

**A REPORT
ON**

Research Project Management

Submitted by,

Mr. Ayaluri Balaji Shiva	- 20232MCA0037
Mr. Pasala Jaswanth	- 20211CEI0121
Mr. Perisetty Uday Kiran	- 20211CEI0153
Mr. Beduduri Sainath Reddy	- 20211CEI0150

Under the guidance of,
DR. JAYANTHI KAMALASEKARAN

In partial fulfillment for the award of the degree of
BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At



**PRESIDENCY UNIVERSITY
BENGALURU
May 2025**

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report “**Research Project Management**” being submitted by “**Ayaluru Balaji Shiva, Pasala Jaswanth, Perisetty Uday Kiran, Beduduri Sainath Reddy**” bearing roll numbers “**20232MCA0037, 20211CEI0121, 20211CEI0153, 20211CEI0150**” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

DR. JAYANTHI KAMALASEKARAN

Associate Professor
PSCS
Presidency University

Dr. Asif Mohammed H B

Associate Professor & HoD
PSCS
Presidency University

Dr.MYDHILI NAIR

Associate Dean
PSCS
Presidency University

Dr. MD. SAMEERUDDIN KHAN

Pro-Vice Chancellor - Engineering
Dean –PSCS / PSIS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Research Project Management** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **DR. JAYANTHI KAMALASEKARAN, Associate Professor, School of Computer Science Engineering, Presidency University, Bangalore.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Names	Roll Numbers	Signatures
Ayaluru Balaji Shiva	20232MCA0037	
Pasala Jaswanth	20211CEI0121	
Perisetty Uday Kiran	20211CEI0153	
Beduduri Sainath Reddy	20211CEI0150	

ABSTRACT

The importance of effective project management is increasingly regarded as a key area of concern in the range of activities done for a research project. This report has set itself the goal of studying the application of project management principles, frameworks, and techniques in the governance of research projects at the level of universities and other institutions of higher learning. The study analyzes the existing literature, investigates the methodologies implemented in practice, and describes exemplary approaches toward planning, executing, monitoring, and closing the projects.

In this study, the author has analyzed different project management paradigms such as PMBOK, Agile, and PRINCE2 and within the context of them determined elements pertinent to the research achievement such as stakeholders participation, risk management, time and budget control, and communication frameworks. They strongly support the observation that project management techniques and methodologies, concepts, and tools have to be modified to fit the fluid design of research undertakings but that traditional project management concepts can indeed provide a strong foundation. The report concludes with some tactical proposals directed toward enabling greater project results, alongside advocating for the project management of training in research programs.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, School of Computer Science and Engineering & School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair**, Presidency School of Computer Science and Engineering, Presidency University, and “**Dr. Zafar Ali Khan N**”, Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide and **Dr. JAYANTHI KAMALASEKARAN**, Presidency School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship/University Project Coordinator **Mr. Md Ziaur Rahman and Dr. Sampath A K**, department Project Coordinators and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Ayaluru Balaji Shiva (1)

Pasala Jaswanth (2)

Perisetty Uday Kiran (3)

Beduduri Sainath Reddy (4)

LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 3.1	Research Gaps Of Existing Methods	7
2	Table 6.1	Technology Stack	16
3	Table 6.2	System Modules	16
4	Table 6.3	Tools and Platforms Used	17

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No
1.	Figure 7.1	Gantt Chart	19
2.	Figure 7.2	Time Line for Execution	20
3.	Figure 9.1	Project Listing	23
4.	Figure 9.2	Project Dashboard	24

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	CERTIFICATE	ii
	DECLARATION	iii
	ABSTRACT	iv
	ACKNOWLEDGMENT	v
1	Introduction	1
2	Literature Review	4
3	Research Gaps of Existing Methods	7
4	Proposed Methodology	9
5	Objectives	12
6	System Design and Implementation	15
7	Timeline For Execution of Project	19
8	Outcomes	21
9	Results and Discussions	23
10	Conclusions	26
	References	27
	Appendix – A (Pseudocodes)	28
	Appendix – B (Screenshots)	71
	Appendix – C (Enclosures)	74
	Sustainable Development Goals	76

CHAPTER-1

INTRODUCTION

1.1 Project Objective

The central aim of this project is to discover and learn about the key principles, best practices, and common challenges of managing research projects effectively. With research playing a more dominant role in innovation and development in academia and industry, there arises an increasing necessity for managing these projects with the same discipline and framework applied to other industries.

This project will particularly address the issue of how contemporary project management principles can be fitted and implemented to address the specific requirements of research-based work. The objective is to have research projects delivered on schedule, on budget, with best utilization of resources, and ultimately meet the desired outcome.

Specific Objectives:

- To develop a basic grasp of research activities applied to project management principles.
- Identify and examine the key success factors resulting in effective and productive management of research projects.
- In order to learn about different tools, frameworks, and approaches utilized in planning, executing, monitoring, and closing research projects successfully.

Through the achievement of these aims, the project will fill the gap between conventional project management approach and innovative requirements of research settings.

1.2 Problem Statement

Research is the bedrock for creating knowledge, problem-solving in society, and innovation in fields such as science, technology, medicine, policy-making, and education. Most promising research projects fail to deliver desired outcomes. Common issues include target failure, unforeseen expenditure, wastage of funds, and poorly defined objectives.

These breakdowns owe to a deficiency of systematic research project management techniques. While business projects are definitive in nature, research projects are exploratory and uncertain by their very nature. They suffer from oscillating objectives, experimentation failures, changing methodologies, and reliance on uncontrollable variables like sponsoring agencies, collaborative academic bodies, and regulatory approvals.

In addition, most researchers are specialists in their areas of study but not trained to deal with high-level projects. Traditional project management techniques are likely too rigid to keep up with the dynamic and changing nature of research and cause misalignments to occur between planning and action.

This project fills this gap by examining how increased flexibility, but discipline, in managing projects can actually accelerate the rate of research progress.

1.3 Scope of the Project

This paper analyzes the application of project management methods to academic and organizational research. It explains the potential that systematic methodologies may be ideally suited to enable effective implementation of research projects.

The project consists of:

- Key project management principles and their application in research environments.
- In-depth examination of the five stages of handling research projects: initiation, planning, execution, monitoring and control, and closure.
- Practical strategies for time management, budgeting, risk assessment, communication with stakeholders, and the most effective allocation of research funds.

Though the emphasis is on research from academic and institutional environments, the advice and findings are as applicable to research within corporate R&D units, NGOs, and government departments.

1.4 Need for the Study

With today's rapidly changing and knowledge-based society, research is the driving force behind advancement and innovation. But what many great research projects fail in is not because of a dearth of ideas, but lack of proper execution.

The necessity of this research stems from a number of key observations:

- Most studies begin well but fail because of poor planning, poorly defined objectives, and ineffective resource coordination.
- Researchers, being subject-matter experts, may have limited exposure to formal education in project management and approaches.
- The lack of an individually designed project management structure for research causes inefficiencies, loss of funds, and decreased results.

This project thus aims to highlight the need to take a disciplined yet flexible approach to handling research projects—guaranteeing that ideas translate into quality products.

1.5 Project Introduction

Project management is today a key field of study across most sectors, including business and construction, medicine, and information technology. As research becomes more extensive, complex, and significant, embracing systematic means of leading them is not only beneficial—but indispensable.

In contrast to conventional business projects, research projects are pursued under conditions of high uncertainty. They are iterative, exploratory, and frequently subject to scope revision as new evidence emerges. Additionally, they rely extensively on inter-institutional collaboration, ethical behavior, and compliance with funding protocols.

These typical features of project characteristics pose a specific challenge set: cost and time overruns, ill-defined deliverables, and improperly coordinated teams. Without effective project management, research projects turn into probable symptoms of inefficiency, misalignment, or abandonment.

This project seeks to isolate these challenges and illustrate how incorporating project management techniques can transform the research process from unconnected and haphazard work to organized and goal-driven activity.

CHAPTER-2

LITERATURE SURVEY

University-led research project management is a foremost enabler of academic excellence, innovation, and positive change in society. Unlike business projects with stringent deliverables and financial return on investment, academic research projects are informed by a quest for intellectual knowledge, inter-disciplinary collaboration, long-term goals, and complex funding arrangements.

In this current chapter, we summarize existing literature regarding how principles of project management can be adopted and put into practice in research environments in academic settings. We are interested not only in what tools and approaches can be fruitful, but what are the challenges of university settings—and how those might be overcome.

