ProjectHub – Centralized Platform for Project Management and Collaboration

Synopsis Report submitted in partial fulfillment of the requirement for the degree of
B. E. (Information Technology)

Submitted By

Anuj Gill
Sahil Shangloo
Arkan Khan

Under the Guidance of
Prof. Kanchan Dhuri
Department of Information Technology

Vidyalankar Institute of Technology
Wadala(E), Mumbai 400 037

University of Mumbai

2025-26

CERTIFICATE OF APPROVAL

For

Project Synopsis

This is to Certify that

Anuj Gill
Sahil Shangloo
Arkan Khan

Have successfully carried out Project Synopsis work entitled
ProjectHub – Centralized Platform for Project Management and Collaboration

in partial fulfillment of degree course in Information Technology
As laid down by University of Mumbai during the academic year 2025-26

Under the Guidance of
Prof. Kanchan Dhuri

Prof. Kanchan Dhuri
Guide

Dr. Vidya Chitre
Head of Department

Examiner 1

Examiner 2

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student	Roll No.	Signature
1. Anuj Gill	22101A0016	
2. Sahil Shangloo	22101A0027	
3. Arkan Khan	22101A0049	

Date

Acknowledgements

I would like to express my profound gratitude to my project supervisor **Prof. Kanchan Dhuri,** for their invaluable guidance, encouragement, and continuous support throughout the duration of this project. Their insightful feedback and constructive suggestions were instrumental in shaping the direction and quality of this research.

I would also like to extend my sincere thanks to the faculty members of the **Department of Information Technology** at **Vidyalankar Institute of Technology,** for providing a strong academic foundation and a research-oriented environment that greatly contributed to the successful completion of this work.

Finally, I would like to thank my peers, friends, and family for their patience, motivation, and constant support throughout this journey. Their encouragement kept me focused and driven to bring this project to completion.

Table of Contents

Abstract

Managing academic projects in engineering colleges often involves scattered tools such as WhatsApp, Google Forms, and emails, resulting in lost information, poor tracking, and limited access to past records. Students struggle to find reference projects or suitable mentors, while faculty members face challenges monitoring progress and maintaining transparency. To overcome these inefficiencies, **ProjectHub** provides a unified web platform that digitizes the complete project lifecycle covering mentor allocation, project submissions, progress tracking, and structured reviews. Built using **Next.js**, **NestJS**, and **PostgreSQL**, it ensures scalability, reliability, and smooth collaboration among students, mentors, and administrators.

A key highlight of ProjectHub is the integration of **AI via Google's Gemini API**, which introduces intelligent project recommendations, automated documentation summarization, and smart search features across archived projects. The platform's comprehensive archive enables users to explore previous work by domain, technology, batch, or mentor, complemented by features like upvoting and live demos. By replacing manual, unorganized workflows with a transparent, efficient, and data-driven system, ProjectHub saves faculty time, improves project quality, and preserves institutional knowledge offering a modern, cost-effective solution for academic project management in engineering education.

Chapter 1: Introduction

The academic project lifecycle in engineering institutions involves multiple stages spanning from team formation to final submission. Currently, this entire process is managed through disparate tools including WhatsApp groups, Google Forms, and email communications, leading to fragmented workflows, loss of critical information, and poor tracking mechanisms.

ProjectHub is proposed as a comprehensive web-based platform designed to digitalize and streamline the complete project management workflow. The platform aims to centralize all project-related activities including mentor allocation, topic approval, progress tracking, review management, and archival of past projects. By consolidating these processes into a single unified system, ProjectHub will enhance transparency, improve communication between students and mentors, and create a valuable repository of past projects that can inspire and guide future batches.

The platform incorporates artificial intelligence capabilities to assist students in project ideation, provide automated summaries of project documentation, and enable intelligent search through the project archive. This combination of workflow automation and AI-powered features positions ProjectHub as a transformative solution for academic project management.

Chapter 2: Aim and Objectives

2.1 Aim

The aim of this project is to develop a centralized digital platform that streamlines the entire academic project lifecycle from team formation to final submission while creating an accessible archive of past student projects.

