

5th_SEM_Report[1].pdf



My Files



My Files



University Of Engineering And Management, Jaipur

Document Details

Submission ID

trn:oid:::9639:120119071

Submission Date

Nov 6, 2025, 9:17 PM GMT+5:30

Download Date

Nov 6, 2025, 9:23 PM GMT+5:30

File Name

5th_SEM_Report[1].pdf

File Size

654.2 KB

21 Pages

2,569 Words

16,553 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



ProjecxHub: A Project Showcase App



UNIVERSITY OF ENGINEERING & MANAGEMENT,
JAIPUR

ProjecxHub: A Project Showcase App

Submitted in the partial fulfillment of the degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE & ENGINEERING

Under
UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

BY
ANSHU KUMAR
University Roll no: 12023002001004
ANKSHIKA GHOSH
University Roll no: 12023002026008
SUPRATIM MODAK
University Roll no: 12023002001020

UNDER THE GUIDANCE OF
PROF. DEBAJYOTI CHATTERJEE
COMPUTER SCIENCE & ENGINEERING



UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Approval Certificate

This is to certify that the project report entitled “Project title” submitted by Ankshika Ghosh (Roll:12023002026008), Anshu Kumar (Roll:12023002001004), Supatim Modak (Roll:12023002001020) in partial fulfillment of the requirements of the degree of Bachelor of Technology in Computer Science & Engineering from University of Engineering and Management, Jaipur was conducted in a systematic and procedural manner to the best of our knowledge. It is a bonafide work of the candidate and was conducted under our supervision and guidance during the academic session of 2023-2027.

Prof. Debajyoti Chatterjee

Project Guide, Associate/Assistant Professor (CSE)

UEM, JAIPUR

Prof. Dr. G Uma Devi

Asso. Dean of Engineering

Head of the Department (CSE)

UEM, JAIPUR

ACKNOWLEDGEMENT

The endless thanks go to Lord Almighty for all the blessings He has showered onto us, which have enabled us to write this last note in our research work. During the period of our research, as in the rest of our lives, we have been blessed by Almighty with some extraordinary people who have spun a web of support around us. Words can never be enough in expressing how grateful we are to those incredible people in our lives who have made this thesis possible. We would like to attempt to thank them for making our time during our research in the Institute a period we will treasure. We are deeply indebted to our research supervisor, Professor Debajyoti Chatterjee, for introducing us to such an interesting thesis topic. Each meeting with him added valuable aspects to the implementation and broadened our perspective. He has guided us with his invaluable suggestions, lightened up the way in our darkest times, and encouraged us a lot in our academic journey.

Ankshika Ghosh
Anshu Kumar
Supratim Modak

ABSTRACT

The ProjexHub: Project Showcase App is a digital platform for college students and company employees to upload, store, and display their innovative projects. Users can create profiles and share project details such as titles, descriptions, member names, photos, videos, PPTs, and report files. The main goal was to build a strong, scalable, and efficient application from a common code base. This approach cuts down development time and makes maintenance easier. The system uses Flutter for the frontend and Node.js with MongoDB for the backend, which is known as the FMN stack. We chose Flutter for its hot-loading feature and speed, providing near-native performance and high-quality UIs for both Android and iOS from a single code base. We selected Node.js with the Express.js framework to create the RESTful API because of its asynchronous and non-blocking design. MongoDB is a NoSQL database that allows for a flexible and schema-less data model. This flexibility is important for quickly testing application features. The app also includes secure authentication, cloud storage, media management, and project categorization. It allows users to explore, learn, and find inspiration from shared work, which promotes knowledge sharing and professional networking. The app increases the visibility of student and employee innovations while offering an easy interface for archiving and discovering projects.

