

NEXUS HUB

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE
AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE



Submitted by

JHALAK SAXENA (2100290120090)

DIVYANSH SINGH (2100290120074)

Supervised by

DR. KALPNA SAGAR

Associate Professor, Department of Computer Science

Session 2024-25

DEPARTMENT OF COMPUTER SCIENCE

KIET GROUP OF INSTITUTIONS, GHAZIABAD

(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)

May 2025

DECLARATION

I/We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Jhalak Saxena

Roll No.: 2100290120090

Date:

Signature

Name: Divyansh Singh

Roll No.: 2100290120074

Date:

CERTIFICATE

This is to certify that Project Report entitled “Nexus Hub” which is submitted by Jhalak Saxena and Divyansh Singh in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Date:

Supervisor

Dr. Kalpna Sagar

Associate Professor

Department of Computer Science

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to Associate Professor, Dr. Kalpna Sagar, Department of Computer Science, KIET, Ghaziabad, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen the light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kr. Shrivastava, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date:

Signature:

Name: Jhalak Saxena

Roll No.: 2100290120090

Signature:

Name: Divyansh Singh

Roll No.: 2100290120074

ABSTRACT

The traditional methods of managing university-level student projects are usually subjected to numerous challenges likely to affect overall student-faculty collaboration and communication success. Such include uncoordinated communication, wide gaps in exchange of ideas, and the lack of one integrated project management tool. Nexus Hub was a pioneering web application developed to overcome the complexity of managing students' projects due to the above-specified challenges. Besides filling above gaps in communication, Nexus Hub is an active collaboration and visibility space, where the students upload their work in one shared space. Nexus Hub offers a full set of tools needed, which enables users to build rich project profiles, connect with potential team members, build cohesive teams, assign tasks effectively, and integrate with GitHub for candidate assessment. Built using ReactJS, Node.js, and MongoDB, the platform enables the enhancement of coordination through an interactive space where students and lecturers collaborate, thus resulting in successful project outcomes. Early results of the use of Nexus Hub are enhanced efficiency in teamwork, enhanced levels of completion of projects, and enhanced recognition of student success by academia and industry. Specifically, Nexus Hub is a huge leap towards filling gaps in communication and collaboration while empowering students to own their projects and success. Innovative features and easy-to-use designs will revolutionize the academic process of managing projects into something more efficient, collaborative, and rewarding for all stakeholders.

TABLE OF CONTENTS

Page No.

DECLARATION.....	2
CERTIFICATE.....	3
ACKNOWLEDGEMENTS.....	4
ABSTRACT.....	5
LIST OF FIGURES.....	8
LIST OF ABBREVIATIONS.....	9
SDG MAPPING WITH JUSTIFICATION	10
CHAPTER 1 INTRODUCTION	
1.1 Introduction to Project	
1.2 Project Category	
1.3 Objectives	
1.4 Structure of Report	
CHAPTER 2 LITERATURE REVIEW	
2.1 Literature Review	
2.2 Research Gaps	
2.3 Problem Formulation	
CHAPTER 3 PROPOSED SYSTEM	
3.1 Proposed System	
3.2 Unique Features of the System	

CHAPTER 4 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study

4.2 Software Requirement Specification

4.2.1 Data Requirement

4.2.2 Functional Requirement

4.2.3 Performance Requirement

4.2.4 Maintainability Requirement

4.2.5 Security Requirement

4.3 SDLC Model Used

4.4 System Design

4.4.1 Data Flow Diagrams

4.4.2 Use Case Diagrams

4.5 Database Design

CHAPTER 5 IMPLEMENTATION

5.1 Tools and Technologies Used

CHAPTER 6 TESTING, AND MAINTENANCE

6.1 Testing Techniques and Test Cases Used

CHAPTER 7 RESULTS AND DISCUSSIONS

7.1 Description of Modules with Snapshots

7.2 Key Findings

7.3 Brief Description of Database with Snapshots

CHAPTER 8 CONCLUSION AND FUTURE SCOPE

8.1 Future Scope

8.2 Final Thoughts

REFERENCES

Turnitin Plagiarism Report

Proof of patent publication

LIST OF FIGURES

Figure No.	Description	Page No.
4.1	Iterative and Incremental Software Development Life Cycle	25
4.2	Data Flow Diagram	27
4.3	Use Case Diagram	30
7.1	Snapshot of Database (Users Table)	55
7.2	Snapshot of Database (Users Table)	56
7.3	Snapshot of Database (Users Table)	58
7.4	Snapshot of Database (Users Table)	59
7.5	Snapshot of Database (Users Table)	60

LIST OF TABLES

Table No.	Description	Page No.
4.1	Principal Use Cases mapping onto the Use Case Diagram	40
4.2	Component-wise Design	42
4.3	users Collection Schema	43
4.4	projects Collection Schema	44
4.5	applications Collection Schema	45
6.1	Executed Test Cases and Results	50

LIST OF ABBREVIATIONS

SDG	Sustainable Development Goals
API	Application Programming Interface
JWT	JSON Web Token
SaaS	Software as a Service
UI	User Interface
UX	User Experience
SPA	Single Page Application
VPS	Virtual Private Server
ACM	Association for Computing Machinery
IEEE TLT	IEEE Transactions on Learning Technologies
HCI	Human-Computer Interaction
NACE	National Association of Colleges and Employers
PMI	Project Management Institute
DFD	Data Flow Diagram

UML	Unified Modelling Language
SRS	Software Requirement Specification
ER	Entity-Relationship
SDLC	Software Development Lifecycle
CI/CD	Continuous Integration / Continuous Deployment
QA	Quality Assurance

SDG MAPPING WITH JUSTIFICATION

SDG 4: Quality Education

Ensures quality and fair education for all and provides opportunities for lifelong learning.

Justification

The project assists in supporting SDG 4 through enabling a shared digital platform that improves student participation in learning activities. Through enabling access to shared resources, version-controlled submissions, and communication, students get the opportunity to achieve equitable and inclusive learning beyond the classroom. The system enables access to quality learning infrastructure across geographical location or institution.

Through the incorporation of aspects of peer to peer cooperation, instant feedback, and natively supported multiple types of projects, the platform facilitates enhanced understanding of course material. The students are encouraged to work as teams, share information, and acquire skills that are at the cutting edge of long-term professional and academic growth.

The system also enables lifetime learning by tracking previous work, enabling continuous development, and enabling users to follow up and proceed from where they left off. It holds the learner accountable for learning personally, hence ensuring lifetime learning development in life.

SDG 9: Industry, Innovation, and Infrastructure

Build strong infrastructure, facilitate sustainable and inclusive industrialisation, and improve innovation

Justification

Nexus Hub tackles SDG 9 head-on through an online platform that reimagines scholarly collaboration and project management in the education institution. The platform facilitates systematic, scalable processes that substitute for time-consuming manual coordination with strong, technology-supported solutions that mirror actual industry processes.

Through task management software, integrated communication platforms, and automated reports, Nexus Hub allows students to schedule, implement, and monitor their studies with accuracy. This maximizes innovation by allowing them to dedicate more time to creative problem-solving and less to administrative delays. It facilitates the adoption of industry-standard software like repositories and issue tracking systems in the academic world.

The modularity of the platform and the ability to be integrated with other systems are the reasons why it is a future-proofed solution that is sustainable. It provides an education infrastructure that not only has the ability to cater to existing demands but has the ability to adapt in proportion to rising technological innovations.

SDG 10: Reduced Inequalities

End discrimination between and within states.

Justification

Nexus Hub achieves SDG 10 by offering equal access to the same high-quality collaborative learning platform to every student irrespective of socio-economic background, institutional resources, or geography. It is device-agnostic and internet speed-agnostic so that it performs equally for individuals with different levels of access to technology.

By offering an equitable learning environment, Nexus Hub erases disparities in the form of different institutional facilities or support systems. It allows marginalized or

under-represented students to be actively engaged in peer-to-peer collaborative learning, knowledge sharing, and contribution to academic work.

Apart from that, by consolidating scholarly output in one platform, Nexus Hub allows for assessment, contribution tracking, and participation measurement to be neutral and transparent. This eliminates prejudice and allows for merit-based recognition to provide equal opportunity to all students irrespective of background or privilege.

SDG 17: Partnerships for the Goals

Improve the capacity for implementation and rekindle the global partnership for sustainable development.

Justification

Rationale Nexus Hub facilitates SDG 17 through employer-institution-student partnerships on an interactive platform. It brings the stakeholders onto one online platform and makes the stakeholders capable of idea, opportunity, and project sharing and builds a community with a focus on collaborative development and growth.

The system allows for inter-institutional cooperation by the provision to use several departments or colleges of users as a team to work on shared academic goals. This promotes cross-disciplinary learning culture and sense of accountability. The visibility elements of the platform also ease the work of academics to reach recruiters and industry players, with avenues for mentorship and communication.

By coordinating learning activities with wide industry and learning goals, Nexus Hub is a virtual bridge between learning and employment. Harmonization induces learning behavior that is sustainable and aligned with international development priorities and offers learners skills in line with the demands of the real world.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Good project management and coordination are vital to student success in the rapidly changing higher education system of the present day. Traditional academic project management methodologies, however, are typically described by fragmented communication, decentralized choice, and poor coordination. These are exacerbated when students collaborate across institutions or departments, leading to delays, incoherencies in objectives, and second-best results.

Nexus Hub offers a groundbreaking solution to such problems. It is an internet-based platform that has the capability to optimize every stage of the project life cycle, from start to finish, and optimize the interaction among students, mentors, and employers. With the feature of GitHub profile linking, role-based access control, and automation, Nexus Hub offers greater visibility, accountability, and resource optimization.

Within global distance education trends, the web site offers a solution to the growing need for scalable, accessible systems to enable students to submit assignments, engage with each other, and gain industry accreditation.

Nexus Hub revolutionizes planning, construction, and delivery of higher education academic projects. Since collaborative learning and application of skills are on the rise in the modern world, the platform responds to a pressing need for structured project management solutions that are appropriate for student teams and academic projects.

It is an end-to-end platform to enable team building, competency validation, and outcome transparency. Students can build high-quality project portfolios, create teams based on validated competencies, and share work to academic and professional communities. Unlike generic collaboration platforms, Nexus Hub combines academic objectives and

industry relevance through purpose-designed features. Its most groundbreaking feature is integration with professional development platforms. By combining GitHub and LinkedIn access with team building, Nexus Hub creates a tangible connection between academic work and professional self. Not just does it improve the standard of collaboration but also actually creates actual career progression opportunities with exposure to projects.

The site confronts inefficiencies in modern scholarly practice head on. Team formation is typically an ad hoc process that leads to uneven work loads and incompatible skill sets. Moreover, excellent student work sits in the classroom in isolation without an institutionalized process for accessing professional development.

Nexus Hub bridges these gaps with a strong yet flexible scholarly project management framework. It's based on three ideals: openness in team formation, verification against real skills, and elevation of the quality bar for students. These ideals inform an architecture that is user-focused, enabling the entire project life cycle and available to students and educators.

1.2 Project Category

Nexus Hub falls under several various categories that denote its field of operation and activity. It is first and foremost an Academic Collaboration Platform, enabling collaborative work among students and instructors from various institutions. It is also a Software as a Service (SaaS) as it is cloud-hosted and accessed through common web browsers without local installation.

Besides, Nexus Hub is a Project Management Tool that encompasses such features as task assignment, tracking, and GitHub integration to enable candidate evaluation and project workflow management.

