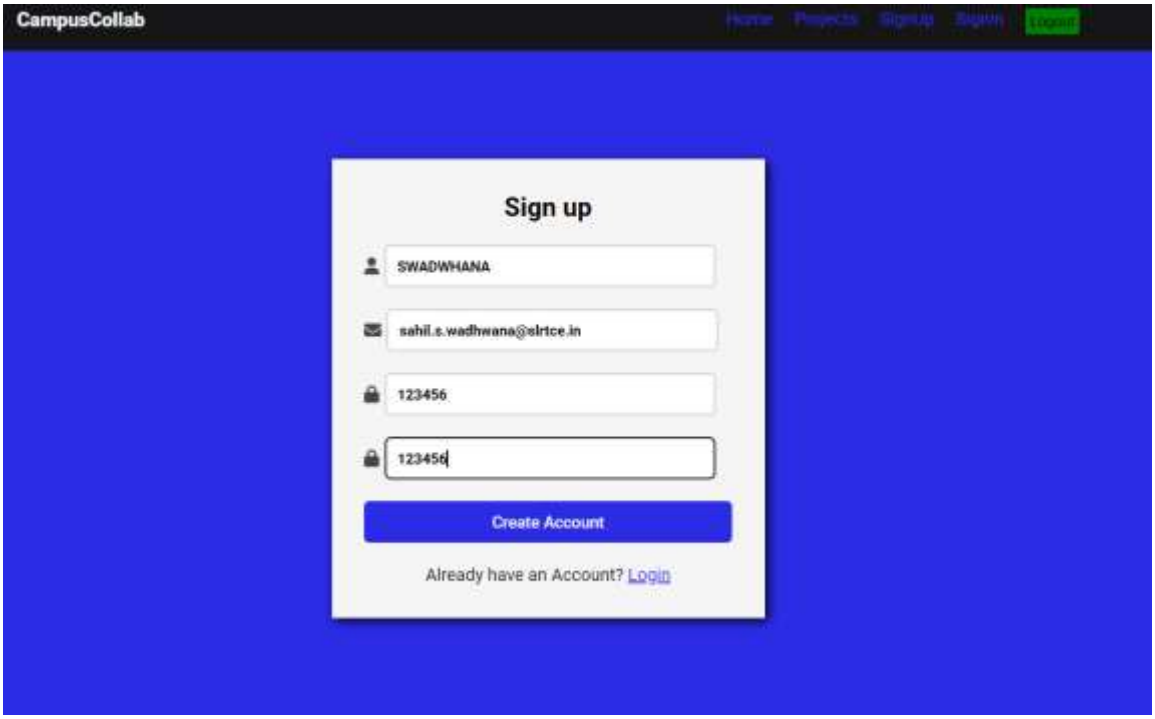
6. Procedure And Working :

Login Process Workflow: Front-END

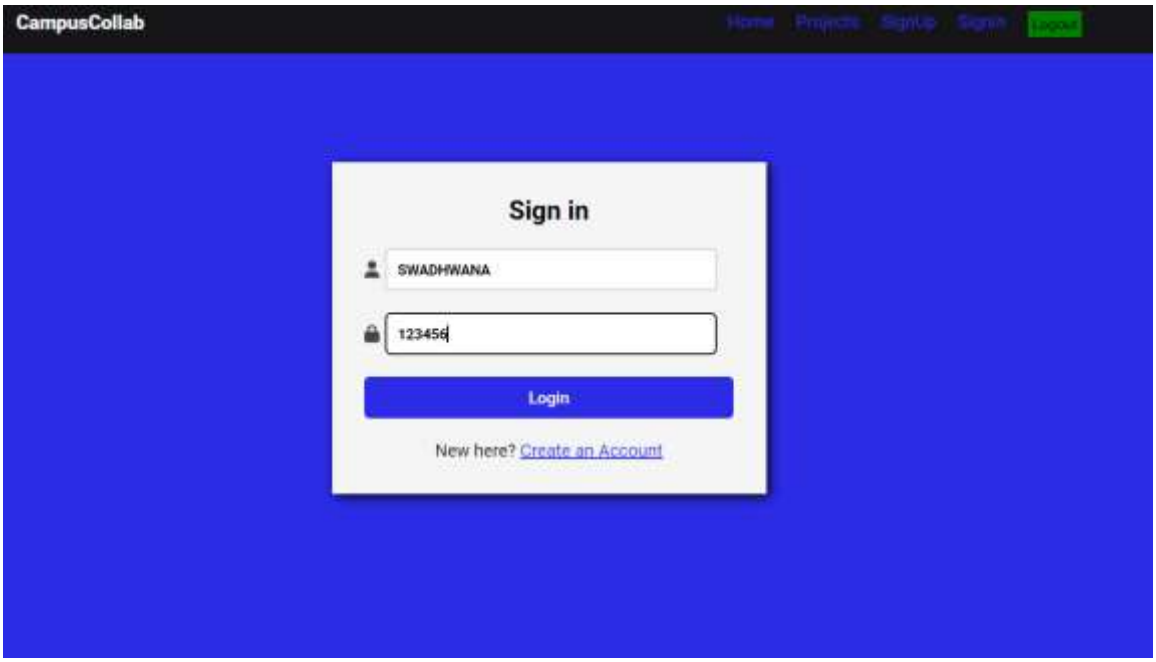
(Fig 6.1)