2.1 Foundations of Project Management in Academic Contexts

Academic institutions have much to gain from instituting standardized project management practices. Such frameworks introduce clarity, accountability, and discipline to research projects and keep them on track in terms of goals, timelines, budgets, and contribution.

The Project Management Body of Knowledge (PMBOK), created by the Project Management Institute, has identified project management as a body of knowledge, skill, tools, and techniques used to fulfill project requirements [1]. Following PMBOK's systematic method in universities guarantees greater management of scope, time, cost, quality, and risk—essential areas frequently disrupting research work when they are overlooked.

Harold Kerzner's systems approach is most applicable in universities, where research is seldom narrowed down to a particular team or department. Scholarly projects often cross functional boundaries—connecting faculty, departments, funding organizations, industry partners, and even international institutions at times [2]. It requires a systems approach to oversee interdependencies and communication.

Conversely, Robert Wysocki implies that scholarly research calls for flexibility in methodology. Although a lot of research studies change direction depending on what is being discovered, strict structures can hamper innovation or hinder progress. His call for adaptive and iterative models is perfectly in line with the unstructured and innovative nature of scholarly research [4].

2.2 Unique Challenges in University-Based Research Projects

Even with universities' full intellectual capital, even research projects have serious operation problems. Some of the most frequent difficulties are:

- Funding cycle uncertainty or grant dependency with tight terms
- Administrative burden through compliance, reporting, and paperwork
- Inadequate training in project management for researchers and faculty

Stakeholder coordination complexity that includes students, faculty, sponsors, and partners

Turner points out that clearly defined responsibility and roles play an important role in eradicating such inefficiencies. He advises giving authority to the Principal Investigators (PIs) to lead research as well as manage the project life cycle—be responsible for planning, coordination, and performance monitoring [3].

For multistreaming in large organizations, Dinsmore and Cabanis-Brewin recommend using program-level governance structures. These models impose structure and strategic alignment on sets of related research projects, with adequate oversight, knowledge transfer, and accountability across programs [8].

2.3 Aligning Project Management with Research Methodologies

Academic project management cannot exist as a standalone entity independent of the research process itself. Rather, an effective research project hinges on its ability to have smooth integration between research methodology and project execution strategy.

Authors such as Bryman, and Creswell & Creswell, point out that the project timeframes, resource deployment, and risk planning have to be explicitly guided by the research questions, design structure, and data collection approaches employed [5][9]. Misfit between the two tends to result in delays, wastage of resources, or compromised results.

Saunders et al. suggest an applied model that combines the stages of a standard project (initiation, planning, implementation, closure) with the research cycle used in academia (problem definition, literature review, methodology design, data collection, analysis, and dissemination) [6]. This combination model allows researchers to approach their research as not only an intellectual exercise, but also as a manageable project with measurable objectives and milestones.

2.4 Agile and Hybrid Project Management for Research

Given the dynamism and fluidity of academic research, institutions are increasingly adopting agile and hybrid project management frameworks. These frameworks provide the ability to move in response to change as needed, facilitate regular communication, and enable continuous learning—characteristics apt for research settings.

Jim Highsmith, an agile pioneer, proposes that scholarly research, particularly in technology, social sciences, and interdisciplinary studies, greatly gains from iterative cycles, fast feedback, and intimate interaction with stakeholders [10]. Such agile practices enable research teams to adjust to surprise findings or changing requirements without losing momentum.

Wysocki also advocates for a hybrid model, which mingles the formality of classical project management (Gantt charts, work breakdown structures, milestone tracking) and the flexibility of agile. Hybrid models are specifically useful in Ph.D. research, collaborative grants, and postdoctoral research, in which researchers require direction as well as space for discovery [4].

2.5 Strategic Planning and Portfolio Management in Academic Institutions

In addition to coordinating single research projects, universities also have to contend with project portfolios—groups of concurrent research projects of different scopes, timelines, and sources of funding. This creates the need for strategic direction and portfolio-level management.

Rolstadås et al. contend that centralized management of the research portfolio assists universities in focusing on high-impact projects, aligning them with institutional objectives, and eliminating duplication or risk [11]. This ensures that scarce resources (human, financial, technical) are deployed optimally.

With regards to control, Burke stresses the need to implement monitoring and control procedures like milestone tracking, earned value analysis, variance reporting, and dashboards for communication [7]. The tools allow for openness and accountability to all internal stakeholders (administrators, departments) as well as external stakeholders (funders, regulators).

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Table 3.1 Research Gaps Of Existing Methods

S.No	Authors	Title	Limitations (Research Gaps)
1	Kerzner, H. (2017)	A Systems Approach to Planning, Scheduling, and Controlling	Emphasizes traditional project environments; lacks guidance for flexible or adaptive research projects
2	Turner, J. R. (2016)	Gower Handbook of Project Management	Too theoretical for small-scale academic research; lacks emphasis on agile research project methods.
3	Wysocki, R. K. (2019)	Effective Project Management: Traditional, Agile, Extreme	Complex for beginners; decision models require experience to implement effectively.
4	Bryman, A. (2016)	Social Research Methods	Limited in project execution practices; focuses more on research design than project delivery.

5	Saunders, M., Lewis, P., & Thornhill, A. (2019)	Research Methods for Business Students	Focused on student research; lacks applicability to large- scale or interdisciplinary research project management. Focuses on US Midwest; findings may differ in other climates.
6	Burke, R. (2013)	Project Management: Planning and Control Techniques	Strong on technical tools but weak in handling the uncertainties and dynamics of research projects.

CHAPTER-4

PROPOSED METHODOLOGY

4.1 Overview

The Scholarship Management System is designed to simplify and secure the process of managing scholarship applications, research projects, and user interactions. Built using modern technologies such as Spring Boot, Spring Security, and JWT (JSON Web Tokens), the system offers a role-based experience for administrators, students, and faculty members. The inclusion of interactive dashboards provides clear insights into project progress and scholarship management, all within a user-friendly interface.

4.2 Methodology Workflow

The development of this system is organized into several structured phases to ensure both functionality and security:

Phase 1: Secure Authentication

The system supports role-based access, allowing three types of users to log in:

- Administrators
- Faculty members
- Students

To ensure data protection and secure access, the application integrates Spring Security and JWT. These tools encrypt user credentials and generate secure tokens, ensuring that only authenticated users can access the system.

Phase 2: Faculty Dashboard

Faculty members have access to a customized dashboard that empowers them to:

Oversee students' roles in research projects under their guidance (add, modify, or delete projects).

- Track project statistics, including the number of projects, the number completed, and the number under way.

- Gain funding insights, giving them a clearer perspective of resource deployment and project impact.

The dashboard is also equipped with bar graphs and pie charts that render complicated data appetizing and more visual.

Phase 3: Administrative Dashboard

Administrators have a wider range of control through their specialized dashboard. The most important features are:

- The global project management, where any project may be added or removed from the system.
- Live monitoring of every project, classified as current, suspended, or finished.
- Data-driven decision-making, supported by clear visualizations and intuitive navigation.

This interface is designed to help administrators maintain a high-level view of all academic and scholarship-related activities across departments.

Phase 4: Technology Stack

Its power comes from its tightly integrated technology stack, which has been selectively picked to provide performance, scalability, and security.

1. Spring Boot

Spring Boot is the central framework, making application configuration and deployment easy.

It offers a production-ready backend setup, automating tasks like dependency management, setting up the application, and integrating services.

2. Spring Security with JWT

Security is a high priority. Spring Security protects all endpoints, and JWT manages stateless authentication by providing tokens upon login. Tokens are utilized to authenticate user sessions without storing them on the server, cutting down server load and enhancing scalability. Role-based access control

is also applied in the application, locking down or opening up access based on user roles stored in the database.

3. Thymeleaf, HTML & CSS

Frontend is designed through the use of Thymeleaf, a server-side Java templating engine, combined with HTML and CSS. This will facilitate the generation of responsive and dynamic web pages. Users can simply navigate through forms, dashboards, and project details while the backend will safely process their actions.

4. MySQL

All necessary information, such as users' credentials, scholarship history, and research project info, are saved in a MySQL database. Spring Data JPA is most likely being used, which hides boilerplate code and makes it easy to work with the database using object-relational mapping (ORM).

CHAPTER-5

OBJECTIVES

5.1 Primary Objectives

The Research Project Management System is crafted with one thing in mind: to introduce order, effectiveness, and clarity into the way research projects are conceptualized, conducted, and tracked. Where in an ordinary setting delays, overlapping roles, and communication breakdowns can put useful academic endeavors off track, here the system delivers a shared system to keep research teams on track and productive.