Table of Contents

<u>ACKNOWLEDGEMENT</u>	Error! Bookmark not defined.
<u>ABSTRACT</u>	Error! Bookmark not defined.
<u>LIST OF FIGURES</u>	VII
1. <u>INTRODUCTION</u>	Error! Bookmark not defined.
1.1. <u>Problem Statement</u>	Error! Bookmark not defined.
1.2. <u>Objectives of the Project</u>	Error! Bookmark not defined.
1.3. <u>Scope of the Work</u>	2
2. <u>LITERATURE REVIEW</u>	3
2.1. <u>Existing Works</u>	3
2.2. <u>Research Gaps Identified</u>	3
3. <u>METHODOLOGY</u>	5
3.1. <u>Hardware and Software Specification</u>	5
3.2. <u>System Design/Framework (Block diagrams/flowcharts)</u>	5
4. <u>DESIGN & IMPLEMENTATION</u>	7
4.1. <u>Detailed Design Diagrams (block diagrams, flowcharts, etc.)</u>	7
5. <u>RESULTS & DISCUSSIONS</u>	9
5.1. <u>Testing/Experiments Performed</u>	9
5.2. <u>Result Analysis</u>	9
6. <u>CONCLUSION & FUTURE SCOPE</u>	12
6.1. <u>Conclusion</u>	12
6.2. <u>Limitations and Future Scope</u>	12
<u>REFERENCES</u>	14

LIST OF FIGURE

Figure 1.1: Detailed Flowchart of the Application.....	7
Figure 2.1: Home Screen.....	10
Figure 2.2: Project Submit Form.....	10
Figure 2.3: Projects Detailed Page.....	11

1. INTRODUCTION

The ProjexHub: Project Showcase App offers a single platform for people to store and display their project work online. In many academic and professional settings, project documents are spread across different devices or drives. This app solves the problem by providing an organized, searchable, and easily accessible storage space.

1.1. Problem Statement

Students and professionals often struggle to manage and share project work efficiently. Projects frequently exist in either distributed or decentralized locations so that they are stored away from others (locally on drives, or maybe in cloud storage folders and without any structure), making it very difficult to locate, compare, or formally review them. There is a lack of **centralized storage** where others can explore projects and learn from them. Additionally, a location for all projects, standardized and readily accessible for all, does not exist and remains a barrier for general transparency of features and assessment of quality. The project would work towards developing a system that centralizes project data in a singular thorough approach, while promoting simplifying the processes of project data storage and display.

1.2. Objectives of the Project

The Project Showcase system aims to achieve:

- Design an easy to use mobile app utilizing Flutter.
 - Implement backend APIs providing functionality to the system, using Node.js and MongoDB.
 - Create a safe and trustworthy database to store project files and metadata.
 - Allow users to upload projects which have attachments that include media and documents.
 - Create an indexed project repository that users can search, and the items be sharable.
 - Create a fair and trustworthy rating system, where projects can be rated by registered users using a quantifiable mechanism.
 - Increase project visibility to a larger user group to increase awareness and feedback.

1.3. Work Scope

The application may be used by different universities, organizations, or start-ups to organize all portfolio projects. This task covers the entire development life cycle, including:

- **Front-end Development:** Developing the user interface (UI) for viewing, searching, and rating projects.
- **Back-end Development:** Developing the application logic for handling user authentication, project submission, data validation, and rating calculation.
- **Database Management:** Development of a database schema that efficiently stores information on projects, users, and ratings.
- **Security:** Security of files and user data.

2. LITERATURE REVIEW

2.1. Existing Works

The currently available sites to share and collaborate on projects may be focused on overlapping but different audiences, but none was curtail within the context of academia or organizational project showcase. For example, GitHub is primarily focused on distributed version control and collaborative software development workflows each with the centralized focus on managing code with little emphasis on structuring documentation in multiple formats and show a project [1]. In contrast, Behance was an area for creative professionals showcase portfolios and virtually present a project and the process by which the project was developed; while engineering a project file, there is no operational means for technical project file sharing, versioning, and organizing academic metadata [2]. Separately, even though users may showcase their achievements and share projects outcomes on sites similar to LinkedIn; LinkedIn is not capable of keeping the initial project [3].

While students and project groups may sometimes use cloud drives like Google Drive to store project documents, these tools are not designed for structured project showcasing, public visibility, or validated user feedback mechanisms [4]. Flutter has emerged as a high-performance cross-platform framework capable of delivering unified mobile UIs, making it a suitable framework for building an application interface for a project showcase system [5]. And similarly, Node.js and Express.js enable developers to build scalable REST APIs that are effective at enabling data communication for web and mobile based systems [6]. Nonetheless, a means of integrating such technologies to provide secure showcasing, storing, and evaluation of academic or internal organizational projects is what is missing.