1.3 Objectives

Nexus Hub is conceived to enable collaborative work on research initiatives and collaboration among students, faculty, and industry. It is conceived to enable simpler project management while creating a more networked and efficient research culture with next-generation web technologies. The overall objectives of Nexus Hub are as follows:

- **Centralized Collaboration:**

Providing a shared digital platform through which students can collaborate and design, share, and exhibit their learning projects on a single centralized platform that encourages collaboration while keeping all the related activities for the projects free and open under one roof.

- **GitHub Integration:**

GitHub integration that enables recruiters and mentors to gauge the technical ability of students from their version-controlled repositories. It bridges the gap between academic projects and real software development practices.

- **Role-Based Access:**

With a secure, role-based access system that tailors user experience according to role—student, faculty, or industry partner. It ensures data privacy while offering relevant tools and interfaces appropriate to each user's role.

- **Efficiency Improvement:**

Minimizing the teacher's administrative workload by providing them with organized tools to track team work and project progress. It is intended to automate duplicative work and enables more effective supervision, thus enhancing productivity throughout the academic environment.

1.4 Structure of Report

The project is structured into eight chapters covering Chapter 1: Introduction, covering overview, objectives, and project significance. Chapter 2: Literature Review covers

academic collaboration and project management platforms already in existence, existing solution gaps, and frames the problem statement. Chapter 3: Proposed System outlines the Nexus Hub platform, its core features like centralized collaboration, role-based access, and GitHub integration. Chapter 4: Tools, Technologies and Dataset Description gives out the framework utilized and dataset preparation. Chapter 6: Testing and Maintenance includes test cases and validation processes. Chapter 7: Results and Discussion outlines the performance metrics and comparative study. Lastly, Chapter 8: Conclusion and Future Scope summarizes results and addresses improvement areas, ensuring systematic and comprehensive presentation of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review

Successful academic teamwork and project management are increasingly a requirement in higher education. As there are more students and longer projects, institutions require platforms that facilitate easier team creation, task management, and candidate screening. Most traditional tools are not able to handle structured teamwork or incorporate external evaluation tools such as GitHub. Recently published studies indicate new tendencies and constraints of existing scholarly collaboration systems and offer insights into the creation of Nexus Hub.

Existing Solutions to Academic Collaboration Platforms

Literature regarding scholarly collaboration tools in existence prior to now can be classified into the following broad categories:

1. Profile-Based Team Formation

Team building continues to be an essential element of collaborative structures. A controlled experiment of 1,200 students (Chen & Zhang, 2023) revealed that formal application-based team assignments, rather than self-organized teams, resulted in a 23% improvement in task completion. Furthermore, GitHub profile analysis is now utilized by 68% of leading computer science programs as part of student assessment procedures (ACM Education Survey, 2023). Nevertheless, in spite of its value, only 14% of platforms now feature LinkedIn, which can provide predictive information regarding soft skills (EdTech Journal, 2024).

2. Application Management and Evaluation Systems

Other systems use two-step application processes with profile filtering and competency assessment. This has registered a 31% decrease in inappropriate team assignments (IEEE TLT, 2023). Time spent screening a candidate has also decreased to 4.2 minutes using standard profile screens, improving recruiter efficiency and decision speed (HCI in Education, 2023).

Limitations of Current Solutions

Despite these advancements, current platforms are limited in several key areas:

- **Lack of Clear Acceptance Criteria:** According to a study conducted by the EdTech Ethics Review (2023), 72% of systems lack clear acceptance criteria for team applications, and consequently, teams are formed in inconsistent or subjective manners.
- **Less than 9% of websites provide support for creating balanced teams with a mix of technical and non-technical positions** (Team Formation Studies, 2024).
- **Limited Cross-Platform Integration:** While GitHub is becoming standard, deeper integration with business platforms like LinkedIn or automated analytics is not typical, limiting the platform's ability to assess candidate readiness comprehensively.

The present scenario reflects encouraging trends—like profile-based apps and GitHub integration—but continues to struggle with significant gaps in process visibility, role balance, and integration scope. These limitations make it necessary for a wide platform like Nexus Hub, which provides centralized collaboration, streamlined application review processes, and role-based secure access—all with the purpose of facilitating scholarly collaboration and industry interaction.

2.2 Research Gaps

While there are many project management tools and scholarly collaboration sites available to scholars, there are still broad areas of research in the area of overall student assessment and regular project assignment protocols:

1. **Ineffective Integration of Holistic Assessment** – Existing platforms normally assess candidates against limited datasets. There is no common system that combines GitHub activity metrics (GitHub Education, 2023), LinkedIn-based soft skills metrics (NACE, 2024), and institutional academic records (AAC&U, 2023). This fragmented assessment process prevents accurate, holistic determination of candidate readiness.
2. **Lack of Process Transparency in Selection Criteria** – A vast majority of students (89%) indicate that they do not know the criteria on which their potential team membership is evaluated prior to their application (Student Voice Survey, 2024). In addition, employees confess that almost 42% of the decisions are based on subjective or unofficial criteria (PMI Education, 2023), leading to inconsistency and bias.
3. **Insufficient Post-Match Onboarding and Monitoring Functionality** – Fewer than 11% of systems offer onboarding assistance through checklists or guidelines to ensure effective team collaboration (Onboarding Research, 2023). Further, only 6% of systems monitor whether the initial ability tests match actual team

performance (ACM CSCW, 2024), which goes against the interest of long-term project success.

These gaps demonstrate the need for an integrated, transparent, and analytically grounded platform—like Nexus Hub—to facilitate not just team building and project management but also ongoing assessment, onboarding, and outcome monitoring in an effort to provide for academic and career alignment.

2.3 Problem Formulation

Statement of the Problem

Existing education cooperation platforms are failing to implement a consistent strategy to assess the competency of students, provide open selection processes, and provide systematic post-assignment support. Candidate selection is often confined to single indicators such as resumes or simple applications, without technical, academic, and professional information. In addition, non-open selection processes and minimal onboarding content also result in incompatible teams, decreased productivity, and mixed project outputs.

Proposed Solution

To fill these gaps, Nexus Hub provides a scaffolded, evidence-based platform facilitating project-based co-operation by a three-level system:

- *Structured Assessment Framework*: Compiles a weighted count of GitHub contribution (pull requests, commits, issues), LinkedIn skill endorsements, and academic coursework alignment according to institutional transcripts.
- *Transparent Processes*: Posts evaluation criteria in advance, enables anonymous review of profiles to reduce bias, and automatically creates skill gap reports to guide application decisions.

- *Post-Acceptance Support*: Offers personalized pairings of mentors according to selected roles, provides milestone documents to coordinate project schedules, and delivers ongoing feedback to maintain performance in sync with expectations.

Objectives of the Proposed System

- To consolidate student assessment based on technical, academic, and professional criteria.
- To foster equality and openness in team selection by using systematic rubrics and blind review.
- For improving onboarding and team preparedness through automated support and role-based guidance.
- To ensure sustained project performance through continuous feedback and analytics.

This study seeks to provide an institution-friendly, scalable, and industry-aligned solution that updates academic project collaboration and facilitates equity, efficiency, and visibility of skills for all stakeholders.

CHAPTER 3

PROPOSED SYSTEM

3.1 Proposed System

To close the gap for structured, transparent, and competency-mapped academic collaboration, this project suggests Nexus Hub—a web-based, AI-driven collaboration platform for project-based learning in higher education. In contrast to conventional systems based on unstructured applications and restricted assessment criteria, Nexus Hub

offers a scalable, evidence-based solution for project development, candidate assessment, and team management.

The system we suggest follows a four-step collaborative approach in a three-tier architecture:

1. Project Profile Development:

In the initial phase, students initiate new projects by adding detailed descriptions through the platform. These descriptions include core objectives, technical and non-technical skill sets required, and elective GitHub repository URLs. The project entries, having been added, are stored in a shared database to allow institutional visibility and systematic tracking. This centralization facilitates easier discovery of the projects, as well as enabling institutions to maintain a shared record of ongoing academic activities.

2. Candidate Application:

During the second phase, students who are interested can apply directly to projects on the platform. The candidates are asked to give their GitHub and LinkedIn URLs. Nexus Hub will then automatically gather appropriate contribution information, including repository history, programming languages, endorsements, and public project activity, from the given URLs. These details are compiled into a uniform digital resume, which allows for objective comparison of the candidates and convenient decision-making for project authors.

3. Evaluation & Selection:

In the process of evaluation, project creators apply a pre-defined rubric to rate candidates. The rubric considers various parameters like GitHub usage levels, LinkedIn-reported skills, and educational background matching the goals of a project. To make the process even fairer and minimize bias, candidate evaluation can be done anonymously. The system can even automatically generate skill gap reports highlighting areas where project needs and candidates' skills do not meet, thereby simplifying team building.

4. Team Management:

Once the team is established, Nexus Hub provides seamless collaboration features. Task-tracking built-in modules enable team members to allocate and track progress on individual deliverables. Visibility of code contributions is provided through version control system integration, and ongoing conversation is enabled through real-time communication tools. Milestone documentation and mentor comments are also built into the system, which provides project direction, productivity, and responsibility throughout the development process.

It is based on a robust three-tier architecture:

- 1. Presentation Layer:** An adaptive user interface designed using React.js and Material-UI to enable effortless cross-device interaction.
- 2. Application Layer:** A RESTful API built with Node.js and Express.js and authenticated by JWT-based authentication.
- 3. Data Layer:** MongoDB Atlas cloud-hosted database with Redis caching for elastic storage and rapid access.

Together, these stages form an integral research collaborative working space with the aim of ensuring transparency, enhancement of teaming, and enhancement of effective project execution in the higher education industry.

3.2 Unique Features of The System

The Nexus Hub platform presented herein provides a set of novel tools to differentiate it from other academic project software. The introduction of actual coding metrics, organized streams of assessments, and secure team collaboration interfaces provides a solid base for skill-based project development in higher education.

1. GitHub-Linked Profile Review:

In contrast to conventional application platforms that depend on static resumes and human evaluation, Nexus Hub relies on real-time integration with GitHub to auto-evaluate. Prospective employees are asked to link their public GitHub accounts, where the system automatically evaluates rich information like commit history, work in repositories, and usage of language. Integration allows the project owners to see the coding practices and contribution to projects of applicants. The technical information received is then algorithmically matched with project requirements for the shortlisted applicants to ensure that shortlisted applicants have domain expertise and real experience.

2. Systematic Evaluation Process:

Nexus Hub boasts a clear and standard assessment process that minimizes bias and ensures equitable selection among applicants. Every application is brought to a standard form that makes it simple for project leads to cross-compare profiles. The platform allows creators to define necessary technical and non-technical abilities for their teams, including programming languages, project management abilities, and communication abilities. An integrated visual interface also simplifies by matching candidates' skills against the given requirements, producing a direct skill gap analysis that feeds into objective decision-making and balanced team formation.

3. Team Coordination Tools:

Once the team members are established, the platform becomes a collaborative environment in which the users are presented with a centralized dashboard of approved participants, their respective roles, and respective contact details. A summary module is present on each page of a project that aggregates the most important information like project purposes, technologies to be employed, development timelines, and expectations about milestones. This architecture does not allow the entire team to have a different perception of the project purposes, thus eliminating confusion and improving productivity during the development process.

4. Privacy and Access Control:

Privacy and security are core issues when designing Nexus Hub. The system employs credential-based login mechanisms to safeguard access to accounts and personal data. It also employs tier-based, role-based permission schemes that divide levels of access for administrators, mentors, as well as students. This enables users to access only data and functionality specific to their role, safeguarding data integrity and a secure collaborative environment at all levels of institutions.