User Authentication (Account Creation for Students & Teachers):

- Students and Teachers create their accounts using the slrtce.in email domain.
- Students register via the Student Portal using their official slrtce.in email address (e.g., student@slrtce.in). They fill in details like their name, email, and student ID during the registration process.
- Teachers register via the Teacher Portal using their slrtce.in faculty email (e.g., teacher@slrtce.in) and faculty credentials.
- Once registered, both students and teachers receive a verification email to their slrtce.in account to confirm their registration before they can log in.
- The system verifies email domain restrictions (ensuring only users with slrtce.in addresses can register) and uses a secure authentication method to protect the accounts.



User Registration (Fig 6.2)

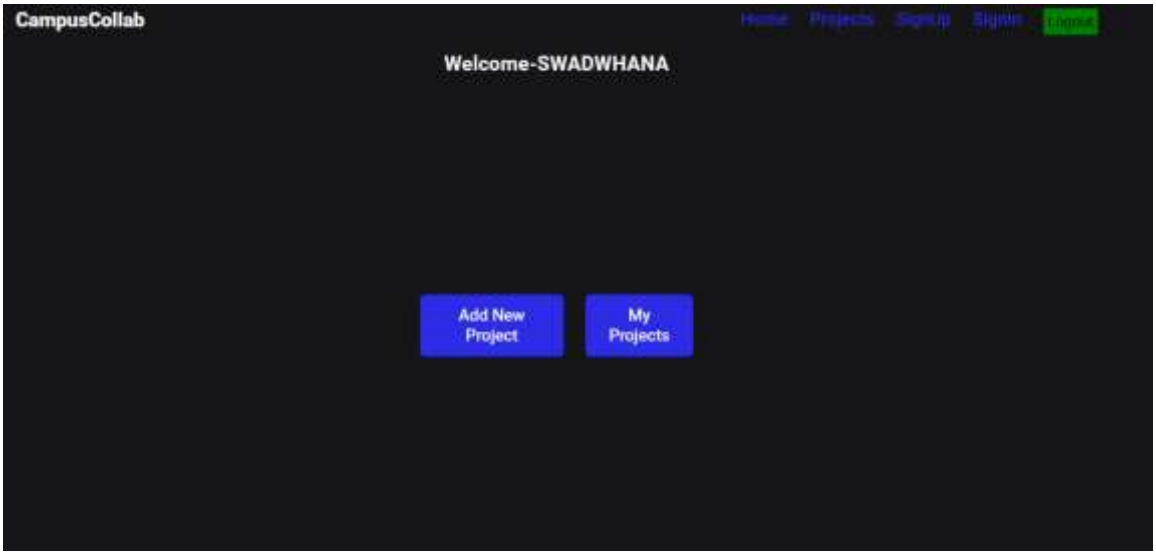


User login (Fig 6.3)