2.2. Research Gaps

Studies have repeatedly demonstrated that using collaborative tools enhances productivity and knowledge sharing in academic and workplace settings, especially when projects can be documented, accessed, and organized in an online space [7]. Most tools to date, however, do not present the ability to coexist in one working environment with organized project repositories, stored documents securely, and showcase the portfolio of projects with public access.

Another major issue is the existence of a system that allows for secure and authenticated user-based rating, using administrative controls that ensure user data is protected and not providing unsubstantiated or biased feedback [8]. While cloud storage systems do provide secure file transfer and management services, they do not provide structured metadata or contributor history, nor workflow control in the assessment of projects for academic purposes [9].

Thus, again, there is a need for a dedicated project showcase platform which is able to:

- Store any documentation, images, code, and all multimedia securely.
- Organize projects through metadata and contributor information.
- Disseminate project or semi-public showcase possibilities based on approvals.
- Contribute to authenticated rating/feedback systems that provide credentialing.
- Integrate collaborative and academic assessment on a usable platform.

The system being proposed in this project is designed to directly address these gaps and create a unified, secure showcase project ecosystem that aligns academic standards.

3. METHODOLOGY

3.1. Technologies

- Hardware Specifications:
 - Mobile Device: Android version 11 or later; Apple iOS 12 or later:
 - Internet Access: Required to use the API and gain access to the database.
- Software Specifications:
 - Flutter SDK - cross-platform mobile development tooltip
 - Dart - frontend logic and user interface
 - Node.js - server-side scripting
 - Express.js - generates a RESTful API
 - MongoDB Atlas - cloud-based NoSQL database
 - Multer - for uploading files (PDF, PPT, images, video)
 - JWT - for user authentication and session handling
 - Postman - API testing and debugging
 - Git & GitHub - version control
 - IDE: VSCode / Android Studio

3.2. System Design

- System Overview:
 - This is a client–server application structure with a Flutter front-end and a back-end with a Node.js + Express checkbox with a MongoDB database.
 - Users have the ability to register, log in, upload, and view projects (PDFs, PPT, videos, and images).
 - All information about the user and their projects is stored securely, and that data is later retrieved as a response from the RESTful APIs.
- Flow of the System:
 - The user opens the Flutter app and logs in safely.
 - The app will send authentication and request upload information to the Node.js

backend.

- Node.js will check the JWT tokens and receive the request.
- Multer will ultimately receive the uploaded files and store the uploaded files in their respective folders.
- MongoDB will log the URLs, user information, and project information.