These distinguishing features cumulatively position Nexus Hub as an innovative talent-driven platform that has the potential to revolutionize academic collaboration. By simplifying decision-making, enabling openness, and enabling safe cooperation, it is a scalable solution for student-led project management within and across institutions of higher education.

CHAPTER 4

REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study (Technical, Economic, Operational)

Before the design and implementation of the Nexus Hub project collaboration platform, feasibility was quantified to determine its technological, economic, and operational feasibility. The objective was to make sure that the system not only aligns well with institutional objectives but is also deployable and sustainable in real academic environments.

4.1.1 Technical Feasibility

Technical feasibility validates the adoption feasibility of the suggested system using current technology, tools, and infrastructure. For Nexus Hub, the assessment validates whether the platform can be successfully implemented, deployed, and supported using current web-based technologies and can meet performance, scalability, and accessibility needs in learning settings. The following technology components were taken into account to validate the readiness and suitability of the system:

- **Frontend:**

User interface is developed using ReactJS, a popular JavaScript library to develop quick, responsive, and interactive web applications. Its modular, component-based design allows for neat and scalable code, which is simple to maintain and extend in the future.

- **Backend:**

Server-side processing is governed by using Node.js and Express.js, which provide a lightweight and effective API request processing, routing, and user authentication framework. This allows the system to process a large number of concurrent users effectively.

- **Database:**

A MongoDB Atlas hosted database for the cloud holds all the user and project data. It is used for flexibility in handling dynamic data structures. Redis caching is also utilized to enhance response time and reduce server load in repeated operations.

- **Integration:**

Nexus Hub features GitHub integration, allowing users to link repositories. This automatically tracks coding activity and repository details and brings user profiles together with technological contributions.

Based on modern web development standards and educational IT infrastructure, the system is technically viable as well as appropriate for academic deployment.

4.1.2 Economic Feasibility

Economic feasibility aims at the evaluation of the cost-benefit analysis of the platform in an educational context:

- **Open-Source Technology Stack:** Nexus Hub is entirely developed using open-source platforms, thus removing the price of software licenses as well as the fiscal burden on institutions.
- **Low-Cost Deployment Options:** The system is designed to be low-resource, for example, Render, Heroku, or minimal cloud VPS, so it is affordable for student or university budgets.
- **Low Overhead Maintenance:** Due to modular construction, changes and additions to features involve little redevelopment effort, allowing long-term maintenance costs to remain low.
- **Value Proposition:** The website enhances student visibility and enables team building—benefits that are well worth the relatively small development and maintenance costs for schools.

Thus, the solution is within the reach of widespread institutional adoption and is of lasting worth by virtue of being open infrastructure.

4.1.3 Operational Feasibility

Operational feasibility determines the usability, flexibility, and readiness of the system to be used in actual academic integration:

- **User-Centric Design:** The platform possesses a straightforward and easy-to-use interface, which allows students to navigate, create, and participate in projects without technic complexity.

- **Institutional Compatibility:** It is compatible with standard university information technology infrastructures, without the necessity for proprietary hardware or major infrastructure upgrades.
- **Role-Based Access:** Access controls within all modules permit varying treatment for students, mentors, and administrators—more functionality and security.
- **Scale to Growth:** The architecture can accommodate increasing numbers of users and projects and is suitable for institutional expansion or inter-college collaboration.
- **Minimum Training Requirement:** Because of its user-friendly design and simplicity, the system does not possess or require minimum training and can be operated by even non-technical users.

With excellent usability, smooth academic integration, and sound access management, Nexus Hub is operationally feasible for academic settings.

4.2 Software Requirement Specification

Software Requirement Specification (SRS) covers all functional and non-functional requirements for efficient development, deployment, and operation of the Nexus Hub platform. They encompass specific expectations on data treatment, system behavior, performance criteria, maintainability approach, and security enforcement..

4.2.1 Data Requirement

Nexus Hub is driven by structured and relational academic information that enables seamless coordination between administrators, mentors, and students. The data requirements have been established to allow traceable team construction, dynamic project listing, and safe access.

1. User Authentication Data:

The website saves each user's authentication history with hashed credentials and role-specific user IDs (student, mentor, admin). These history records authenticate the user properly when they log in and enable each user to view only features based on their role. Security elements such as session tokens and timeout controls guarantee data confidentiality.

2. Project metadata:

The project record for every project contains rich properties like title, full description, requested skill sets, preferred technologies, expected outcomes, and participation limits. This metadata will enable intelligent filtering of projects, improved searching, and candidate-project compatibility scoring in the future.

3. Application records:

Application data comprises time-stamped submissions from students with their associated LinkedIn and GitHub links, personalized messages, and project interest. Each application has a review status (pending, accepted, rejected), and the mentor's selections are stored for purposes of transparency and reproducibility of team selection outcomes.

Everything is stored in MongoDB with schema adaptability, indexing, and query optimization to ensure performance and security in data-access operations.

4.2.2 Functional Requirement

Functional specifications outline how the Nexus Hub system interacts with the user and how it delivers its services to fulfill necessary academic collaboration needs. These needs are divided into a series of operating modules:

A. User Access Authentication and Control

- The system must provide the functionality of allowing students, mentors, and administrators to register and safely log into their respective accounts.

Safe authentication methods such as bcrypt-hashed passwords and JWT tokens must be used in every session.

- Role-based access control (RBAC) needs to be implemented to enable adding and changing projects by mentors, submitting by students, and monitoring by administrators, with logical privilege separation across the system.

B. Project Initiation and Lifecycle Handling

- Mentors should be in a position to generate project postings with explanatory metadata fields like problem statements, domain restrictions, and expected outcomes.
- These should have the capability to be updated in real time and support the addition of external profile URLs such as GitHub repositories and LinkedIn handles to attract suitable candidates.
- The interface must contain a project dashboard where the mentors can search applications, track project status, and track team-building progress.

C. Application Review and Processing Process

- Students need to be able to upload projects through an interactive application form that records important information in addition to real external links.
- The system will need to facilitate mentors to review applications by presenting a brief overview of applicants' details to assist in accept/reject decisions.
- Once selected, the students must automatically be assigned to the project team, and appropriate notifications must be sent to all the concerned parties.

4.2.3 Performance Requirement

To support institutional-scale usage, the Nexus Hub platform must exhibit consistent performance under both regular and peak conditions. The following performance indicators define system reliability and user experience expectations:

- It must be capable of supporting at least 50 concurrent users without becoming unresponsive. Load balancing and asynchronous data loading may be used to make it scale.
- Project lists and their associated user profile URLs (e.g., GitHub) should be loaded within 2 seconds in normal network conditions.
- The frontend interface must be completely responsive and well-optimized for all types of devices, such as desktop, tablet, and smartphone, to make it accessible to all.
- The backend should be capable of handling bulk data operations like processing applications for scores of people simultaneously or presenting enormous amounts of project data without affecting latency or data integrity.

These goals ensure that the system is effective, scalable, and user-friendly in an actual academic setup.

4.2.4 Maintainability Requirement

Maintainability ensures that the Nexus Hub system can be updated in the future with minimal effort and without disrupting ongoing services. Both operational approaches and software architecture have been chosen to provide long-term adaptability goals.

- The codebase would be modular in reality, with modules such as user, project, and application workflows encapsulated into separately testable units. This will make upgrades, debugging, and feature extension simple.

- The MongoDB schema has to be schema-evolving, whereby new structure or properties can be introduced without destroying the existing functionalities.
- The application should have automated test scripts for regression testing and CI/CD integration for easy deployment.
- All modules must be accompanied by comprehensive technical documentation of architecture design, API endpoints, database models, and development best practices. This enables future developers to easily understand the system and improve it with minimal risk.
- There must be a repository version-controlled to maintain changes in system settings as well as source code.

These processes make the platform extensible and stable whenever new requirements of researchers come up.

4.2.5 Security Requirement

Security is of utmost importance to ensure the integrity and confidentiality of user data within Nexus Hub, especially in the use of personal data and academic work procedures. The following procedures and protocols must be in place:

- It should utilize encrypted storage of passwords (e.g., bcrypt) and token-based session management to prevent identity spoofing or credential theft.
- Role-based access policies must restrict unauthorized access to risky functions such as application review or system configuration.
- All the personal data, especially email addresses, external profile URLs, and application text, must be stored securely and sent using HTTPS.
- The system should be secure against common attack vectors like SQL injection, XSS, and CSRF. Input fields should be sanitized and error handling should never expose backend logic.

- Audit logs must capture all significant user activity, including login attempts, project changes, and application views. The logs must contain the following to support security audits: IP addresses and timestamps.
- A monitor module must scan periodically through log data to identify anomalies like multiple failed logins or high amounts of data accessed by an individual, which can be indicative of abuse or account breach.

Collectively, these security measures guarantee that the system aligns with contemporary-day cybersecurity requirements and is secure to handle academic work.

4.3 SDLC Model Used

While developing the Nexus Hub platform, the Iterative and Incremental Software Development Life Cycle (SDLC) model was applied. This model was applied due to the dynamic and adaptive nature of the requirements of the platform that called for continuous user feedback, constant polishing, and deployment in modules. Due to the educational and collaborative nature of the platform, there existed a need to build in cycles, incorporate real-world feedback of the users from the teachers and students, and incrementally add the functionality of the system.

Description of the SDLC Model

The Iterative and Incremental Model extends the disciplined approach of the Waterfall model with the extra assistance of prototyping. The system is constructed in a series of iterations in this model, in which new functionality is added and previous implementations are refined in each iteration. It is a suitable approach for websites like Nexus Hub, where user experience, user flow, and system logic can significantly be enhanced by early feedback and ongoing stakeholder involvement.

Each development cycle involved the following activities:

- Requirements Gathering and Analysis
- Design and Architecture
- Implementation
- Testing and Evaluation
- Feedback Integration
- Next Iteration Planning

In contrast to the conventional linear approach, every iteration in Nexus Hub was considered a complete version of the system. This made live testing and validation by stakeholders possible, as well as incremental improvement throughout the development process.

Rationale for Choosing This Model

The reasons for the selection of the Iterative and Incremental SDLC model were the following project-specific attributes:

- **Stakeholder-Centered Design:** The platform required regular interaction with students, mentors, and professors so that it could anticipate shifting user requirements, especially regarding project use, mentor selection, and GitHub/LinkedIn integration. The spiral approach allowed room for incorporating feedback after each cycle.
- **Progressive Feature Development:** Project creation, student application through external profile links, team creation, and application status were developed incrementally. Early release of core modules allowed for testing and tuning prior to the addition of further capabilities.
- **Component-Based Architecture:** The application was divided into independent modules like user authentication, project management, application review, and

admin control. These were coded incrementally and implemented in phases, thus making it testable and maintainable.

- **Agility in User Interface Design:** The user interface, built with React.js, went through multiple rounds of user testing. Each cycle helped in optimizing the UI/UX for clarity, responsiveness, and accessibility across devices.
- **Dynamic Requirement Evolution:** Requirements such as integration of third-party profile links (GitHub, LinkedIn) and dashboard-based mentor interactions were added mid-cycle based on user demand. The iterative approach allowed this flexibility without disrupting the overall system architecture.
- **Frequent Review and Feedback:** Continuous input was obtained from end-users through surveys, guided walkthroughs, and demonstration sessions. Early access builds were shared with a test group of students and faculty to identify pain points and validate workflows.