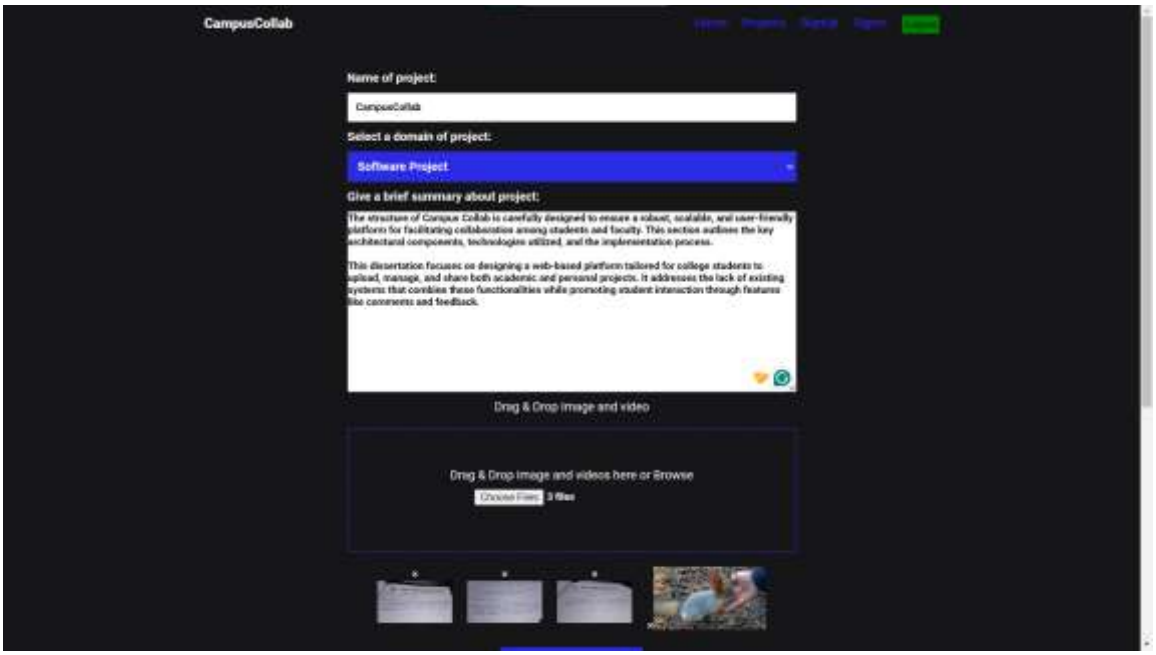
Create Project (For Students):

- After logging in through the Student Portal (using their slrtce.in credentials), students can create a new project by providing:
- Project Title
- Project Description

- Students can also manage or update previously submitted projects from this section, ensuring the project stays up to date.



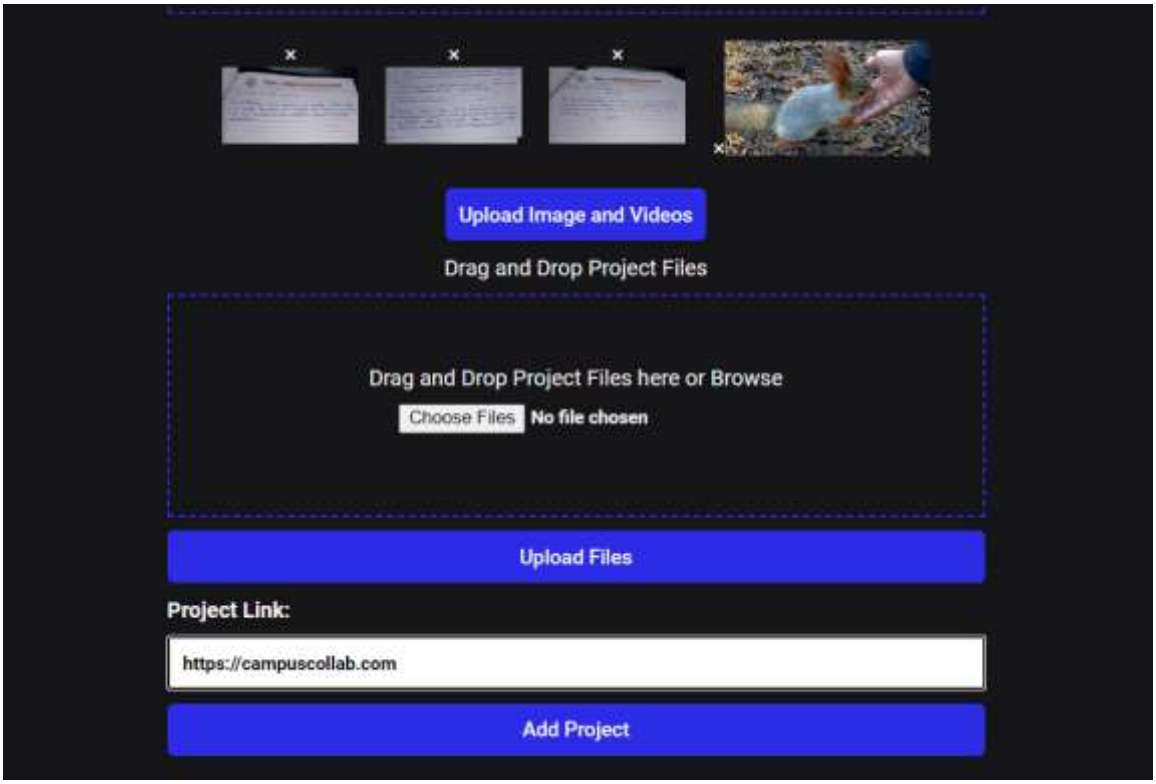
Home page (Fig 6.4)