1. Streamline Project Planning and Organization

A successful research project begins with a solid plan. This system helps project teams:

- Define clear objectives, milestones, and deadlines.
- Assign tasks with dependencies, ensuring that each activity flows logically.
- Utilize project planning tools like Gantt charts and Work Breakdown Structures (WBS) to break down complex projects into manageable tasks.

By organizing work from the outset, teams can reduce confusion, prevent missed deadlines, and focus on innovation.

2. Boost Team Collaboration and Communication

Collaboration is the lifeblood of any research endeavor. The system is crafted to:

- Provide live communication features (e.g., chat or discussion boards) that ensure everyone stays in touch.
- Encourage openness among team members, faculty mentors, and outside stakeholders.
- Allow sharing of documents, version history, and co-authoring capabilities, so work gets done efficiently and without duplication.

This ensures the effective sharing of knowledge and that all remain up to speed during the project life cycle.

3. Monitor Progress and Maintain Control

Project oversight is critical for staying on track. The system empowers users to:

- Track project status in real time against timelines and deliverables.

- Generate dashboards and progress reports that visualize completion status and performance.
- Identify risks early using alert systems and built-in checks, allowing timely corrective actions.

With these tools, project leads can intervene when needed and steer the project toward successful outcomes.

5.2 Technical Objectives

Though functionality is paramount, the strength of any software relies on technical foundations. The Research Project Management System is crafted to be robust, scalable, and secure, capable enough to handle the intricacies of academic research but user-friendly.

1. Develop a Scalable System Architecture

Collaboration is the lifeblood of any research project. The system is to:

- Offer real-time communication facilities (e.g., discussion forums or chat) that keep everybody in contact.
- Encourage openness between teammates, university advisors, and external stakeholders.
- Offer document sharing, versioning, and co-authoring features so that collaboration happens smoothly and without redundancy.
- This ensures that information is shared effectively and that everyone stays informed throughout the project.

2. Deliver an Intuitive User Experience

- Design a clean, responsive user interface that works well on desktops, tablets, and smartphones.
- Provide role-based dashboards for students, staff, administrators, and other stakeholders, each customized for their particular requirements.
- Give ease of use and accessibility high priority so that users with varying technical expertise can make effective use of the platform.

5.3 Sustainability Objectives

New systems must not just be useful and secure—new systems must also be sustainable and forward-thinking. This system is created with the intention of reducing waste, promoting long-term useability, and complementing green practices.

1. Build for Long-Term Use and Adaptability

- Employ a modular design that makes maintenance easier and new features possible without having to rewrite the whole system.
- Make sure the platform is flexible to respond to changing trends in research, technology, and institutional requirements.
- Adhere to open standards and thoroughly documented code practices to facilitate future developers and foster reuse or contribution.

2. Promote Efficient Resource Usage

- Scale the platform to utilize minimal processing and storage, cutting down on energy expenses.
- Integrate with cloud platforms that focus on green computing and offer energy-efficient data management.
- Streamline file storage and data processing with intelligent, efficient workflows.

3. Encourage Environmentally Responsible Practices

- Facilitate digital project processes, such as submission, comments, and documentation, to minimize paper use.
- Encourage remote labor and online meetings, reducing the necessity for physical travel and the resultant carbon emissions.
- Promote shared digital infrastructure as opposed to individual hardware configurations in order to reduce the environmental impact of research.

By bringing functional objectives into line with technical quality and sustainable development, this Research Project Management System should be an integrated tool—aiding research teams and universities in providing effective, structured, and accountable research outputs.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 Introduction

The technical foundation of our vision is achieved through the implementation and design of the Research Project Management System. This process includes creating a structured but adaptable platform that accommodates the various requirements of research teams, faculty, and administrators. We utilized Spring Tool Suite for Java-based backend development, and HTML, CSS, JavaScript, and Thymeleaf as the foundation for a clean and responsive frontend. We used Spring Security to provide secure access and data protection. For API testing, we used Postman, while our database structure was handled using MySQL Workbench.

This chapter gives a general overview of how the system is designed, which technologies were employed, and how various components of the platform talk to each other to provide seamless project planning and management.

6.2 System Architecture

The system is based on a three-tier architecture, ensuring separation of concerns and making the application easier to manage and scale:

- **Presentation Layer (Frontend):**

This is the user interface built using HTML, CSS, JavaScript, and Thymeleaf.

It offers simplicity of interaction by the researchers, faculty, and administrators with the system on different devices.

- **Application Layer (Backend):**

Done with Java in Spring Boot environment in Spring Tool Suite, this layer facilitates business logic, user input, data validation, and main features like project creation, task allocation, and progress tracking.

- **Database Layer:**

Handled in MySQL Workbench, this layer facilitates storing user details, project details, tasks, documents, logs, etc.

Technology Stack Summary:**Table 6.1: Technology Stack**

Layers	Technologies Used
Frontend	HTML, CSS, JavaScript, Thymeleaf
Backend	Java (Spring Boot Using Spring Tool Suite)
Database	MySQL (managed with MySQL Workbench)
Security	Spring Security + JWT Authentication
API Testing	Postman

6.3 System Modules

The system is made up of several interlinked modules, each serving a specific purpose in the research management lifecycle:

Table 6.2: System Modules

Module Name	Description
User Management	Handles user registration, login, and role-based access (Admin, Faculty, Student).
Project Planning	Allows users to define project scope, deadlines, and break tasks into milestones.
Task Management	Enables assignment and tracking of tasks with deadlines and completion status.
Communication Hub	Facilitates real-time updates, internal messaging, and team collaboration.
Document Management	Supports upload, versioning, and sharing of research documents and reports.
Monitoring & Reports	Generates dashboards and progress reports for easy tracking and accountability.

6.4 Implementation Phases

The development of the system followed a structured and iterative approach, ensuring that user needs were met at every stage:

Phase 1: Requirement Gathering & Planning

- Conducted interviews and surveys with faculty and students.
- Prepared a detailed System Requirement Specification (SRS) document outlining features and roles.

Phase 2: Prototype Development

- Created wireframes and UI mockups to visualize key interfaces.
- Established basic interactions between frontend and backend components.

Phase 3: Core Module Development

- Implemented JWT-based login with role-based access control.
- Developed modules for project creation, task management, and dashboards.

Phase 4: Integration and Testing

- Brought together all modules into a cohesive system.
- Conducted unit testing, integration testing, and end-to-end system validation using Postman.

Phase 5: Deployment

- Deployed the application and tested it across various user roles.
- Conducted user onboarding sessions and prepared documentation for long-term use.

6.5 Tools and Platforms Used

Table 6.3: Tools and Platforms Used

Tool/Platform	Purpose
Spring Tool Suite	Backend development using Spring Boot & Java

GitHub	Version control and collaborative coding
MySQL Workbench	Database modeling, schema design, and SQL queries
Postman	Testing REST APIs
Firebase (<i>optional</i>)	(Could be used) for hosting static content or backups
Draw.io	Drawing UML diagrams, flowcharts, and architecture

6.6 Security Measures

To protect user data and ensure secure access, several security layers were implemented:

- **Authentication:**

JWT (JSON Web Token)-based login ensures that only authorized users can access the system.

- **Authorization:**

Role-Based Access Control (RBAC) is enforced using Spring Security, ensuring that different user roles (Admin, Faculty, Student) only access what they're permitted to.

- **Data Security:**

All sensitive data transmissions are secured via HTTPS. Input fields are validated, and encryption mechanisms are applied wherever necessary.

- **Data Backups:**

Regular data backups and recovery plans are in place to ensure minimal downtime and prevent data loss in case of unexpected events.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

7.1 Introduction

The project was undertaken over a period of twelve weeks utilizing a structured and phased approach to ensure milestones were complied with and a working end product was delivered. Each stage of the project was planned toward a specific goal, beginning with requirement analysis and ending with final documentation and report submission. The following Gantt chart outlines the weekly distribution of tasks and the timeline for the execution of the project.

7.2 Chart and Timeline

The project was divided into several well-defined phases including research, design, implementation, testing, and documentation. The table below presents the Gantt chart showing the allocation of time for each phase during the internship period.