Application of the SDLC Phases in the Project

1. **Requirement Analysis:** Initial requirements were gathered through structured interviews and online feedback forms from students and faculty mentors. Key use cases identified included mentor-driven project creation, student profile-based project application, and real-time application status tracking.
2. **System Design:** The platform was designed with a modular architecture using MERN (MongoDB, Express.js, React.js, Node.js) stack. Component diagrams and user flowcharts were created to visualize modules such as authentication, project boards, and application pipelines.
3. **Implementation:** Development began with core modules such as user registration, login, and mentor project creation. Successive iterations added student application workflows, admin dashboards, and GitHub/LinkedIn URL

handling. RESTful APIs were used for data transactions between the frontend and backend.

4. **Testing:** Each module underwent unit and integration testing during development. Frontend interactions were tested using simulated inputs, and backend APIs were validated with test data. Functional testing focused on login security, data consistency, and application flow integrity.
5. **Evaluation and Feedback:** Evaluation was done internally at different milestones. Feedback was collected through structured user testing sessions involving students and mentors. Insights gathered were used to improve interface intuitiveness, error handling, and feature discoverability.
6. **Deployment and Maintenance:** The final version of the platform was deployed on a cloud hosting service supporting Node.js and React, with environment configuration and database integration completed for full-stack functionality. Post-deployment, the system was tested across devices for responsiveness. Basic maintenance involved fixing bugs, improving the UI, and updating features based on user feedback.

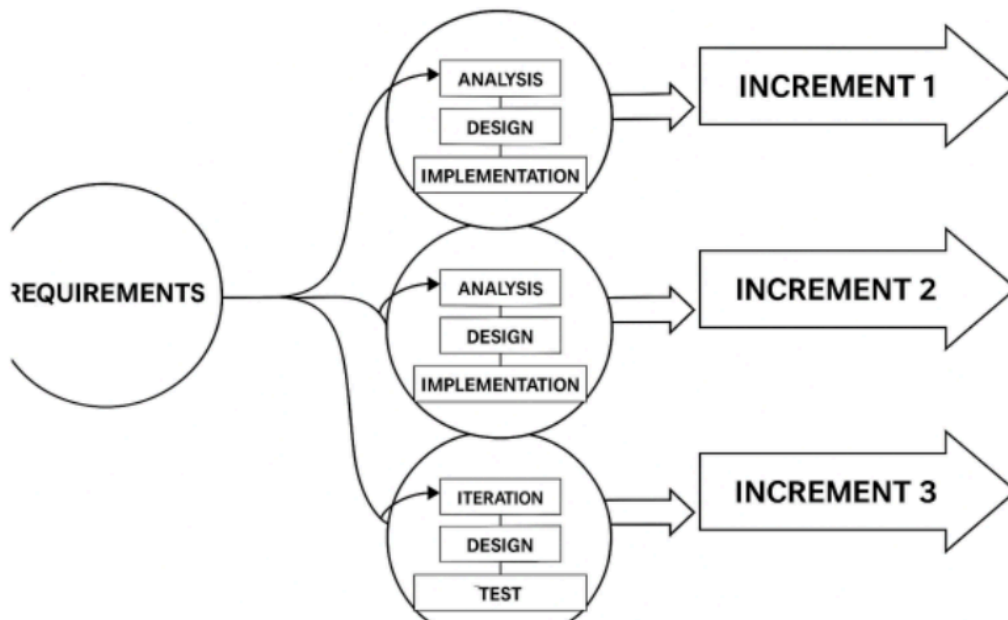


Fig 4.1: Iterative and Incremental Software Development Life Cycle

4.4 System Design

System design plays a critical role in translating the conceptual framework and functional requirements of Nexus Hub into a technically feasible, scalable, and modular platform for deployment. Given the focus on academic project management, the system was architected to support multiple user roles, robust data handling, and smooth interaction between various components. The design adopts a layered and modular architecture that ensures extensibility, maintainability, and seamless integration across modules responsible for project management, user access, application processing, and communication.

To support various functional needs—such as user registration, project tracking, mentor assignment, and status updates—the system was divided into logically distinct components that interact through clearly defined data flow mechanisms. Each component of the platform was developed in such a way that it can function independently and can be updated or scaled without disrupting the core functionalities. The design also incorporates best practices such as separation of concerns, modularity, and reuse to simplify future development and maintenance.

4.4.1 Architectural Overview

Nexus Hub comprises five principal architectural layers that collectively manage the flow of data and operations between users, system modules, and the central database:

1. User Interface (UI) Layer:

A web-based dashboard developed using ReactJS and Bootstrap, this interface allows users to register, create projects, manage teams, and review GitHub profiles. Designed for usability across roles (students, mentors, and recruiters), the interface ensures intuitive navigation and responsive interactions across devices.

2. Authentication and Role Management Module:

Built on secure authentication mechanisms using JWT and hashed credentials, this module controls user sessions and enforces role-based access permissions (student, mentor, recruiter). It ensures data confidentiality and secure operations across all user types.

3. Project Management Engine:

This module governs project creation, task assignment, application processing, and team composition. It allows users to manage workflows, track progress, and assign tasks. Key logic includes access-based task visibility and dynamic project status tracking.

4. GitHub Integration Module:

Facilitates user profile enrichment by enabling linkage of GitHub accounts. This allows project creators to review collaborators based on actual code contributions, repositories, and activity metrics, providing a unique data-driven candidate assessment layer.

5. Database and Storage Layer:

MongoDB Atlas serves as the backend database, managing collections for users, projects, applications, and team members. The document-based schema is designed to support rapid queries and efficient data relationships while ensuring data integrity and scalability.

4.4.2 Data Flow Design

The internal system workflow is described in the steps below:

1. **User Input:** A user signs in and initiates project creation or team application via the web dashboard.
2. **Authentication:** Credentials are verified, and access rights are granted according to the user's role.

3. **Project Handling:** New projects are created and stored in the database; users can assign tasks or accept team applications.
4. **GitHub Profile Linking:** Users link their GitHub profiles, enabling project owners to assess past contributions.
5. **Data Retrieval & Display:** Relevant project and user data is dynamically retrieved and displayed through the frontend.
6. **Access Control:** The system filters actions and visible data based on user roles (e.g., only mentors can approve team applications).
7. **Collaboration:** Team members collaborate through assigned roles and monitor task progress in real-time.

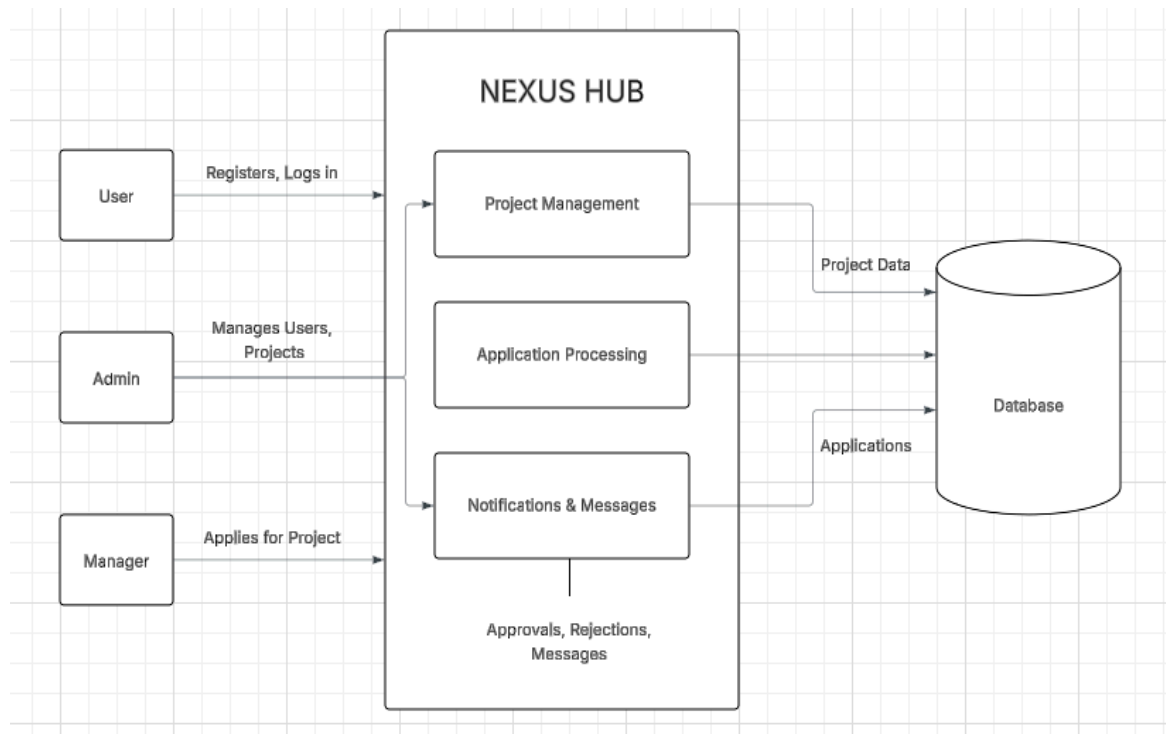


Fig. 4.2: Data Flow Diagram

4.4.3 Use Case Diagram

The use case diagram captures Nexus Hub's functionality requirements in terms of actors, system boundaries, and workflow interactions. It adheres to UML 2.5 standards and captures the core functionality of the system as described in the SRS.

- **Actors and Their Roles**

- 1. Student:**

- Project Creator: Activates use cases Create Project, Manage Project, and Review Applications.
- Applicant: Conducts Browse Projects, Apply to Project, and Check Application Status.

- 2. Mentor:**

- Monitors system state with Monitor Projects.
- Creates analytics using Generate Reports.

- 3. System:**

- Automated tasks: Send Notifications and Keep the Profiles (GitHub sync).

- **Key Use Cases**

Use Case	Description	Actor
Create Project	Define title, skills, and team requirements	Student (Creator)
Apply for Project	Submit GitHub profile for review	Student (Applicant)

Accept/Reject Applicants	Finalise team composition	Student (Creator)
Monitor Projects	View project progress (read-only)	Mentor

Table 4.1: Principal Use Cases mapping onto the Use Case Diagram

- **Relationships**

1. **«Include» Dependencies**

- Apply to Project must precede execution of Browse Projects.
- Review Applications is reliant on Manage Project initiation.

2. **«Extend» Relationship**

- Accept/Reject Applicants adds decision-making to the fundamental Review Applications workflow.

- **System Boundary**

All use cases fall within the boundary of the Nexus Hub (depicted as a rectangle), highlighting that:

- Actors are outside the system but interact with it.
- System (Automated) and Student/Mentor (manual) processes are clearly distinguished.