2.2 Objectives

- To digitalize the mentor allocation process and make it transparent and efficient
- To implement a structured workflow for project topic submission, approval, and tracking
- To create a comprehensive review management system for tracking project progress
- To build an archive system that makes past projects accessible to all students
- To integrate AI-powered features for project idea recommendation and documentation summarization
- To provide real-time dashboards for students, mentors, and administrators
- To automate notifications and reminders for important deadlines and reviews
- To enable seamless integration with GitHub repositories for code submission

Chapter 3: Problem Statement

The current system for managing academic projects from the second semester onward including mini, minor, and major projects faces several operational challenges. Communication between students, mentors, and coordinators is scattered across platforms like WhatsApp, email, and Google Forms, often resulting in confusion and loss of important information. Mentor allocation is still handled manually, based on preference collection and faculty availability, making the process slow and non-transparent. There is no centralized platform to monitor project progress, which makes it difficult for mentors and administrators to track multiple teams effectively. Additionally, previous project data is neither well-documented nor easily accessible, preventing students from exploring past work or using it as a reference for ideation. The review and evaluation process is also inefficient, involving multiple manual steps with no structured way to record feedback. Students often struggle to discover good project ideas due to the lack of visibility into past submissions. Project documentation, including reports, code repositories, and deployment links, remains inconsistent and scattered, frequently getting lost after course completion. Overall, the absence of a unified system results in weak accountability and makes it difficult to ensure consistent progress throughout the semester.

Chapter 4: Scope

The proposed system aims to fully digitalize the academic project management workflow for mini, minor, and major projects. It will automate key processes such as mentor allocation, project topic approval, and progress tracking, ensuring efficient coordination between students, mentors, and administrators. The platform will also integrate with GitHub to streamline submissions and enable long-term archival of projects, supported by advanced search and filtering options for easy discovery.

To further enhance the user experience, the system will incorporate AI-driven features such as personalized project idea recommendations and automated documentation summarization. Role-based dashboards will be provided to students, mentors, and coordinators, each equipped with real-time notifications to manage reviews, deadlines, and evaluations effectively. These enhancements will not only improve workflow efficiency but also increase accountability and visibility throughout the project lifecycle.

However, certain features are intentionally excluded from the current scope, such as financial or hardware resource management, integration with recruitment platforms or plagiarism detection tools, and support for mobile applications or live video conferencing. In future phases, the platform may introduce capabilities like a resume builder based on project data, a system to allow continuation of impactful projects by future batches, and an analytics dashboard to provide department-level insights and performance metrics.

Chapter 5: Proposed System

ProjectHub is a web-based platform that aims to make the entire academic project process easier and more organized. It will be built using a three-layer structure. The first layer is the frontend made with Next.js, which students and mentors will use through a clean and responsive interface. The second layer is the backend, built using NestJS, which will handle all the logic and communication between the users and the database. The third layer is the PostgreSQL database, which will safely store all user details, project information, and submitted work.

The system will allow students, mentors, and admins to log in with different roles using secure authentication. Students can create teams, update their details, and apply for mentors. The mentor allocation process will be made simple, where teams can give their preferences, and the system will help in automatically assigning mentors in a fair and transparent way.

ProjectHub will also make it easy to submit project topics and track progress weekly or bi-weekly through milestone updates. Mentors can review the updates and provide feedback directly through the platform. The system will also support scheduled mid and final reviews where all feedback will be saved properly for future reference.

After the project is completed, it will be stored in a digital archive. Future students can search past projects by domain, technology, mentor, or batch. The system will also include AI features like project idea suggestions and automatic progress summaries. To keep everyone updated, notifications for deadlines, reviews, and feedback will be sent through email and the dashboard.

Chapter 6: Methodology

6.1 Development Approach

This project will follow the **Agile development methodology**, where features are built and improved step by step through multiple sprints. The work will be divided across **two semesters**. In Semester 7, the focus will be on building the core system. The first sprint will involve gathering all requirements, designing the overall system, and planning the database structure. Next, user login and role-based access will be developed. Then, team formation and mentor allocation features will be implemented, followed by the project topic submission and approval system. The final sprint of this semester will develop the progress tracking system, allowing mentors to monitor teams and give feedback.