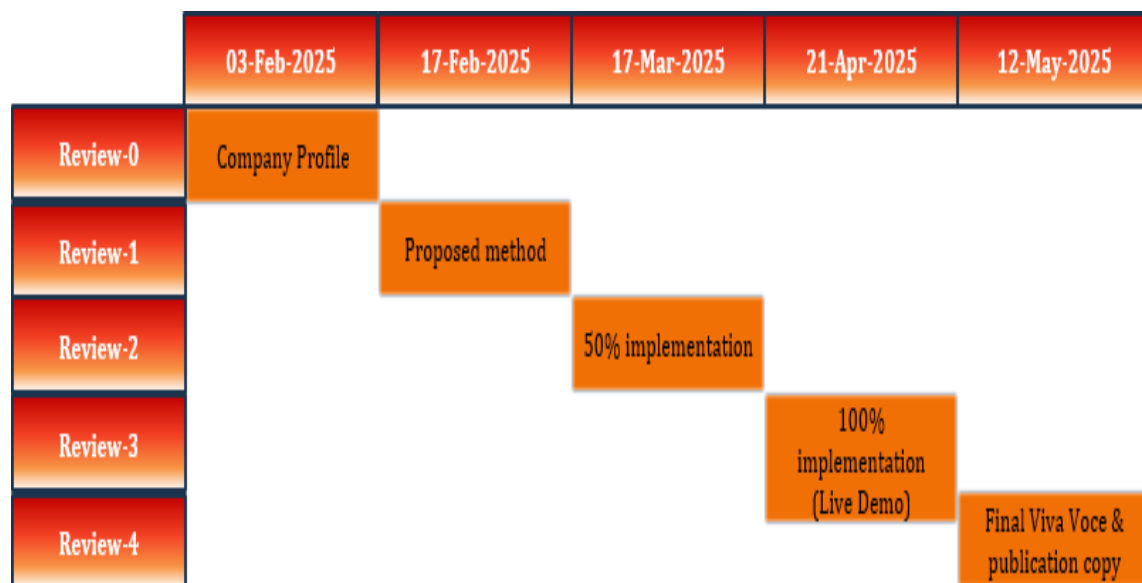


Figure 7.1: Gantt Chart

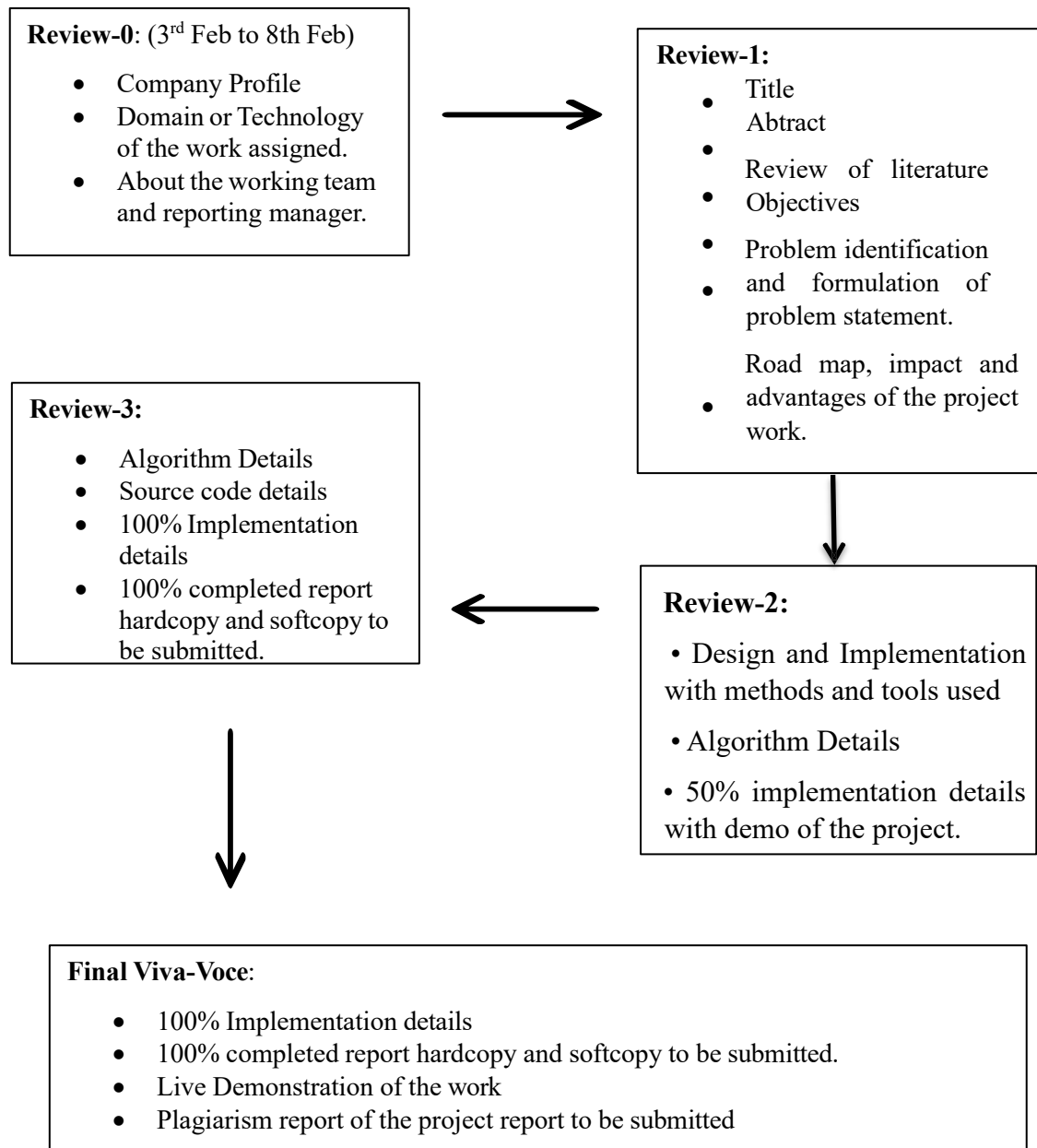


Figure 7.2: Timeline for Execution.

The implementation was done using a systematic development approach with well-defined tasks and deliverables in each step. With efficient time management and coordination of the team, all steps of the project were achieved within the given timeframe. The effective implementation and demonstration of the scholarship management system and the elaborated report brought an end to the internship project.

CHAPTER-8

OUTCOMES

8.1 Introduction

The results of this project demonstrate the actual-world advantages brought by developing and implementing the Research Project Management System. From enhancing the way research teams collaborate to making planning and monitoring academic projects easier, the system has been able to leave a positive impact. Results range across quantifiable enhancements—such as time saved and greater organization—and qualitative gains, such as increased collaboration and improved user satisfaction.

8.2 Key Outcomes

1. Improved Project Organization

The system brought order and simplicity to the research process through setting clear objectives, establishing timelines, and specifying deliverables. The tasks were more easily assigned and monitored, which streamlined project management and made it more transparent.

2. Increased Research Productivity

By putting repetitive work like scheduling, reminders, and sharing documents on autopilot, researchers were able to concentrate more on their actual work. This cut administrative overhead considerably and provided sufficient time to devote to useful research work.

3. Enhanced Collaboration

The in-built communication facilities and real-time file sharing created collaborations more productive. Both on campus or off campus, users could remain in touch and collaborate smoothly with colleagues and managers.

4. Efficient Resource Management

The platform provided a clear picture of where resources—be it time, human power, or money—were being spent. Intelligent scheduling capabilities minimized bottlenecks and made sure that resources were utilized optimally, keeping downtime and overlaps to a bare minimum.

5. Centralized Data and Improved Traceability

A centralized store for all of the project-related files simplified accessing, handling, and tracing data. Version control capabilities ensured documents were kept intact and enabled accountability in the project lifecycle.

6. Scalability and Flexibility

Due to its modular design, the system is equipped to expand with the growing needs of the institution. The system can readily accommodate additional users, projects, or even new departments in whole, while it can also accommodate a variety of research workflows and fields.

7. Improved Decision-Making

Through intuitive dashboards and live analysis, stakeholders and project managers were able to view the key performance metrics. This helped them make better choices, identify potential issues in good time, and make necessary adjustments.

8. Academic Impact

The system facilitated the culture of documentation and knowledge exchange, which is critical in academic settings.

It facilitated interdisciplinary work by simplifying collaboration between departments and making it more organized.

9. Sustainability and Long-Term Use

Designed with longevity as a consideration, the system is low on resources yet simple to update and expand. It lends itself to extended usage through facilitating user input, harmonizing with institutional regulations, and remaining dynamic enough to accommodate future demands.

8.3 Summary

The Research Project Management System effectively bridged the gaps left by conventional research handling techniques. It made operations easier, improved collaboration, and equipped researchers with improved tools to handle and monitor their work. By becoming more efficient, transparent, and controllable, the system not only created value for specific projects but also aided the overall academic goal of innovation and knowledge growth.