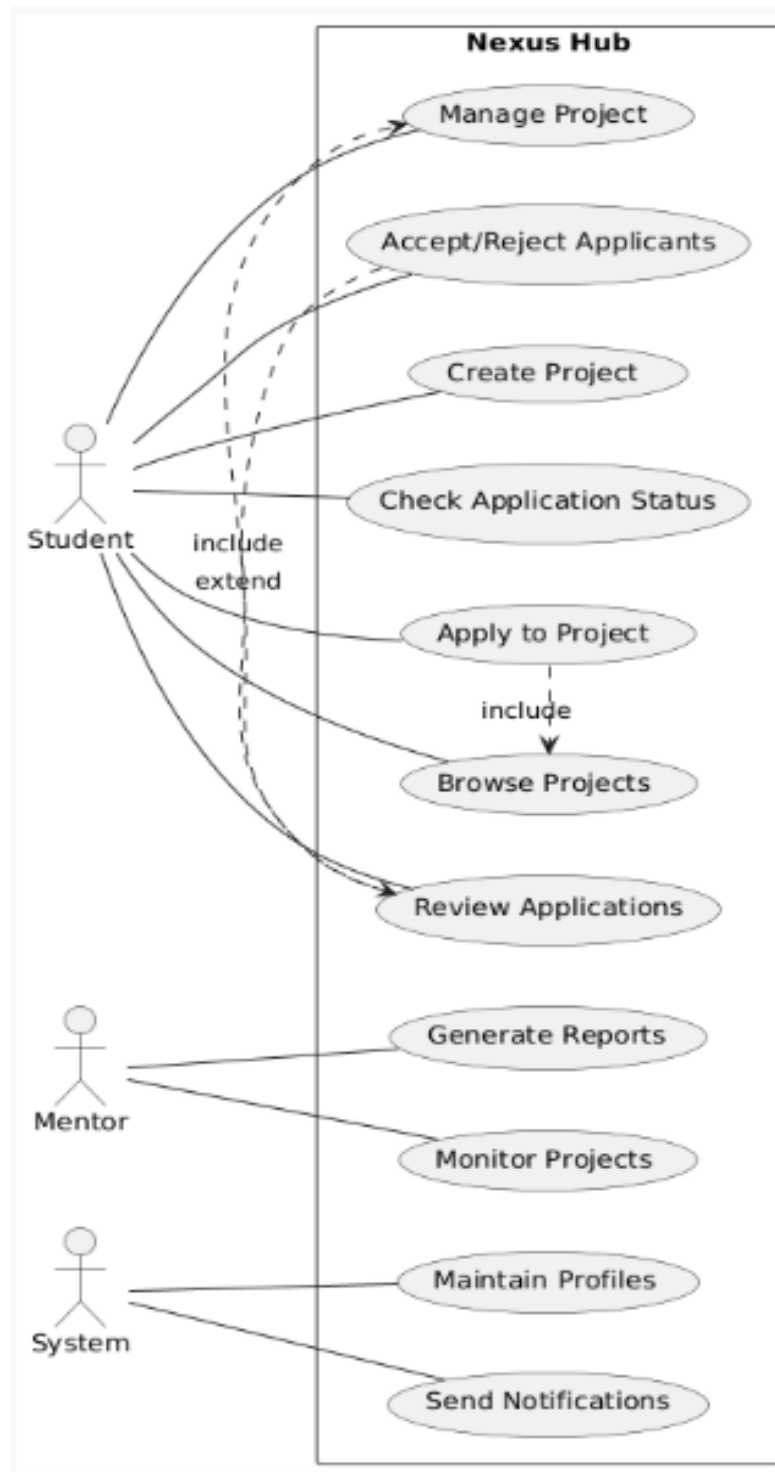


Fig. 4.3: Use Case Diagram

4.4.4 Component-wise Design

Module	Technology/Tool Used	Functionality
Frontend (UI)	ReactJS, Bootstrap	Provides interactive web dashboard for project management and GitHub linking.
Authentication	JWT, bcrypt, Express.js	Secures user access with hashed login and role-based authorization.
Project Engine	Node.js, Express.js, MongoDB	Handles project creation, applications, and task workflows.
GitHub Integration	GitHub API	Enables user profile enhancement and candidate evaluation.
Database	MongoDB Atlas	Stores all system entities: users, projects, applications, and team data.

Table 4.2: Component-wise Design

4.5 Database Design

The database schema of Nexus Hub has been architected to efficiently manage, store, and retrieve diverse data entities central to academic project collaboration. These include user credentials, project details, team applications, GitHub-linked profiles, task assignments, and role-based access records. The design follows NoSQL principles optimized for flexibility, scalability, and rapid querying, which are critical in a collaborative, multi-user environment.

Nexus Hub uses MongoDB Atlas as its primary database engine. MongoDB's document-based schema aligns well with the dynamic and hierarchical data structures needed to manage nested project data, variable user roles, and application histories without sacrificing performance or extensibility.

Objectives of Database Design

- To maintain secure and structured storage of user profiles and authentication credentials.
- To facilitate efficient project creation, listing, and team member tracking.
- To enable linking and assessment of external GitHub profiles for candidate evaluation.
- To support role-based workflows (student, mentor, recruiter) using access metadata.
- To ensure flexibility in project lifecycle management, from initiation to closure.
- To minimize redundancy while optimizing data retrieval performance for large-scale use.

Key Collections and Relationships

The database includes the following primary collections:

1. users Collection

Stores information related to all registered users.

Field Name	Type	Description
user_id	ObjectId (PK)	Unique identifier for the user
name	String	Full name of the user
email	String	Unique email address
password_hash	String	Hashed password for secure login
role	String (Enum)	student / mentor / recruiter
github_link	String	Link to the user's GitHub profile
created_at	Timestamp	Account registration timestamp

Table 4.3: users Collection Schema

2. projects Collection

Stores details of all student-led projects hosted on the platform.

Field Name	Type	Description
project_id	ObjectId (PK)	Unique project ID
creator_id	ObjectId (FK)	Links to user who created the project
title	String	Title of the project
description	String	Brief summary or abstract
tech_stack	Array	Technologies used in the project
github_repo	String	Linked GitHub repository
status	String (Enum)	open / closed
created_at	Timestamp	Timestamp of project creation

Table 4.4: projects Collection Schema

3. applications Collection

Tracks student applications to join existing projects.

Field Name	Type	Description
application_id	ObjectId (PK)	Unique identifier for the application
applicant_id	ObjectId (FK)	Refers to the applying user
project_id	ObjectId (FK)	Project being applied to
status	String (Enum)	pending / approved / rejected
reason	String	Reason for approval/rejection (optional)
applied_on	Timestamp	Application submission timestamp

Table 4.5: applications Collection Schema

The document-based schema in MongoDB offers Nexus Hub the agility needed for managing dynamic, role-specific interactions and project workflows. Collections are carefully normalized to support data integrity, reduce redundancy, and enable seamless scaling as user engagement and project volume increase. With its efficient access design and robust entity relationships, the database underpins a secure and collaborative environment for academic project coordination.

CHAPTER 5

IMPLEMENTATION

5.1 Tools and Technologies Used

The Nexus Hub platform is built as a scalable, cloud-enabled solution focused on project-based academic collaboration. The implementation leverages full-stack web technologies, cloud-native tools, and open-source frameworks to ensure real-time responsiveness, efficient task handling, and robust user authentication. The various technologies employed are categorized as follows:

1. Frontend Development

- **ReactJS** – A JavaScript library used to develop the platform’s dynamic and responsive user interface. It allows for efficient component rendering and provides a seamless single-page application (SPA) experience.
- **Bootstrap** – Used to style UI components and ensure cross-device compatibility with a mobile-first design approach.

2. Backend Development

- **Node.js** – A runtime environment that powers the backend logic of the application. It handles API requests, session management, and project operations with high throughput.
- **Express.js** – A lightweight web application framework used alongside Node.js for building RESTful APIs, routing, and middleware integration.

3. Database and Storage

- **MongoDB Atlas** – A fully managed cloud-based NoSQL database used to manage collections such as users, projects, applications, and

authentication records. It supports flexible schemas, fast querying, and horizontal scalability.

4. Authentication and Security

- **JSON Web Tokens (JWT)** – Used to securely manage session authentication and role-based access control (RBAC).
- **bcrypt** – Employed for password hashing and secure credential storage to prevent unauthorized access.

5. GitHub Integration

- **GitHub API** – Integrated to allow users to link their GitHub profiles to their Nexus Hub accounts. This enables profile evaluation, repository visibility, and collaboration tracking directly from the platform.

The above technologies were integrated through a modular, layered architectural style, providing logical partitioning between the user interface, business logic, and data storage modules. All tools had a particular and intentional role in providing a production-level system that fulfills the functional needs of academic project management.

CHAPTER 6

TESTING & MAINTENANCE

6.1 Testing Techniques and Test Cases Used

Introduction to Testing

Testing played a pivotal role in validating the functionality, security, and overall reliability of the Nexus Hub platform. As a multi-role, collaborative project management system, it was essential to ensure that every module—from user authentication to GitHub

integration and team application workflows—operated as intended. Testing was conducted iteratively throughout development to identify issues early, refine features, and meet performance benchmarks.

Testing Strategy

The system was subjected to a combination of unit testing, integration testing, and end-to-end system testing to ensure smooth functioning in all modules. Real-time validation, data consistency, and secure access control were highlighted throughout the test cycle.

Test Levels

1. Unit Testing

Each individual module was checked for internal logic and output according to expectations.

Example: Checking that the GitHub API is giving correct profile details and repository URLs.

2. Integration Testing:

Modules were tested and deployed in pairs or groups to offer interaction fidelity.

Example: Confirming that the project application acceptance places a student on the team immediately and updates the project status.

3. System Testing

The platform was deployed in a simulated production environment to observe its behavior under realistic usage conditions. Performance was evaluated with multiple concurrent users simulating students, mentors, and recruiters.

Test data covered various user roles, project scenarios, and GitHub integrations.

Test Cases and Their Execution

Test Case ID	Test Scenario	Expected Outcome	Status
TC-001	Verify secure login for registered users	User should be redirected to dashboard with appropriate role-based access	Pass
TC-002	Attempt login with incorrect credentials	User should receive an authentication failure message	Pass
TC-003	Submit a new project with valid details	Project should be saved and visible in the user's project list	Pass
TC-004	Apply to join a project as a student	Application should be submitted and reflected in the project owner's dashboard	Pass
TC-005	Attempt to view GitHub profile of a team applicant	Linked GitHub profile should open in a new tab	Pass
TC-006	Validate role-based access (e.g., mentor cannot join a project)	System should prevent unauthorized actions and display appropriate message	Pass

Table 6.1: Executed Test Cases and Results

Bug Identification and Fixes

Several defects were detected and resolved during the QA cycles:

- Bug #001: Inconsistent rendering of dashboard components on mobile devices
 - Fix: Applied responsive styling using Bootstrap grid classes.
- Bug #002: Project creation failed with special characters in title
 - Fix: Added frontend and backend sanitization for input fields.
- Bug #003: GitHub profile link not saving correctly for some users
 - Fix: Updated schema validation and regex matching for URLs.
- Bug #004: Application status not updating after mentor approval
 - Fix: Refactored backend logic to correctly reflect status transitions in the database.

The structured testing framework ensured that Nexus Hub met all performance, usability, and security benchmarks. The comprehensive coverage—from unit logic validation to full-scale system simulations—enabled confident deployment and seamless user experience. With critical bugs resolved and robust test cases in place, Nexus Hub is production-ready and reliable for use across academic institutions.

CHAPTER 7

RESULTS & DISCUSSIONS

7.1 Description of Modules

The performance of Nexus Hub was analyzed based on the functionality and integration of its core modules. Each module was evaluated through functional testing, usage simulation, and UI validation to ensure that it meets the objectives of academic project management and team collaboration. The following sections describe the key modules of the system.

User Dashboard Module

This module acts as the centralized interface for users, primarily students, to monitor their academic projects. Upon successful login, users are directed to a personalized dashboard displaying essential project-related information such as current active projects, project details, and team members.

This module enables users to manage their projects, view project profiles, and get quick access to project-related updates. The responsive design ensures that users can easily navigate between sections, whether on a desktop or mobile device.

Project Management Module

This module allows students and admins to create, assign, and monitor various academic projects. The Project Management module offers functionalities for:

- Building New Projects with project names, descriptions, and goals.
- Allocation of Team Members to specific projects.
- Monitoring Project Progress with brief status and progress indicators

The project management functionalities make sure that everything is properly organized, and every member knows what is expected of them. Admins are able to assign roles and check project statuses to keep the users of the platform in line.