(Fig 6.5)

Project Media Upload (Images, Videos, Code, etc.):

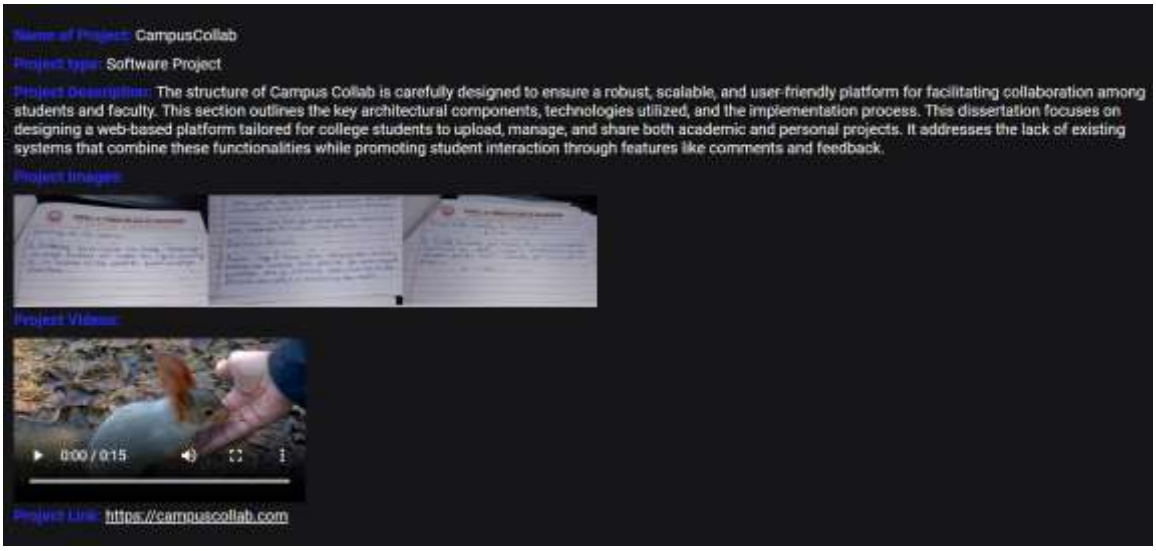
- After logging in through the Student Portal (using their slrtce.in credentials), students can create a new project by providing:
- Images: Screenshots, diagrams, or design prototypes (e.g., `.png`, `.jpeg`).
- Demo Videos: Demonstrations of the project in action (e.g., `.mp4`, `.avi`).
- Code: Source code files (e.g., `.zip`, `.tar.gz`) or links to the GitHub repository.
- Documents: Project requirements, user manuals, or technical reports in `.pdf` format.
- Students can also manage or update previously submitted projects from this section, ensuring the project stays up to date.



(Fig 6.6)

FINAL RESULT:

- Students can view a summary of their submitted projects in the Student Portal. The page includes:
- Project Name: The title of the project (e.g., "AI-Powered Traffic Control System").
- Project Type: Type of the project (e.g., Software Development, AI, Web Development).
- Project Description: A brief overview of the project.
- Project Images & Videos: Thumbnail previews of the images and video demos. Clicking on them opens the media in full view.
- Project Link: A clickable link to access the project code or demo.



Final Output (Fig 6.7)

REFERENCES

[1] Cheng Xiaojin. (2014). Research on file upload based on HTML5. 2014 11th International Conference on Service Systems and Service Management (ICSSSM). doi:10.1109/icsssm.2014.6874102 10.1109/ICSSSM.2014.6874102

[2] Janvhi.V Ahire , Rutuja.U Bidgar , Sayali.D Patil Sanajana S Surywansh, Prof. Amit.P Bhuse. Research on Online Integrated Platform For Student The Journal of Computational Science and Engineering. ISSN: 2583-9055

[3] Sagar, Y. (2015). Web indexing using HTML priority system. 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). doi:10.1109/ablaze.2015.7154929

[4] Palanisamy, S., & SuvithaVani, P. (2020). A survey on RDBMS and NoSQL Databases MySQL vs MongoDB. 2020 International Conference on Computer Communication and Informatics (ICCCI). doi:10.1109/iccci48352.2020.9104047