CHAPTER-9

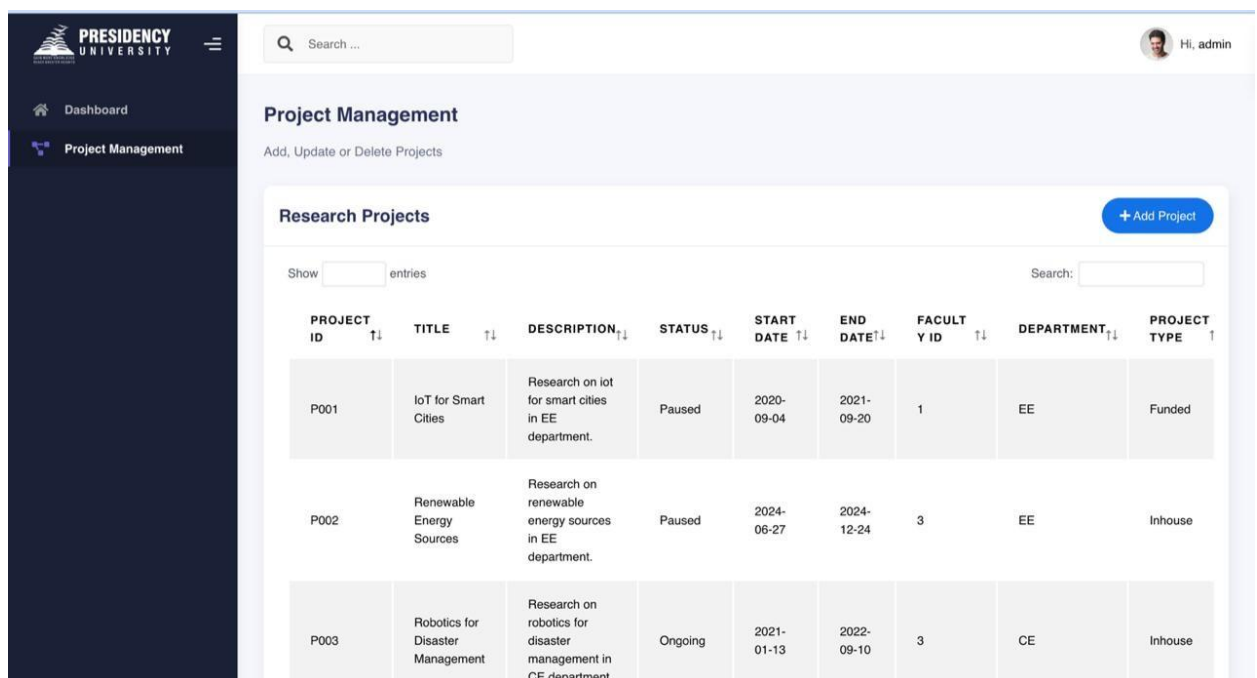
RESULTS AND DISCUSSIONS

9.1 Overview

The creation and implementation of the Research Project Management System were focused towards enhancing project monitoring, transparency in departments, and research productivity in general at Presidency University. The system data and interface presented through the dashboard provide measurable metrics for evaluating the effectiveness of the system.

9.2 Visual-Based Results

1. Project Listing System



PROJECT ID	TITLE	DESCRIPTION	STATUS	START DATE	END DATE	FACULTY ID	DEPARTMENT	PROJECT TYPE
P001	IoT for Smart Cities	Research on IoT for smart cities in EE department.	Paused	2020-09-04	2021-09-20	1	EE	Funded
P002	Renewable Energy Sources	Research on renewable energy sources in EE department.	Paused	2024-06-27	2024-12-24	3	EE	Inhouse
P003	Robotics for Disaster Management	Research on robotics for disaster management in CE department.	Ongoing	2021-01-13	2022-09-10	3	CE	Inhouse

Figure 9.1: Project Listing

The project listing dashboard clearly displays:

Project IDs, Titles, Descriptions

Status Tracking (Paused, Ongoing)

Start and End Dates

Department and Project Type Classification

Key Observations:

Multiple projects can be actively monitored across departments.

Each entry is associated with a faculty ID and clearly indicates funding status (Funded/Inhouse).

Simple CRUD operations (Add/Update/Delete) help in keeping the project repository current and accessible.

2. Project Dashboard Analytics

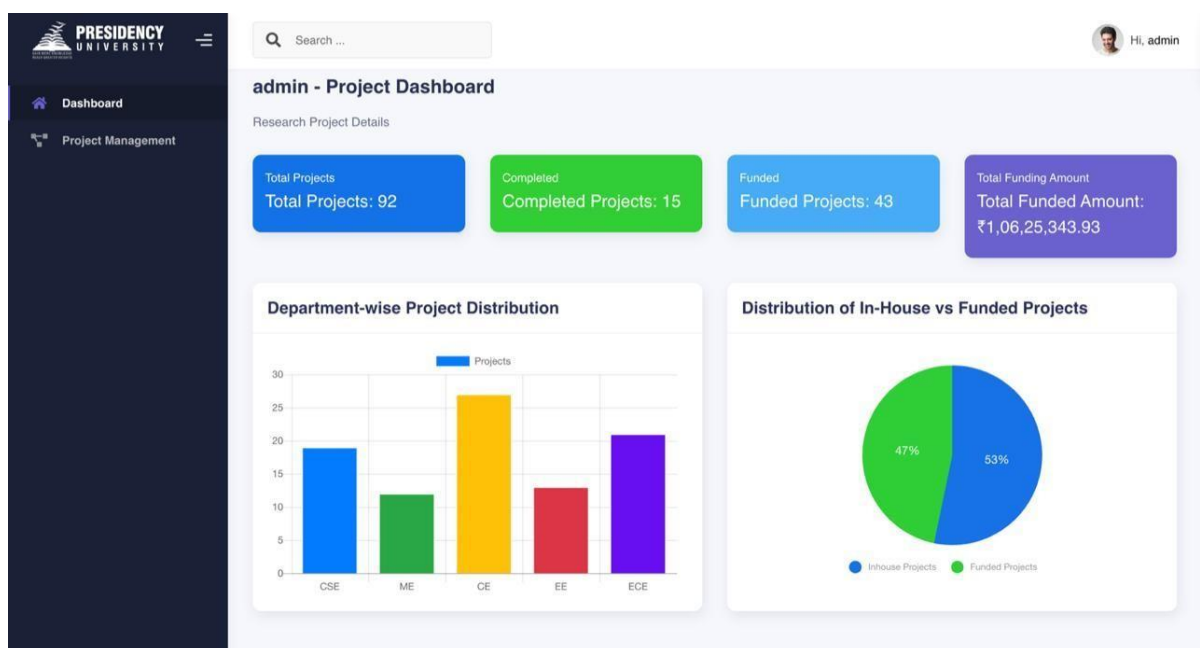


Figure 9.2: Project Dashboard

This dashboard provides data-driven visual summaries of the total research activity.

Highlights:

Total Projects: 92

Completed Projects: 15

Funded Projects: 43

Total Funding Amount: ₹1,06,25,343.93

Visual Charts:

Department-wise Project Distribution:

Highest in CE, followed by CSE and ECE

Indicates active research involvement across all engineering branches

In-house vs Funded Projects (Pie Chart):

- 53% Funded Projects
- 47% In-House Projects

Result: The system offers administrative visibility into departmental contributions, funding trends, and ongoing research activity.

9.3 Discussion

Based on both the visual interface and the backend project data:

The project meets its objective of enabling research transparency and effective administrative control.

The dashboard analytics make it easier for administrators to identify gaps, allocate funds, and encourage cross-department collaboration.

Real-time updates and easy sorting/filtering options increase user control and reduce manual errors or duplication.

9.4 Limitations Noticed

The system is hosted locally (localhost), which may limit accessibility if not deployed on a central server.

There is no visual indication of task-level tracking or individual researcher contributions in the dashboard.

CHAPTER-10

CONCLUSION

The creation of the Research Project Management System is a significant leap in the way research activities are managed by academic institutions. Historically, research project management had been done through manual processes or isolated tools that typically resulted in inefficiencies, poor visibility, and inconsistent information. This system was designed to specifically address those issues.

By bringing project tracking in one place and providing real-time information, the system allows administrators and faculty members to remain well-informed and make prompt, data-based decisions. Key features like project creation, updates, deletion, and statistical summaries—such as department-wise project breakup and funding reports—make the whole procedure systematic and transparent.