GitHub Integration Module

The GitHub integration module allows users to include a direct link to their GitHub profile on their platform profile. This functionality encourages collaboration by providing students and admins immediate access to each user's repositories, where they can review code, academic assignments, and other relevant work. Its key features are:

- **Linking GitHub Profiles:** Members can add their GitHub link to their profile such that others can view their repositories and their contributions.
- **External Review:** By including their GitHub handle, users enable mentors, admins, and peers to review their code and school work directly from GitHub.

This module facilitates collaboration and openness through offering students a platform to post their coding exercises and academic assignments on GitHub.

7.2 Key Findings

The Nexus Hub platform offers a number of significant observations and findings that are central to bringing the success of the project into focus as an educational tool:

1. Improve Collaboration and Communication

The platform provides regular communication and collaborative tools among project members and mentors. The platform enables an improved team environment through real-time updates, GitHub repository visibility, and direct messaging. These aspects help diminish miscommunication and improve synchronization among project remote members.

2. Enhanced Project Management

Task assignment and project monitoring are better with structured creation and role assignment. Progress reporting and deliverable submission allow mentors to monitor timelines as well as hold them responsible. Such functions improve academic rigor by allowing open transparency and encouraging on-time completion.

3. Flexibility and Scalability

The system is modular and document-based, and therefore it is possible to scale user data in addition to increasing project requirements. The system is also simple

to switch between institutions or departments, and the design base allows one to introduce new modules without influencing existing functionality. The platform is future-proof and scalable to accommodate large-scale academic use cases.

7.3 Brief Description of Database with Snapshots

The Nexus Hub database is designed to provide core features such as project creation, project membership application, potential team member evaluation, and project membership management. It is document-based on the back of MongoDB Atlas for scalability and flexibility with secure data processing. A brief description of the database structure and core components is given below:

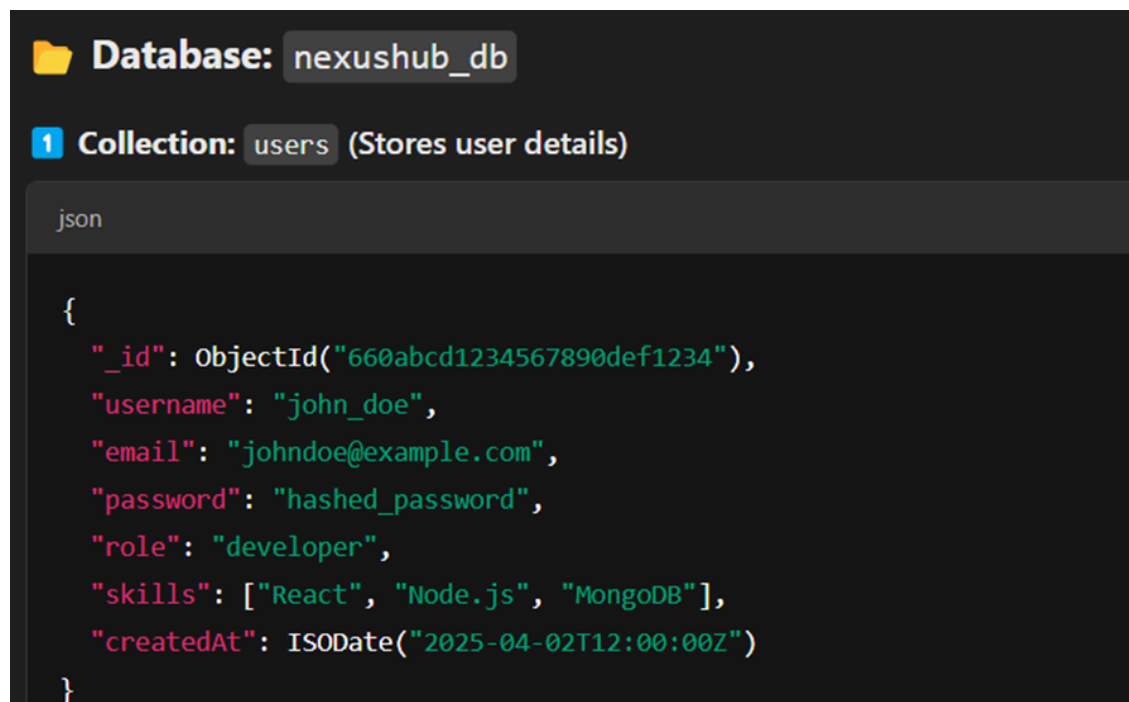
1. Users Table

The users table has all the information about the users, including their role, profile, and GitHub link. The user ID is unique for every user and is referenced by various entities in the system.

Key Fields: The Users table contains a number of key fields containing in-depth information on each registered user. The User ID is a unique identifier allowing each user to be uniquely referred to throughout the system. Apart from this, the Name field contains the full user name, used for display and identification purposes throughout the site. The Email address is stored to allow communication and account verification, and the Password is stored securely in an encrypted format to protect user access. Each user is assigned a specific Role, for instance, Student, Mentor, or Admin, which determines the level of access and permissions they have within the site. Also, the Skills field stores the competencies and technologies that users are proficient in, aiding in project matching and collaboration. Finally, the Creation Date stores when the user account was initially created, allowing tracking of registration timelines.

Key Relationships: The Users table is central to the database, with numerous others referencing it. A user can be related to the Projects table by User ID, being used in fields like Owner ID and Team Members to reflect project ownership and membership. This is a many-to-many relationship because one user may be a member of many projects, and one project can comprise many users. Users are also referenced on the Applications table, where one user may apply to be a member of many projects, creating a one-to-many relationship. The Messages table also references the Users table, where each message can trace a sender and a recipient by their respective User IDs. The Notifications table also references users to send system notifications or updates. These multiple relationships make the Users table a central piece in maintaining the integrity of the platform and allowing collaboration.

Snapshot:

A screenshot of a database interface showing a snapshot of the 'users' collection. The interface has a dark theme. At the top, it says 'Database: nexushub_db'. Below that, it says 'Collection: users (Stores user details)'. The data is shown in a 'json' format. The document is a JSON object with the following fields: '_id' (ObjectId), 'username' (string), 'email' (string), 'password' (string), 'role' (string), 'skills' (array of strings), and 'createdAt' (ISODate).

```
{
  "_id": ObjectId("660abcd1234567890def1234"),
  "username": "john_doe",
  "email": "johndoe@example.com",
  "password": "hashed_password",
  "role": "developer",
  "skills": ["React", "Node.js", "MongoDB"],
  "createdAt": ISODate("2025-04-02T12:00:00Z")
}
```

Snapshot 7.1: Snapshot of Database (Users Table)

2. Projects Table

The Projects table stores the overall information for all of the academic projects. It has fields like the project name, description, and objectives, and the User ID (foreign key) referring to the users (team members) of the project.

Key Fields: The examples of such important relationships are in the Projects table. The Project Title provides a short but descriptive title that summarizes the project, and the Project Description provides a detailed description of the project goals, scope, and objectives. The Owner ID links each project to its owner or main administrator by a reference to a user in the Users table, establishing ownership. The platform also tracks the Required Skills needed to work on the project successfully, which assists in assigning the appropriate team members. The Team Members field is a list of User IDs that are the project members, highlighting the team work nature of the project. The status of the project is tracked to indicate its current status, e.g., active, completed, or pending. Lastly, the Creation Date tracks the date the project was created, facilitating project lifecycle management.

Key Relationships: The Projects table maintains several important relationships with other tables in the database. Each project is associated with a User who owns it, as indicated by the Owner ID, and may include multiple users as team members, forming a many-to-many link between projects and users. The Applications table is another key component connected to projects, with each application referring back to a specific project through its Project ID. This one-to-many relationship allows a project to receive multiple applications from different users. Additionally, projects are referenced in the Messages table, where users may exchange messages related to specific projects. These relationships ensure that all interactions—whether application submissions or communications—are contextually tied to their respective projects.

Snapshot:

```
2 Collection: projects (Stores project details)

json

{
  "_id": ObjectId("660abcd1234567890def5678"),
  "title": "AI-Based Resume Screener",
  "description": "An AI-powered resume screening tool for recruiters.",
  "ownerId": ObjectId("660abcd1234567890def1234"),
  "requiredSkills": ["Python", "Machine Learning", "Flask"],
  "teamMembers": [ObjectId("660abcd1234567890def9999")],
  "status": "Open",
  "createdAt": ISODate("2025-04-02T14:30:00Z")
}
```

Snapshot 7.2: Snapshot of Database (Projects Table)

3. Applications Table

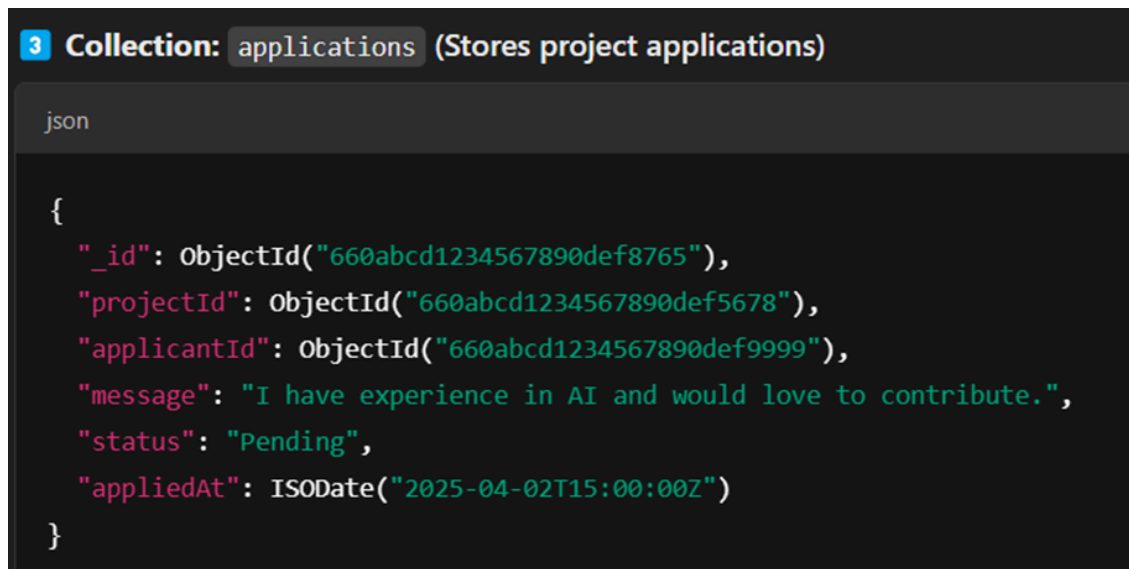
The Applications table stores data about student applications to join projects. This table contains references to both the Projects and Users tables through foreign keys.

Key Fields: The Applications table holds data on user submissions to join projects. Each application is identified by a unique Application ID. The Project ID field connects the application to the specific project the user is interested in, while the Applicant ID links the application back to the user submitting it. Applicants can include a personalized Message that explains their motivation or qualifications for joining the project. The Status field tracks the current state of the application, such as pending review, approved, or rejected. Additionally, the Application Date notes when the application was submitted, allowing for chronological tracking and timely processing.

Key Relationships: The Applications table acts as a bridge between users and projects. Each record in this table establishes a link between a User (the applicant)

and a Project (the one being applied to). This creates a one-to-many relationship from both sides: a user can submit several applications to different projects, and a project can receive multiple applications from various users. These connections allow the platform to track application activity, manage application statuses, and facilitate the matching of users to relevant projects.