In **Semester 8**, the platform will be enhanced with advanced features. The first sprint will work on creating a project archive with smart search options. Then, AI features using Gemini API will be added to provide project idea suggestions and auto summaries. After that, notifications through email and dashboard alerts will be integrated. Once all major features are complete, the system will go through testing, bug fixing, and performance improvement. The final sprint will focus on proper documentation, deployment, and preparation for presentation.

6.2 Technology Implementation Strategy

The frontend of the system will be built using Next.js and React, ensuring fast performance and a modern user interface. Tailwind CSS will be used for responsive design so the platform works smoothly on all devices. For managing user data on the client side, React Context or Redux will be used. The **backend will be developed using NestJS**, which provides a modular and scalable structure. All communication will be done through secure RESTful APIs with **JWT authentication** to protect user data.

A **PostgreSQL database** will be used to store all structured data such as users, teams, projects, updates, and reviews. For AI features, the **Gemini API** will be used to generate project ideas, summarize reports, and suggest similar past projects through smart prompt engineering. The

project will be deployed using Git and GitHub with CI/CD pipelines. The frontend will be hosted on **Vercel**, and the backend on platforms like **Render or Railway**.

To ensure quality, the system will go through multiple testing stages. Unit tests will check each small function, while integration tests will verify if API endpoints work correctly. User acceptance testing will be done with real students and mentors for feedback. Finally, performance testing will be done to ensure the platform can handle multiple users at the same time.

Chapter 7: Analysis

7.1 Process Model Used for the Project

ProjectHub will follow the **Agile methodology using the Scrum framework** to ensure smooth and flexible project development. The system will be built in **iterations across two semesters**, where each sprint delivers a usable part of the system. This model supports continuous feedback from mentors and students, allowing the system to evolve based on real academic needs. Each sprint will last for two weeks and include sprint planning, daily stand-up meetings, sprint reviews, and retrospectives. This approach ensures continuous progress, regular validation, and adaptability to any changes suggested during pilot testing or academic reviews.

7.2 Feasibility Study

From a **technical perspective**, the system is feasible because it uses modern technologies like Next.js, NestJS, PostgreSQL, and Gemini API — all of which are reliable, well-documented, and supported by strong developer communities. The development team is already familiar with web development, and the required cloud platforms offer generous free tiers.
In terms of **operational feasibility**, ProjectHub directly solves existing problems faced by students, mentors, and coordinators during academic project handling. It fits well into the current academic workflow and can be adopted gradually without disrupting existing processes.
The **economic feasibility** is also strong, as the development cost is very low, limited to around ₹500–1000 for a custom domain. Hosting and tools are available for free or at minimal cost, making it highly budget-friendly. The system will also save significant time and effort in the long run through automation.

Finally, the **schedule feasibility** is practical, with major features planned for Semester 7 and advanced features like AI and automation scheduled for Semester 8. There is adequate buffer time included for testing, refinement, and presentation preparation.

7.3 Cost Analysis

The **development phase** of ProjectHub involves a one-time cost of approximately ₹500–1000 for purchasing a custom domain. All other tools and services such as hosting, APIs, and SSL certificates can be managed using free-tier plans.

During the **deployment phase**, the expected recurring cost per year is around ₹5000–8000 for cloud hosting and database services, along with ₹1000 for domain renewal. SSL certificates will remain free through Let's Encrypt.

In terms of **cost-benefit value**, the investment is highly justified. The system can save nearly **200–300 mentor hours per semester**, reduce miscommunication, eliminate manual work, and improve project quality. It will also create a **permanent, searchable project archive** that will benefit future students and faculty.