4. DESIGN AND IMPLEMENTATION

4.1. Detailed design diagrams (block diagrams, flowcharts, etc.)

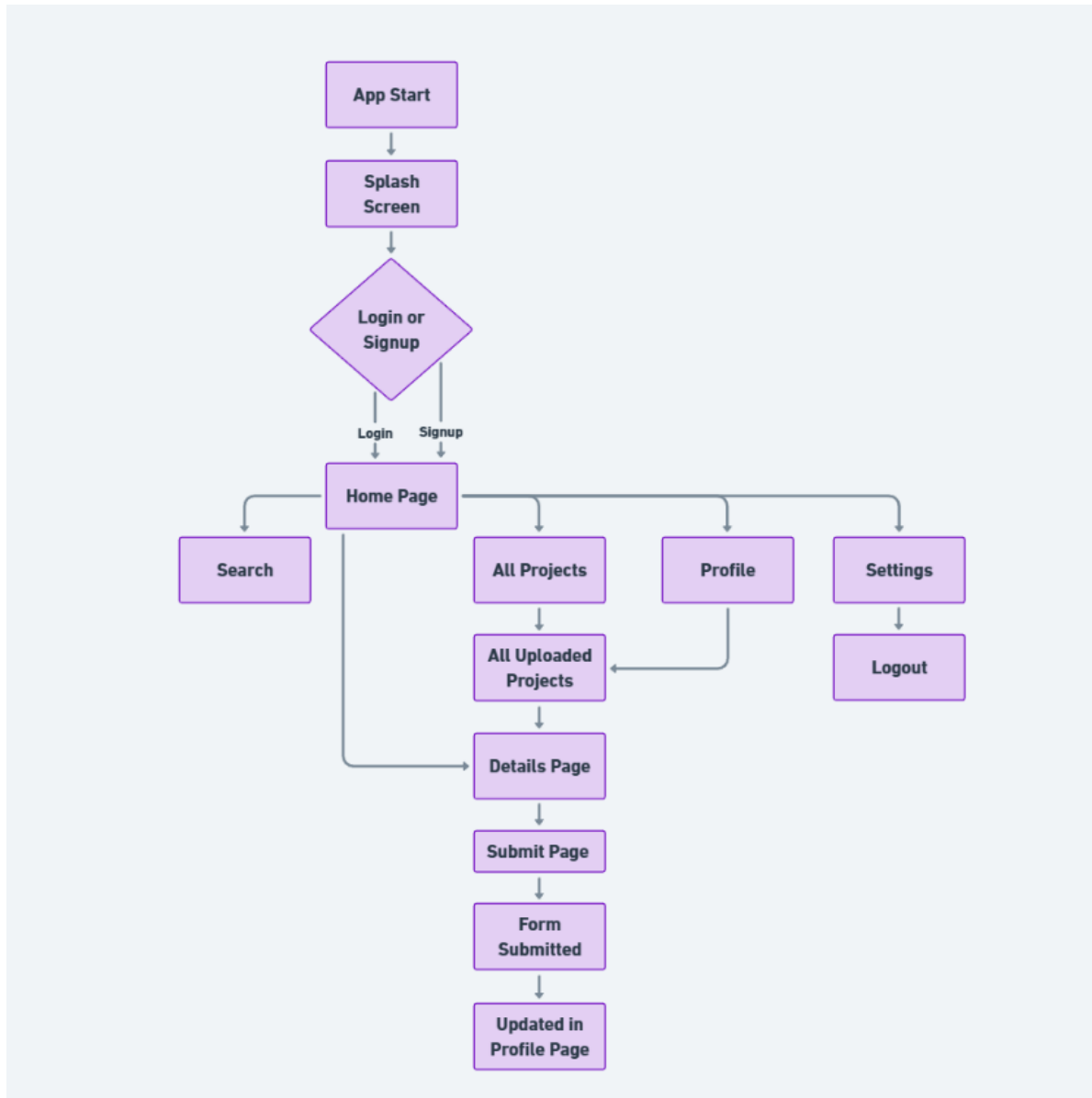


Figure 1.1 : Detailed flowchart of the application

The design consists of a simple, clean, and user-friendly interface that allows the visual presentation of projects to serve as the focal point. The detailed design consists of the basic screens and a few key features outlined below:

- Login/Signup page with Firebase Authentication.
- Project upload form that consists of fields for title, description, members, media, and files.
- Project gallery with search and filters.
- Project detail screen that will show links to download/view.

Flutter widgets were used for UI development. The speed with which the framework allowed us to access nearly native performance for Android and iOS from a single code base was crucial to the implementation. Node.js APIs empowered the backend logic and CRUD (Creating, Reading, Updating and Deleting) logic while MongoDB as a NoSQL database allowed the necessary schema-less data model flexibility required to iterate quickly in testing application features. The combination of the FMN stack provided performance and developer speed better than traditional development methodologies of native applications.

5. RESULTS & DISCUSSION

5.1. Testing and Experiments

Testing was performed on both the Android and web apps. Functional tests were conducted to confirm each of the upload, download, and authentication aspects of the applications worked as expected. Testing of the applications showed that interfacing between the Flutter front-end and the node.js/MongoDB back-end appears to be reliable and secure with regard to these important functions.

5.2. Result Analysis

The analysis of the results suggested the upload and retrieval of multimedia and documents worked without problems. The UI was internally rated with high usability. The system appears to have resolved all basic components of centralized storage and a working UI to explore projects and thus resolved the problem of messy project documentation.