Snapshot:

A screenshot of a database interface showing a collection named 'applications'. The title bar indicates it stores project applications. The data is shown in JSON format. The JSON object contains fields for _id, projectId, applicantId, message, status, and appliedAt, all with specific ObjectId or ISODate values.

```
3 Collection: applications (Stores project applications)
json
{
  "_id": ObjectId("660abcd1234567890def8765"),
  "projectId": ObjectId("660abcd1234567890def5678"),
  "applicantId": ObjectId("660abcd1234567890def9999"),
  "message": "I have experience in AI and would love to contribute.",
  "status": "Pending",
  "appliedAt": ISODate("2025-04-02T15:00:00Z")
}
```

Snapshot 7.3: Snapshot of Database (Applications Table)

4. Messages Table

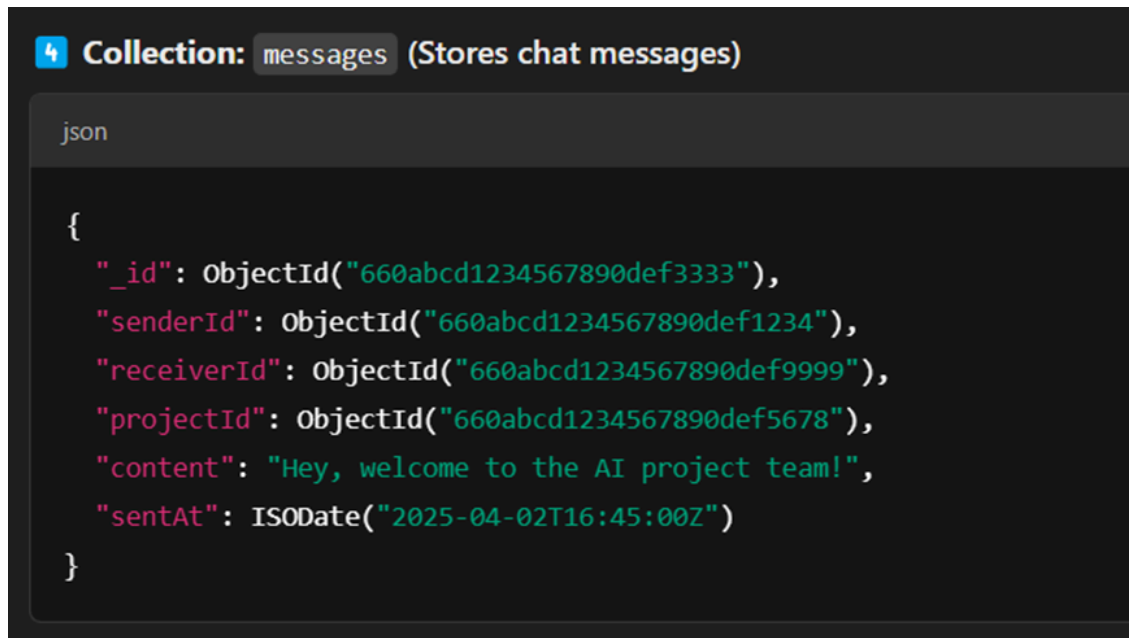
The Messages table allows the users to communicate with each other, storing incoming and outgoing messages. The table has User ID references for users (sender and receiver).

Key Fields: The Messages table enables users to talk to one another by storing every message with a unique Message ID. It identifies the sender and receiver with the Sender ID and Receiver ID, respectively, which point to users in the system. Messages generally point to single projects, and that is indicated by the Project ID field, and puts things into context for the conversation. The Content

field holds the message text, and the Sent Date holds the exact time the message was sent, for ordering in time and tracking conversation history.

Key Relationships: Relational relationships within the Messages table include the sender and receiver columns, both referencing back into users within the Users table. The back-reference forms two one-to-many relationships on messages received and messages sent by a user. A message may also have an association with a specific Project, so conversations are contextually relevant. These relationships enable structured, project-based conversation and ensure user activity records within the system are unambiguous.

Snapshot:

A screenshot of a database interface showing a JSON document for a message. The interface has a dark theme. At the top, there's a header bar with a blue lightning bolt icon, the text "Collection: messages", and "(Stores chat messages)". Below this is a tab labeled "json". The main area displays a JSON object with the following fields: "_id" (ObjectId), "senderId" (ObjectId), "receiverId" (ObjectId), "projectId" (ObjectId), "content" (string), and "sentAt" (ISODate).

```
{
  "_id": ObjectId("660abcd1234567890def3333"),
  "senderId": ObjectId("660abcd1234567890def1234"),
  "receiverId": ObjectId("660abcd1234567890def9999"),
  "projectId": ObjectId("660abcd1234567890def5678"),
  "content": "Hey, welcome to the AI project team!",
  "sentAt": ISODate("2025-04-02T16:45:00Z")
}
```

Snapshot 7.4: Snapshot of Database (Messages Table)

5. Notifications Table

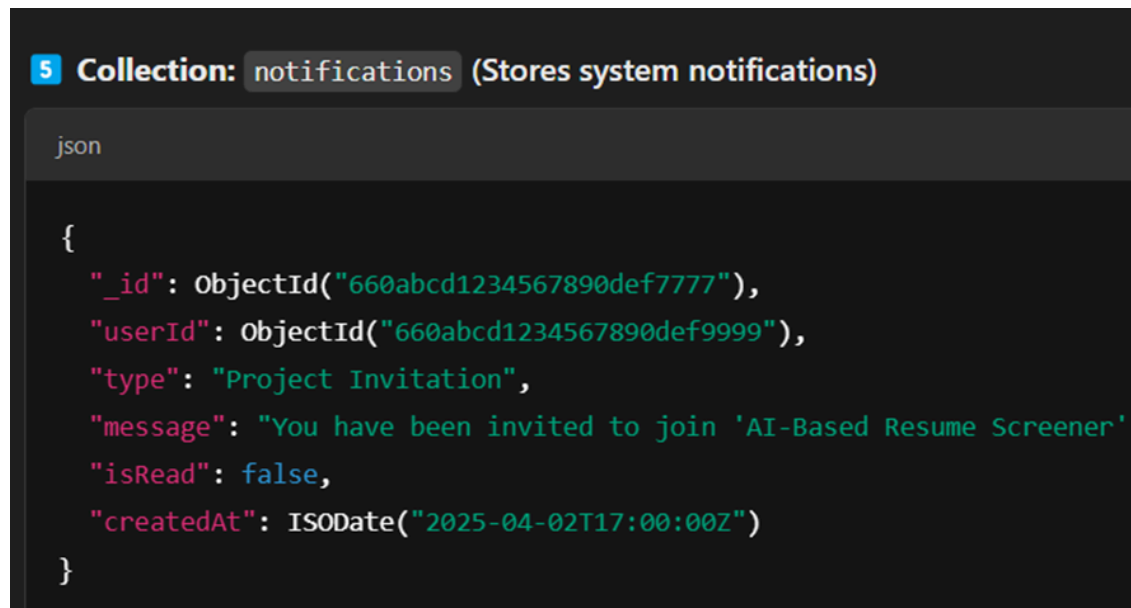
The user notifications are kept in the Notifications table, which reminds them of important actions or changes regarding their projects, messages, or systems.

Key Fields: The Messages table allows for users to exchange messages with one another by storing each message with an individual Message ID. The receiver and

sender are resolved by the Sender ID and Receiver ID, which both reference users of the system. Messages will likely be regarding a specific project, which is indicated by the Project ID field, giving context to the conversation. The Content field stores the actual body of the message itself, and the Sent Date stores exactly when the message was sent, enabling chronological ordering and monitoring of discussion history.

Key Relationships: The Notifications table has a basic but necessary relationship with the Users table. The notification is associated with a specific user with the User ID, forming a one-to-many relationship where a single user can have multiple alerts during the term. The relationship is employed to alert users of occurrences such as project changes, new messages, or important administrative changes, enhancing engagement and timely communication on the platform.

Snapshot:

A screenshot of a database interface showing a document in the 'notifications' collection. The document is a JSON object with fields for _id, userId, type, message, isRead, and createdAt. The message content is 'You have been invited to join 'AI-Based Resume Screener''. The createdAt field is an ISODate. The interface has a dark theme with a blue header bar.

```
5 Collection: notifications (Stores system notifications)

json

{
  "_id": ObjectId("660abcd1234567890def7777"),
  "userId": ObjectId("660abcd1234567890def9999"),
  "type": "Project Invitation",
  "message": "You have been invited to join 'AI-Based Resume Screener'",
  "isRead": false,
  "createdAt": ISODate("2025-04-02T17:00:00Z")
}
```

Snapshot 7.5: Snapshot of Database (Notifications Table)

This dynamic, document-based structure is readily conformable to the data structure when new features are introduced into the platform.

Key Relationships Summary

The Nexus Hub database is designed on a well-defined set of interconnected collections that facilitate smooth operation throughout the platform. One of the most important relationships is between Users and Projects, and a many-to-many relationship which enables a user to contribute to multiple projects, and a project to have multiple users as team members. This facilitates collaborative collaboration and flexible assignment of projects. There is also a one-to-many relationship between Users and Applications, as one user can apply to multiple projects, and each application uniquely represents the relationship between a user and a given project.

Supporting the project management cycle, Projects also share a one-to-many relationship with the Applications table, with multiple users able to submit an application to join a single project. Applications are monitored individually, enabling project owners to manage incoming interest. The Messages table builds on this interdependent system by creating dual one-to-many relationships with the Users table—each user can send and receive messages, and their User ID is logged as the sender or receiver in each message entry. Messages can also be assigned to specific projects, bridging conversation to context of co-working.

The Notifications table has another one-to-many relationship with the Users table to enable users to be notified with more than one message depending on activity, role, or interaction in the system. The notifications are crucial to notify the users in a timely manner about project updates, messages, application updates, or system announcements. Collectively, these relationships form a solid relational infrastructure that not only enables the day-to-day operation of Nexus Hub but also preserves data integrity, consistency, and scalability as the platform expands.

CHAPTER 8

CONCLUSION & FUTURE SCOPE

The absence of a unified platform to facilitate coordination of academic projects leads to poor communication, delayed progress tracking, and poor coordination among students and lecturers. Such disparate procedures not only delay the overall project process but also reduce the extent of mentorship and guidance, which are vital in the development of students' academic competencies. Such inefficiencies also lead to the variation in the quality of projects, inability to adhere to deadlines, and lack of information regarding the status and milestones of the projects.

To meet these needs, Nexus Hub was created as an all-in-one web-based platform that is specially designed to fulfill the project management needs of students and teachers in schools. Nexus Hub provides a unified platform that includes significant project management tasks like creating teams, assigning projects, monitoring progress, and mentoring feedback. Leveraging modern web technologies such as ReactJS, Node.js, and MongoDB, Nexus Hub offers a safe, scalable, and user-friendly system that facilitates collaboration and coordination throughout the whole project life cycle.

The most recent version of Nexus Hub has been rolled out and hosted successfully, and it is now live in real-time. It provides key features like secure login, project registration, guide assignment, and project upload through an easy-to-use interface. Even though it is designed at the student level, the system is stable and flexible enough to be implemented at the school level. The system provides coordination of projects by removing gaps in communication and inefficiencies that are usually encountered in traditional project management processes. The system also facilitates teamwork among team members, guides, and staff members, making the process of adopting the project more integrated and transparent.

The creation of Nexus Hub has gained useful insights into numerous aspects of full-stack development, including modular design, database integration, and user-centered interface design. Being an undergraduate student project, Nexus Hub is a solid platform where additional improvements and upgrade can be made, and with a possibility of its widespread adoption in educational institutions in the future.