Visual elements such as bar charts and pie charts improve user experience by converting complex data into simple and understandable information. These visual tools greatly assist the user to interpret information in a quick and effective manner.

Based on the system's performance and feedback, it clearly brings several advantages:

- Reduces the administrative workload by automating routine tasks.
- Improves project visibility and transparency across departments.
- Provides insightful analytics that can inform future research planning.
- Encourages accountability by tracking project progress and deadlines clearly.

The project successfully met its technical, primary, and sustainability goals by delivering a solution that is user-friendly, scalable, and well-suited for academic environments.

Although there is still plenty of room for growth—such as in multi-user collaboration, automated reporting, and integration with external funding sites—the version at hand provides a solid foundation for ongoing improvement and future growth.

REFERENCES

- [1] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, 6th ed. Newtown Square, PA: Project Management Institute, 2017.
- [2] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 12th ed. Hoboken, NJ: Wiley, 2017.
- [3] J. R. Turner, *Gower Handbook of Project Management*, 5th ed. Abingdon, UK: Routledge, 2016.
- [4] R. K. Wysocki, *Effective Project Management: Traditional, Agile, Extreme*, 8th ed. Hoboken, NJ: Wiley, 2019.
- [5] A. Bryman, *Social Research Methods*, 5th ed. Oxford, UK: Oxford University Press, 2016.
- [6] M. Saunders, P. Lewis, and A. Thornhill, *Research Methods for Business Students*, 8th ed. Harlow, UK: Pearson Education, 2019.
- [7] R. Burke, *Project Management: Planning and Control Techniques*, 5th ed. Hoboken, NJ: Wiley, 2013.
- [8] P. C. Dinsmore and J. Cabanis-Brewin, *The AMA Handbook of Project Management*, 4th ed. New York, NY: AMACOM, 2014.
- [9] J. W. Creswell and J. D. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 5th ed. Thousand Oaks, CA: Sage Publications, 2017.
- [10] J. Highsmith, *Agile Project Management: Creating Innovative Products*, 2nd ed. Boston, MA: Addison-Wesley, 2010.
- [11] A. Rolstadås *et al.*, "Risk navigation in project portfolios," *International Journal of Project Management*, vol. 32, no. 4, pp. 791–803, 2014.

APPENDIX-A

PSUEDOCODE

1.MODEL

Project Model:

```
package com.techm.rpm.MODEL;

import java.time.LocalDate;

import java.util.ArrayList;
import java.util.List;
import java.util.Objects;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.Lob;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.OneToMany;
import jakarta.persistence.Table;

@Entity
@Table(name = "researchproject")

public class Project {
```

@Override

```
public String toString() {  
    return "Project [projectId=" + projectId + ", title=" + title + ", description=" +  
description + ", status=" +  
        + status + ", startDate=" + startDate + ", endDate=" + endDate + ",  
faculty=" + faculty  
        + ", department=" + department + ", projectType=" + projectType + ",  
fundedAmount=" + fundedAmount  
        + ", fundingAgency=" + fundingAgency + "];"  
}
```

@Override

```
public int hashCode() {  
    return Objects.hash(department, description, endDate, faculty, fundedAmount,  
fundingAgency, projectId,  
        projectId, startDate, status, title);  
}
```

@Override

```
public boolean equals(Object obj) {  
    if (this == obj)  
        return true;  
    if (obj == null)  
        return false;  
    if (getClass() != obj.getClass())
```

```
        return false;

        Project other = (Project) obj;

        return Objects.equals(department, other.department) && Objects.equals(description,
other.description)

                && Objects.equals(endDate, other.endDate) &&
Objects.equals(faculty, other.faculty)

                && Objects.equals(fundedAmount, other.fundedAmount)

                && Objects.equals(fundingAgency, other.fundingAgency) &&
Objects.equals(projectId, other.projectId)

                && Objects.equals(projectType, other.projectType) &&
Objects.equals(startDate, other.startDate)

                && Objects.equals(status, other.status) && Objects.equals(title,
other.title);
    }
}
```

```
    public Project(String projectId, String title, String description, String status, LocalDate
startDate,

                LocalDate endDate, Faculty faculty, String department, String projectType,
Double fundedAmount,

                String fundingAgency) {

        super();

        this.projectId = projectId;

        this.title = title;

        this.description = description;

        this.status = status;

        this.startDate = startDate;
    }
}
```

```
this.endDate = endDate;

this.faculty = faculty;

this.department = department;

this.projectType = projectType;

this.fundedAmount = fundedAmount;

this.fundingAgency = fundingAgency;
}

@Id

@Column(name = "projectId",length = 6)
private String projectId;


@Column(length = 200, nullable = false)
private String title;


@Lob

private String description;


@Column(name = "status",length = 20)
private String status = "Proposed"; // No enum, stored as plain string


@Column(name = "startDate",nullable = false)
private LocalDate startDate;


@Column(name = "endDate",nullable = false)
private LocalDate endDate;
```

@ManyToOne

@JoinColumn(name = "facultyId", nullable = false)

private Faculty faculty;

@Column(name = "department",length = 100)

private String department;

@Column(name = "projectType",length = 10, nullable = false)

private String projectType = "Inhouse"; // 'Inhouse' or 'Funded'

@Column(name = "fundedAmount",columnDefinition = "DECIMAL(12,2)")

private Double fundedAmount = 0.0;

@Column(name = "fundingAgency",length = 200)

private String fundingAgency;

// Constructors

public Project() {}

// Getters and Setters

public String getProjectId() {

 return projectId;

}

public void setProjectId(String projectId) {

 this.projectId = projectId;


```
}
```

```
public String getTitle() {  
    return title;  
}
```

```
public void setTitle(String title) {  
    this.title = title;  
}
```

```
public String getDescription() {  
    return description;  
}
```

```
public void setDescription(String description) {  
    this.description = description;  
}
```

```
public String getStatus() {  
    return status;  
}
```

```
public void setStatus(String status) {  
    this.status = status;  
}
```

```
public LocalDate getStartDate() {  
    return startDate;  
}
```

```
public void setStartDate(LocalDate startDate) {  
    this.startDate = startDate;  
}
```

```
public LocalDate getEndDate() {  
    return endDate;  
}
```

```
public void setEndDate(LocalDate endDate) {  
    this.endDate = endDate;  
}
```

```
public Faculty getFaculty() {  
    return faculty;  
}
```

```
public void setFaculty(Faculty faculty) {  
    this.faculty = faculty;  
}
```

```
public String getDepartment() {  
    return department;  
}
```

```
}
```

```
public void setDepartment(String department) {  
    this.department = department;  
}
```

```
public String getProjectType() {  
    return projectType;  
}
```

```
public void setProjectType(String projectType) {  
    this.projectType = projectType;  
}
```

```
public Double getFundedAmount() {  
    return fundedAmount;  
}
```

```
public void setFundedAmount(Double fundedAmount) {  
    this.fundedAmount = fundedAmount;  
}
```

```
public String getFundingAgency() {  
    return fundingAgency;  
}
```

```
public void setFundingAgency(String fundingAgency) {  
    this.fundingAgency = fundingAgency;  
}  
  
@OneToMany(mappedBy = "project", cascade = CascadeType.ALL, orphanRemoval =  
true)  
  
private List<Student> students = new ArrayList<>();  
}
```