Semester 7 (16 weeks)

Week	Module/Activity	Deliverables
1	Initial Project Setup	Codebase initialization, Repository setup, Development environment configuration, Project structure with Next.js and NestJS
2	Groups Sorting and Selection	Team formation module, Group registration system, Team validation (3 members), Team dashboard
3	Choosing Mentor and Mentor Allocation Logic	Mentor allocation workflow design, Preference collection system, Allocation algorithm implementation
4	Implementing Mentor Allocation in Code	Mentor dashboard, Team selection interface, Allocation finalization module, Notification system for allocation
5	Topic Selection Process & Discussion	Topic submission interface, Multiple topic proposal system, Discussion forum/chat feature, Mentor review dashboard

6	Finalizing Topic for Project	Final review workflow, Panel review interface, Topic approval/rejection mechanism, Feedback documentation system
7	Document Process of Project	Architecture documentation module, Workflow visualization tools, Timeline creation interface, Survey form integration, Document repository
8	Actual Start of Project (Students)	Project kickoff module, Milestone setting interface, Resource allocation system, Initial progress tracker
9	Weekly Progress Tracking	Weekly progress submission form, Progress visualization dashboard, Attachment upload system, Timeline tracking
10	Review 1 (Mid-Sem) Logic & Implementation	Mid-semester review scheduling, Review feedback form, Rating system, Review history documentation
11	Final Review and Suggestion	Final review module, Panel member access, Suggestion documentation, Pre-final review checklist
12	Project Archive	Archive module development, Search and filter functionality, Project metadata management, Upvoting system, Similar project recommendation
13-15	Code Implementation & Testing	Integration of all modules, Unit testing, Integration testing, Bug fixes, Performance optimization, User acceptance testing
16	Deployment and Testing	Production deployment, Final testing, Database migration, Documentation finalization, User guide creation

Semester 8 (12 Weeks)

1-2	AI Features Integration	Gemini API integration, Project idea recommender, Auto-summary generator, Keyword extraction system
3-4	AI-Powered Weekly Progress Analysis	AI-based progress evaluation, Automated guidance system, Comparison of previous vs current state, Suggestions and recommendations engine
5-6	Project ArchiveEnhanced Archive Features	Advanced search with filters, Domain/technology-based categorization, Project continuation request system, Batch-wise project browsing
7-8	Notification & Communication System	Email notification service, Reminder system, Real-time notifications, Deadline alerts, Review reminders
9	GitHub Integration Enhancement	Automatic repo link validation, README parsing, Code statistics, Deployment link verification
10	Admin Dashboard & Reports	Analytics dashboard, Download reports, User management, System configuration, Batch management
11	Final Testing & Documentation	Comprehensive system testing, Performance testing, Security audit, Complete documentation, User manuals
12	Project Final Deployment & Viva Preparation	Production deployment, Final review, Presentation preparation, Demo preparation, Viva documentation

Chapter 8: Design

8.1. Data Flow Diagrams (DFD)

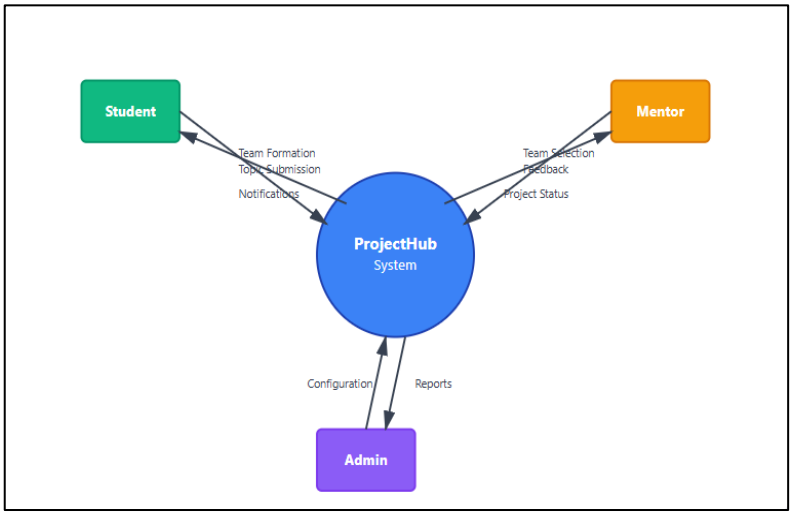

Figure 1: System Interaction Diagram of ProjectHub

It shows how students, mentors, and admins interact with the ProjectHub system through key functions like team formation, feedback, and reporting.