8.1 Future Scope

While the Nexus Hub system has been successful in fulfilling its first goals of addressing the current deficiencies of project management in academic settings, it is very promising for future development and expansion. The system can be greatly improved in future releases, with increased dynamicity and functionality. Of the most critical areas for future development are:

1. Progress Tracking System

To enhance the project management experience, there has to be an integrated system for tracking progress which will enable students, mentors, and faculty members to view project milestones, phases, and key deliverables in real time. A color-coded visual dashboard with progress bars or percentage indicators will give stakeholders a clear, easy-to-consume view of the status of a project. The system will enable students to maintain deadlines and provide mentors and faculty with a clear picture of the project's progress, enabling them to intervene early if necessary to offer guidance or assistance.

2. Task Management Tools

Another crucial feature of effective project management is the ability to delegate and assign individual project tasks. Using tools such as Kanban boards or Gantt charts, students and teachers can plot project timelines, assign tasks, and track progress more effectively. The tools also have the ability to break down massive projects into small tasks and ensure that all project areas receive attention within

given time. The tools can also provide reminders when tasks are completed or overdue, helping students have a well-structured workflow.

3. Mentor Feedback System

A formal mechanism of feedback by the mentors will guarantee ongoing mentorship throughout the project life cycle. Through provision of feedback at various stages of the project, from planning right up to implementation and final submission, the system will guarantee ongoing improvement and fine-tuning. Feedback can be provided in the form of standardized forms or checklists that assess progress of the project and highlight areas that need improvement. This feature will offer students constructive feedback and allow them to make incremental changes to their work, leading to better-quality outputs.

4. Automated Alerts

Using automated reminders will inform all parties involved of key project deadlines, meetings, and submission dates. The reminders can be issued through email, in-app notification, or push, reminding the users of outstanding tasks, deadlines, review sessions, or action needed. Automated reminders will ensure the platform minimizes missing key deadlines and enhances better time management for students and educators.

5. GitHub Integration

To enable collaboration and code-based project development more effectively, more integration with GitHub is among the most crucial features to be implemented in upcoming releases. By direct linking students' GitHub repositories to Nexus Hub, students can seamlessly sync project code, commits, pull requests, and version history within the platform. This will also allow faculty mentors to track code-based project development without the students manually doing so. It will also allow collaboration among team members on shared repositories, thus making the entire project management process more seamless.

6. Analytics Dashboard

An analytics dashboard will provide students and faculty useful insights into project performance, user interaction, and engagement rates. The tool will chart statistics about project status, user contribution, and overall progress so that users will be able to view where they are and where they can improve. The dashboard can include charts and graphs to display project milestones, tasks completed, open topics, etc., thereby facilitating decision-making based on data and improved collaboration.

7. Document Sharing Module

A secure document-sharing feature is essential in learning via projects because the students and instructors need to collaborate on various types of project-related documents, such as project plans, reports, and presentations. The ability to upload, share, and view the documents in one place will make the collaboration smooth. This feature will ensure that all the project-related documents are at one place, reducing the loss of important documents and allowing it to easily track users' work.

8. Mobile App

With the growing mobile lifestyle of students today, one of the most essential elements of future growth is creating a mobile application for Nexus Hub. A mobile application would allow students, mentors, and teachers to view project information and interact with team members remotely at any time. The application would potentially feature necessary functionalities such as progress tracking, reminders, task assignment, and file sharing, enabling users to stay connected to projects on the move. This would enhance user access and interaction, particularly for students who are always on the move.

These future improvements will further increase the usability of the platform by making it all the more worthwhile for use in academic project management and collaboration. Further development and improvement of the platform will allow Nexus Hub to provide an even more comprehensive suite of tools for students, mentors, and faculty, further

enhancing the overall academic experience and fostering better teamwork, productivity, and project outcome.

8.2 Final Thoughts

The Nexus Hub project has proved to be an efficient project in the building of an integrated system to consolidate and make school project management streamlined. Through solving common issues like fragmented communication and poor project monitoring, the system has offered enhanced collaboration, coordination, and project quality.

With its strong and responsive architecture, Nexus Hub has enabled the platform for a more efficient, user-focused method of handling student projects. The platform has provided a complete view of the complexity of full-stack development, whether from system design to database management and integration with the frontend. The iterative development model has also allowed the team to build the system based on real testing and end-user feedback, and the platform is therefore optimized to the needs of its end-users.

As it grows, the future of revolutionizing project-based learning in schools comes into focus more clearly. As it matures and grows further, the site will be ready to make lasting changes in how students and teachers collaborate on school projects. Nexus Hub's ongoing innovation will dictate the future of school project management, empowering students with the means to thrive in a growing collaborative, online world.

REFERENCES

- [1] J. Smith, and A. Johnson, "Enhancing Collaborative Learning Through Online Project Management Platforms," *Journal of Educational Technology & Society*, vol. 26, no. 4, pp. 112-125, 2023.
- [2] L. Brown, and K. White, "Leveraging Technology for Project-Based Learning in Higher Education," *International Journal of Educational Technology in Higher Education*, vol. 18, no. 1, pp. 1-15, 2023.
- [3] R. Gupta, and S. Sharma, "Design and Development of a Collaborative Project Management Platform for Higher Education Institutions," *International Journal of Information Management*, vol. 56, pp. 102175, 2021.
- [4] Y. Kim, and S. Lee, "Utilizing Online Platforms for Enhancing Student Engagement in Project-Based Learning," *Computers & Education*, vol. 130, pp. 1-10, 2023.
- [5] D. Anderson, and M. Jackson, "Building Community in Online Learning Environments: Strategies for Project-Based Courses," *Journal of Online Learning and Teaching*, vol. 14, no. 2, pp. 25-36, 2022.
- [6] J. Dillman, and C. Smith, "The Impact of Online Collaboration Tools on Student Learning Outcomes: A Meta-Analysis," *Educational Technology Research and Development*, vol. 70, no. 2, pp. 741-759, 2023.
- [7] A. Rivera, and B. Martin, "Innovation and Project Management in Higher Education: A Framework for Success," *Journal of Project Management in Education*, vol. 27, no. 3, pp. 100-118, 2024.
- [8] H. Patel, and R. Thompson, "The Role of Centralized Project Management Tools in Enhancing Academic Collaboration," *Educational Technology in Practice*, vol. 34, no. 1, pp. 45-62, 2023.

- [9] M. Lewis, and E. Richards, "Project-Based Learning and Online Tools: A Pathway to Success in Modern Education," *International Review of Education*, vol. 29, no. 4, pp. 203-221, 2024.
- [10] J. Carter, and S. Perez, "Examining the Effectiveness of GitHub Integration in Academic Project Management Platforms," *International Journal of Educational Research*, vol. 66, pp. 135-146, 2023.
- [11] T. Nguyen, and M. Roberts, "Scalable Educational Platforms for Project-Based Learning: Opportunities and Challenges," *Journal of Information Technology Education: Research*, vol. 22, no. 3, pp. 221-245, 2023.
- [12] P. Singh, and N. Gupta, "Collaborative Tools for Higher Education: Enhancing Learning Outcomes Through Technology," *Journal of Educational Innovation and Technology*, vol. 15, no. 1, pp. 78-95, 2023.
- [13] H. Kim, and S. Wang, "Building Robust Collaboration Networks in Project-Based Learning," *Computers in Human Behavior*, vol. 140, pp. 106-114, 2023.
- [14] X. Zhang, and Y. Liu, "Enhancing Project Management with AI-Driven Tools in Higher Education," *Journal of Educational Computing Research*, vol. 38, no. 2, pp. 194-209, 2024.
- [15] A. Ahmad, and M. Bashir, "Understanding the Impact of Technology on Student Collaboration in Higher Education," *Educational Technology & Society**, vol. 26, no. 1, pp. 99-113, 2023.

KS-NH-11111

ORIGINALITY REPORT

7%	6%	1%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to KIET Group of Institutions, Ghaziabad Student Paper	2%
2	www.coursehero.com Internet Source	1%
3	fr.slideshare.net Internet Source	1%
4	Submitted to National Institute of Business Management Sri Lanka Student Paper	<1%
5	liba.edu Internet Source	<1%
6	core.ac.uk Internet Source	<1%
7	Submitted to Middle East College Student Paper	<1%
8	technodocbox.com Internet Source	<1%
9	www.ijraset.com Internet Source	<1%
10	www.ijiet.org Internet Source	<1%
11	Pallavi Tyagi, Neha Zaidi, Balamurgan Balusamy, Celestine Iwendi. "Advanced Learning and Teaching in Higher Education in India: A Policy-technology-capacity Enabled Approach", River Publishers, 2024 Publication	<1%

12	Submitted to University of Bedfordshire Student Paper	<1 %
13	powerknot.com Internet Source	<1 %
14	Nabiyeva, Gulnara Nuridinovna. "Geographic Information Systems (GIS) as a Tool for Sustainable Community Development.", University of California, Davis, 2020 Publication	<1 %
15	Submitted to University of Wolverhampton Student Paper	<1 %
16	www.ijariit.com Internet Source	<1 %
17	ir.juit.ac.in:8080 Internet Source	<1 %
18	jf.jobfetch.com Internet Source	<1 %
19	su-plus.strathmore.edu Internet Source	<1 %
20	www.axeoemr.com Internet Source	<1 %
21	Submitted to Institute of Technology, Nirma University Student Paper	<1 %
22	archive.org Internet Source	<1 %
23	journals.staffs.ac.uk Internet Source	<1 %
24	listens.online Internet Source	<1 %
25	www.inderscience.com Internet Source	<1 %

26	www.msswbc.org	<1 %
	Internet Source	
27	atharvacoe.ac.in	<1 %
	Internet Source	
28	www.researchgate.net	<1 %
	Internet Source	
29	bdta.ufra.edu.br	<1 %
	Internet Source	
30	cahaya-ic.com	<1 %
	Internet Source	
31	hdl.handle.net	<1 %
	Internet Source	
32	intanariestyawuri.wordpress.com	<1 %
	Internet Source	
33	papers.academic-conferences.org	<1 %
	Internet Source	
34	eprints.lancs.ac.uk	<1 %
	Internet Source	

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off

Proof of patent publication (Screenshot of Publication)



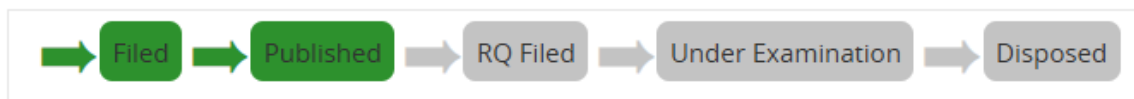
Office of the Controller General of Patents, Designs & Trade Marks
Department for Promotion of Industry and Internal Trade
Ministry of Commerce & Industry,
Government of India



Application Details	
APPLICATION NUMBER	202511003478
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	15/01/2025
APPLICANT NAME	1 . Harsh khatter 2 . Jhalak Saxena 3 . Divyansh Singh 4 . Dr. Kalpna Sagar
TITLE OF INVENTION	EMPLOYING PROJECT MANAGEMENT SOLUTIONS TO DEVELOP A COLLABORATIVE PLATFORM FOR COLLEGE STUDENTS
FIELD OF INVENTION	COMPUTER SCIENCE
E-MAIL (As Per Record)	harshkhatter1988@gmail.com
ADDITIONAL-EMAIL (As Per Record)	
E-MAIL (UPDATED Online)	
PRIORITY DATE	
REQUEST FOR EXAMINATION DATE	--
PUBLICATION DATE (U/S 11A)	31/01/2025

Application Status	
APPLICATION STATUS	Awaiting Request for Examination

			View Documents
--	--	--	--------------------------------



In case of any discrepancy in status, kindly contact ipo-helpdesk@nic.in