Faculty Model:

```
package com.techm.rpm.MODEL;  
  
import java.util.Objects;  
  
import jakarta.persistence.Column;  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;  
  
@Entity  
@Table(name = "faculty")  
public class Faculty {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Integer facultyId;
```

```
@Column(name = "facultyName",length = 50, nullable = false)
```

```
private String facultyName;
```

```
@Column(name = "facultyEmail",length = 50, nullable = false, unique = true)
```

```
private String facultyEmail;
```

```
@Column(name = "facultyPassword",length = 50, nullable = false)
```

```
private String facultyPassword;
```

```
@Column(name = "facultyMobile",length = 20, nullable = false, unique = true)
```

```
private String facultyMobile;
```

```
@Column(name = "facultyDepartment",length = 50)
```

```
private String facultyDepartment;
```

```
public Integer getFacultyId() {
```

```
    return facultyId;
```

```
}
```

```
public void setFacultyId(Integer facultyId) {
```

```
    this.facultyId = facultyId;
```

```
}
```

```
public String getFacultyName() {
```

```
    return facultyName;
```

```
}
```

```
public void setFacultyName(String facultyName) {  
    this.facultyName = facultyName;  
}
```

```
public String getFacultyEmail() {  
    return facultyEmail;  
}
```

```
public void setFacultyEmail(String facultyEmail) {  
    this.facultyEmail = facultyEmail;  
}
```

```
public String getFacultyPassword() {  
    return facultyPassword;  
}
```

```
public void setFacultyPassword(String facultyPassword) {  
    this.facultyPassword = facultyPassword;  
}
```

```
public String getFacultyMobile() {  
    return facultyMobile;  
}
```

```
public void setFacultyMobile(String facultyMobile) {  
    this.facultyMobile = facultyMobile;  
}
```

```
public String getFacultyDepartment() {  
    return facultyDepartment;  
}
```

```
public void setFacultyDepartment(String facultyDepartment) {  
    this.facultyDepartment = facultyDepartment;  
}
```

```
public Faculty(Integer facultyId, String facultyName, String facultyEmail, String  
facultyPassword,  
                String facultyMobile, String facultyDepartment) {  
    super();  
    this.facultyId = facultyId;  
    this.facultyName = facultyName;  
    this.facultyEmail = facultyEmail;  
    this.facultyPassword = facultyPassword;  
    this.facultyMobile = facultyMobile;  
    this.facultyDepartment = facultyDepartment;  
}
```

```
@Override
```

```
public String toString() {
```

```
        return "Faculty [facultyId=" + facultyId + ", facultyName=" + facultyName + ",  
facultyEmail=" + facultyEmail  
        + ", facultyPassword=" + facultyPassword + ", facultyMobile=" +  
facultyMobile + ", facultyDepartment=" +  
        + facultyDepartment + "];  
    }
```

@Override

```
public int hashCode() {  
    return Objects.hash(facultyDepartment, facultyEmail, facultyId, facultyMobile,  
facultyName, facultyPassword);  
}
```

@Override

```
public boolean equals(Object obj) {  
    if (this == obj)  
        return true;  
    if (obj == null)  
        return false;  
    if (getClass() != obj.getClass())  
        return false;  
    Faculty other = (Faculty) obj;  
    return Objects.equals(facultyDepartment, other.facultyDepartment)  
        && Objects.equals(facultyEmail, other.facultyEmail) &&  
Objects.equals(facultyId, other.facultyId)  
        && Objects.equals(facultyMobile, other.facultyMobile) &&  
Objects.equals(facultyName, other.facultyName)
```



```
        && Objects.equals(facultyPassword, other.facultyPassword);  
    }  
  
    public Faculty() {  
        super();  
    }  
  
    // Constructors, Getters, Setters  
}
```

Admin Model:

```
package com.techm.rpm.MODEL;  
  
import java.util.Objects;  
  
import jakarta.persistence.*;  
  
@Entity  
@Table(name = "admin")  
public class Admin {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private int adminId;  
  
    @Column(name = "adminName",length = 50)  
    private String adminName;
```

```
@Column(name = "adminEmail",length = 50, unique = true)
```

```
private String adminEmail;
```

```
@Column(name = "adminPassword",length = 50)
```

```
private String adminPassword;
```

```
// Getters and Setters
```

```
public int getAdminId() {
```

```
    return adminId;
```

```
}
```

```
public void setAdminId(int adminId) {
```

```
    this.adminId = adminId;
```

```
}
```

```
public String getAdminName() {
```

```
    return adminName;
```

```
}
```

```
public void setAdminName(String adminName) {
```

```
    this.adminName = adminName;
```

```
}
```

```
public String getAdminEmail() {
```

```
    return adminEmail;
}
```

```
public void setAdminEmail(String adminEmail) {
    this.adminEmail = adminEmail;
}
```

```
public String getAdminPassword() {
    return adminPassword;
}
```

```
public void setAdminPassword(String adminPassword) {
    this.adminPassword = adminPassword;
}
```

@Override

```
public String toString() {
    return "Admin [adminId=" + adminId + ", adminName=" + adminName + ",
adminEmail=" + adminEmail
        + ", adminPassword=" + adminPassword + "];"
}
```

```
public Admin(int adminId, String adminName, String adminEmail, String adminPassword) {
    super();
    this.adminId = adminId;
    this.adminName = adminName;
}
```

```
        this.adminEmail = adminEmail;

        this.adminPassword = adminPassword;
    }

    @Override
    public int hashCode() {
        return Objects.hash(adminEmail, adminId, adminName, adminPassword);
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Admin other = (Admin) obj;
        return Objects.equals(adminEmail, other.adminEmail) && adminId == other.adminId
            && Objects.equals(adminName, other.adminName) &&
            Objects.equals(adminPassword, other.adminPassword);
    }

    public Admin() {
        super();
    }
}
```

```
}
```

2.Controller

Login Controller:

```
package com.techm.rpm.CONTROLLER;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.techm.rpm.MODEL.Admin;
import com.techm.rpm.MODEL.Faculty;
import com.techm.rpm.REPOSITORY.AdminRepository;
import com.techm.rpm.REPOSITORY.FacultyRepository;
import com.techm.rpm.SERVICE.ProjectService;

import jakarta.servlet.http.HttpSession;

@Controller

public class LoginController {
```

```
    @Autowired
```

```
private AdminRepository adminRepo;
```

```
@Autowired
```

```
private FacultyRepository facultyRepo;
```

```
@Autowired
```

```
private ProjectService projectService;
```

```
@GetMapping("/")
```

```
public String showLoginPage() {
```

```
    return "home";
```

```
}
```

```
@GetMapping("/viewlogin")
```

```
public String showLoginForm() {
```

```
    return "login"; // login.jsp or login.html in templates/
```

```
}
```

```
@PostMapping("/login")
```

```
public String login(@RequestParam String email, @RequestParam String password,
```

```
                  @RequestParam String role, Model model, HttpSession session) {
```

```
    if ("admin".equalsIgnoreCase(role)) {
```

```
        Optional<Admin>
```

```
        admin
```

```
=
```

```
adminRepo.findByAdminEmailAndAdminPassword(email, password);
```

```
    if (admin.isPresent()) {
```

```
        session.setAttribute("admin", admin.get());
```

```

model.addAttribute("admin", admin.get());

model.addAttribute("totalProjects", projectService.getTotalProjects());

model.addAttribute("completedProjects", projectService.getCompletedProjects());

model.addAttribute("fundedProjects", projectService.getFundedProjects());


double    totalAmount    =    projectService.getTotalFundedAmount();    //    or
getTotalfacultyFundedAmount

String formattedAmount = "₹" + formatIndianCurrency(totalAmount);

model.addAttribute("totalFundedAmount", formattedAmount);


model.addAttribute("countCseProjects", projectService.countCseProjects());
model.addAttribute("countEceProjects", projectService.countEceProjects());
model.addAttribute("countEeProjects", projectService.countEeProjects());
model.addAttribute("countMeProjects", projectService.countMeProjects());
model.addAttribute("countCeProjects", projectService.countCeProjects());
model.addAttribute("countInhouseProjects",
projectService.countInhouseProjects());

return "admin/new-admin-dashboard";

}

} else if ("faculty".equalsIgnoreCase(role)) {

    Optional<Faculty>                faculty                =
facultyRepo.findByFacultyEmailAndFacultyPassword(email, password);

    if (faculty.isPresent()) {

        session.setAttribute("faculty", faculty.get());

        model.addAttribute("faculty", faculty.get());

        model.addAttribute("totalProjects", projectService.getTotalfacultyProjects());

```

```
        model.addAttribute("completedProjects",
projectService.getCompletedfacultyProjects());

        model.addAttribute("fundedProjects", projectService.getFundedfacultyProjects());


        double totalAmount = projectService.getTotalfacultyFundedAmount(); // or
getTotalfacultyFundedAmount

        String formattedAmount = "₹" + formatIndianCurrency(totalAmount);

        model.addAttribute("totalFundedAmount", formattedAmount);


        model.addAttribute("countCseProjects",
projectService.countfacultyCseProjects());

        model.addAttribute("countEceProjects",
projectService.countfacultyEceProjects());

        model.addAttribute("countEeProjects", projectService.countfacultyEeProjects());

        model.addAttribute("countMeProjects", projectService.countfacultyMeProjects());

        model.addAttribute("countCeProjects", projectService.countfacultyCeProjects());

        model.addAttribute("countInhouseProjects",
projectService.countfacultyInhouseProjects());

        return "admin/new-admin-dashboard";

    }

}

model.addAttribute("status", "failed");

return "login";

}
```



```
@GetMapping("/logout")
public String logout(HttpSession session) {
    session.invalidate(); // clear session
    return "redirect:/viewlogin";
}

private String formatIndianCurrency(double amount) {
    String amountStr = String.format("%.2f", amount);
    String[] parts = amountStr.split("\\.");
    String intPart = parts[0];
    String decPart = parts[1];

    // Apply Indian format to the integer part
    char[] digits = intPart.toCharArray();
    StringBuilder result = new StringBuilder();

    int len = digits.length;
    int count = 0;

    for (int i = len - 1; i >= 0; i--) {
        result.append(digits[i]);
        count++;
        if (i > 0) {
            if ((count == 3) || (count > 3 && (count - 3) % 2 == 0)) {
```

```
        result.append(",");
    }
}

// Reverse to get final formatted value
result.reverse().append(".").append(decPart);
return result.toString();
}

}
```