Figure 2.1: Home Screen

Figure 2.2: Project Submit Form

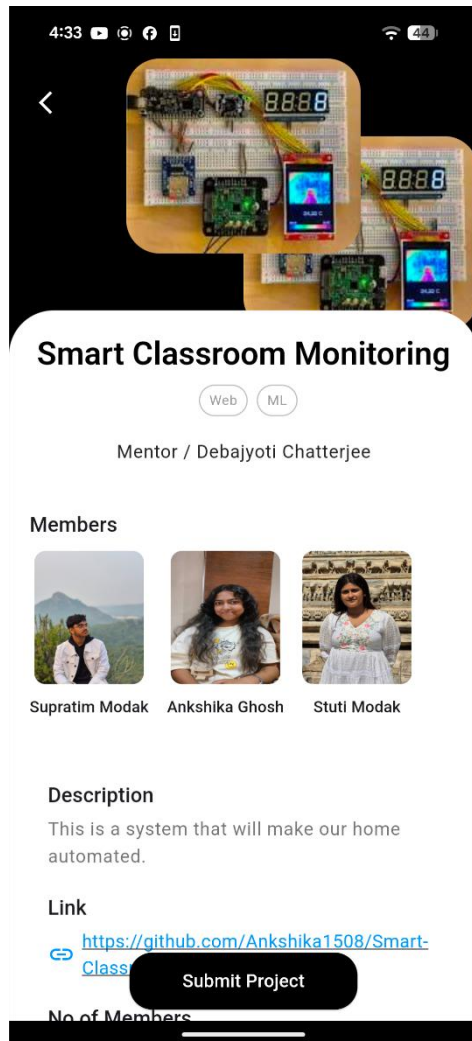


Figure 2.3: Projects Detailed Page

6. CONCLUSION & FUTURE SCOPE

6.1. Conclusion

The ProjexHub Application operates successfully as a digital tool for students, allowing all to collect in one place and upload, display and review thinking and engaging college projects.

It works to overcome the barriers put in place for communication with the students, faculty and evaluators through a single place for documents and reviews.

Using Flutter, Node.js, and MongoDB, the application can meant for performance across devices, scalability of the app and performance.

Authentication is protected using JWT to protect users so that only authorized users can upload and reuse the data.

It has the functionality for the user to also upload additional PDFs, PPTs, videos, and images in two formats, so it is a very versatile app for technical and creative projects.

In the end, the project has been successful in building a reputable user-friendly, secure, and manageable academic project zone for sharing and review.

6.2. Future Scope

- Admin Panel Integration: Build an admin dashboard functionality where your can review, edit or delete images or information have been flagged as inappropriate.
- Project Rating & Feedback System: Build in functionality for users and mentors to rate and provide comments on the project to help improve peer learning.
- Search and Filters: Build in filters for search for example by department/year or the technology being used.
- Cloud File Storage: The media will be stored on AWS S3 or other similar services for scalability.
- AI Driven Recommendation Engine: The application will use algorithms to recommend other related, or newsworthy, projects.
- Web Application: The application will be changed to a responsive web application so it can be accessed through a browser.
- Collaboration: Project teams will be able to communicate, update the project, and share

resources within the application.

- College Integration: The platform will be integrated with college portals or LMS systems to allow projects to be submitted and graded automatically.

REFERENCES

- [1] Chacon, S., & Straub, B. (2014). *Pro Git*. Apress.
- [2] Behance Creative Portfolio Platform. Adobe Inc. <https://www.behance.net>
- [3] LinkedIn Professional Networking Platform. <https://www.linkedin.com>
- [4] Google Drive Documentation. <https://support.google.com/drive>
- [5] Singh, A., & Sharma, R. (2020). Cross-platform development using Flutter. *IJARCS*, 11(3), 45–50.
- [6] Node.js Official Documentation. <https://nodejs.org/en/docs/>
- [7] Alam, S., & Miah, M. S. (2021). Collaborative project management platforms. *IJCA*, 174(12), 25–32.
- [8] Rieh, S. Y., & Danielson, D. R. (2007). Credibility framework. *ARIST*, 41(1), 307–364.
- [9] Williams, T., & Galloway, D. (2020). Secure cloud-based data storage. *Journal of Information Security*, 9(2), 121–133.