8.2. UML Diagrams

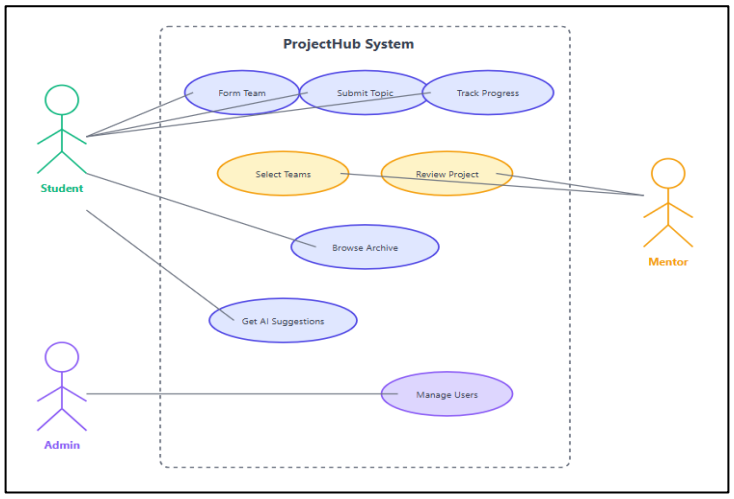

Figure 2: Use Case Diagram of ProjectHub

It depicts the main actions of each user role students, mentors, and admins and their interactions with various modules within the ProjectHub system.

Chapter 9: Hardware and Software Requirements

9.1 Hardware Requirements:

Development Environment:

- Processor: Intel Core i5 or equivalent (minimum)
- RAM: 8 GB (minimum), 16 GB (recommended)
- Storage: 256 GB SSD
- Internet Connection: Broadband with minimum 10 Mbps

Server Requirements (Production):

- Cloud-based infrastructure (Vercel/Render/Railway)
- Auto-scaling capability
- 99.9% uptime SLA
- CDN for static asset delivery

9.2 Software Requirements:

- **Development Tools:** VS Code, Git/GitHub, Postman, Node.js (v18+), PostgreSQL (v14+)
- **Frontend:** Next.js (React + TypeScript), Tailwind CSS
- **Backend:** NestJS (TypeScript, Node.js), JWT authentication
- **Database:** PostgreSQL with Prisma/TypeORM
- **AI Integration:** Google Gemini API
- **Deployment:** Vercel (frontend), Render/Railway (backend), Supabase/Railway (database)
- **Utilities:** SendGrid/Nodemailer (emails), Cloudinary (storage), Custom domain

Chapter 10: Conclusion and Future Work

10.1 Conclusion

ProjectHub streamlines academic project management by replacing scattered tools like WhatsApp, emails, and Google Forms with one unified platform. It simplifies the entire workflow from team formation to final submission ensuring transparency and efficiency.

Students can manage teams, mentor preferences, topics, and progress easily; mentors can review, guide, and track multiple teams effortlessly; and administrators gain full visibility with quick report generation.

The AI integration (Gemini API) enhances productivity by suggesting project ideas, summarizing documents, and recommending similar past projects. The project archive preserves valuable work for future reference, preventing data loss and enabling learning continuity.

Built with Next.js, NestJS, and PostgreSQL, ProjectHub is modern, scalable, and cost-efficient. It minimizes faculty workload, boosts project quality, and can be smoothly implemented without disrupting current systems.

Overall, ProjectHub transforms academic project handling into a seamless, intelligent, and long-term digital ecosystem.

10.2 Future Work

In the future, ProjectHub can be enhanced with AI-driven features like progress analysis, chatbot support, and delay prediction. Collaboration tools such as peer reviews, discussion forums, and mentor-student chats can be added. Integration with college systems, plagiarism tools, and lab databases will further streamline usage, while mobile app support will make project tracking more accessible and efficient.

Chapter 11: References

[1] Next.js - https://nextjs.org/docs

[2] NestJS - https://docs.nestjs.com

[3] PostgreSQL - https://www.postgresql.org/docs/

[4] AWS Cloud Hosting - https://aws.amazon.com/getting-started/

[5] Google AI / Gemini API - https://ai.google.dev

[6] Resend (Email API) - https://resend.com/docs