Project Controller:

```
package com.techm.rpm.CONTROLLER;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.techm.rpm.MODEL.Admin;
```

```
import com.techm.rpm.MODEL.Faculty;

import com.techm.rpm.MODEL.Project;

import com.techm.rpm.SERVICE.ProjectService;

import com.techm.rpm.SERVICE.StudentService;


import jakarta.servlet.http.HttpSession;


@Controller

public class ProjectController {

    @Autowired

    private ProjectService service;

    @Autowired

    private StudentService studservice;


    @GetMapping("/projectlist")

    public String viewProjects(Model model) {

        model.addAttribute("projects", service.getAllProjects());

        return "faculty/new-faculty-projects";

    }

    @GetMapping("/admin-project-list")

    public String viewAllProjects(Model model, HttpSession session) {

        Admin admin = (Admin) session.getAttribute("admin"); // Store in session


        model.addAttribute("admin", admin);

        model.addAttribute("projects", service.getAllProjects());

        return "admin/new-admin-projects";

    }

}
```

```
}
```

```
@GetMapping("/faculty-project-list")
```

```
public String viewFacultyProjects( Model model, HttpSession session) {
```

```
    Faculty faculty = (Faculty) session.getAttribute("faculty");
```

```
    int id = faculty.getFacultyId();
```

```
    List<Project> projects = service.fetchFacultyProjects(id);
```

```
    model.addAttribute("projects", projects);
```

```
    model.addAttribute("facultyId", id);
```

```
    model.addAttribute("faculty", faculty);
```

```
    return "admin/new-admin-projects"; // This should map to view.html in the facultyprojects  
folder
```

```
}
```

```
@GetMapping("/project/new")
```

```
public String addForm(Model model, HttpSession session) {
```

```
    Admin admin = (Admin) session.getAttribute("admin");
```

```
    if (admin != null) {
```

```
        model.addAttribute("project", new Project());
```

```
        model.addAttribute("admin", admin);
```

```
        return "new-add-project";
```

```
    }
```

```
// Check if faculty is logged in
```

```
Faculty faculty = (Faculty) session.getAttribute("faculty");

if (faculty != null) {

    model.addAttribute("project", new Project());

    model.addAttribute("faculty", faculty);

    return "new-add-project";

}

return "redirect:/login";

}

@PostMapping("/addproject")

public String saveProject(@ModelAttribute Project project, Model model, HttpSession
session) {

    service.saveProject(project);

    Admin admin = (Admin) session.getAttribute("admin");

    if (admin != null) {

        model.addAttribute("admin", admin);

        return "redirect:/admin-project-list";

    }

    // Check if faculty is logged in

    Faculty faculty = (Faculty) session.getAttribute("faculty");

    if (faculty != null) {

        model.addAttribute("faculty", faculty);

        return "redirect:/faculty-project-list";

    }

}
```

```
// If no one is logged in (optional fallback)

return "redirect:/login";

}

@GetMapping("/project/edit/{projectId}")

public String editForm(@PathVariable String projectId, Model model, HttpSession session)
{
    Admin admin = (Admin) session.getAttribute("admin");
    if (admin != null) {
        model.addAttribute("admin", admin);
        model.addAttribute("students", studservice.getStudentsInProject(projectId));
        model.addAttribute("project", service.getProjectById(projectId));
        return "new-edit-project";
    }

    // Check if faculty is logged in
    Faculty faculty = (Faculty) session.getAttribute("faculty");
    if (faculty != null) {
        model.addAttribute("faculty", faculty);
        model.addAttribute("students", studservice.getStudentsInProject(projectId));
        model.addAttribute("project", service.getProjectById(projectId));
        return "new-edit-project";
    }

    // If no one is logged in (optional fallback)

    return "redirect:/login";
}
```

```
    }

    @PostMapping("/project/update/{projectId}")

    public String updateProject(@PathVariable String projectId, @ModelAttribute Project
project, Model model, HttpSession session) {

        project.setProjectId(projectId);

        service.saveProject(project);

        Admin admin = (Admin) session.getAttribute("admin");

        if (admin != null) {

            model.addAttribute("admin", admin);

            return "redirect:/admin-project-list";

        }

        // Check if faculty is logged in

        Faculty faculty = (Faculty) session.getAttribute("faculty");

        if (faculty != null) {

            model.addAttribute("faculty", faculty);

            return "redirect:/faculty-project-list";

        }

        // If no one is logged in (optional fallback)

        return "redirect:/login";

    }

    @GetMapping("/project/delete/{projectId}")

    public String deleteProject(@PathVariable String projectId, HttpSession session, Model
model) {

        service.deleteProject(projectId);
```

```
Admin admin = (Admin) session.getAttribute("admin");  
  
if (admin != null) {  
    model.addAttribute("admin", admin);  
    return "redirect:/admin-project-list";  
}
```

```
// Check if faculty is logged in  
  
Faculty faculty = (Faculty) session.getAttribute("faculty");  
  
if (faculty != null) {  
    model.addAttribute("faculty", faculty);  
    return "redirect:/faculty-project-list";  
}
```

```
// If no one is logged in (optional fallback)  
  
return "redirect:/login";    }
```

```
@GetMapping("/faculty-dashboard")  
  
public String showFacultyDashboard(Model model, HttpSession session) {  
    Faculty faculty = (Faculty) session.getAttribute("faculty");  
  
    model.addAttribute("faculty", faculty);  
    model.addAttribute("totalProjects", service.getTotalfacultyProjects());  
    model.addAttribute("completedProjects", service.getCompletedfacultyProjects());  
}
```



```
model.addAttribute("fundedProjects", service.getFundedfacultyProjects());

double    totalAmount    =    service.getTotalfacultyFundedAmount();    //    or
getTotalfacultyFundedAmount

String formattedAmount = "₹" + formatIndianCurrency(totalAmount);
model.addAttribute("totalFundedAmount", formattedAmount);

model.addAttribute("countCseProjects", service.countfacultyCseProjects());
model.addAttribute("countEceProjects", service.countfacultyEceProjects());
model.addAttribute("countEeProjects", service.countfacultyEeProjects());
model.addAttribute("countMeProjects", service.countfacultyMeProjects());
model.addAttribute("countCeProjects", service.countfacultyCeProjects());
model.addAttribute("countInhouseProjects", service.countfacultyInhouseProjects());

return "admin/new-admin-dashboard"; // name of the HTML template (without .html)
}

@GetMapping("/admin-dashboard")
public String showAdminDashboard(Model model, HttpSession session) {
    Admin admin = (Admin) session.getAttribute("admin"); // Store in session

    model.addAttribute("admin", admin);
    model.addAttribute("totalProjects", service.getTotalProjects());
    model.addAttribute("completedProjects", service.getCompletedProjects());
    model.addAttribute("fundedProjects", service.getFundedProjects());

    double    totalAmount    =    service.getTotalFundedAmount();    //    or
    getTotalfacultyFundedAmount
```

```
String formattedAmount = "₹" + formatIndianCurrency(totalAmount);

model.addAttribute("totalFundedAmount", formattedAmount);


model.addAttribute("countCseProjects", service.countCseProjects());
model.addAttribute("countEceProjects", service.countEceProjects());
model.addAttribute("countEeProjects", service.countEeProjects());
model.addAttribute("countMeProjects", service.countMeProjects());
model.addAttribute("countCeProjects", service.countCeProjects());
model.addAttribute("countInhouseProjects", service.countInhouseProjects());

return "admin/new-admin-dashboard"; // name of the HTML template (without .html)
}
```

```
private String formatIndianCurrency(double amount) {

    String amountStr = String.format("%.2f", amount);

    String[] parts = amountStr.split("\\.");

    String intPart = parts[0];

    String decPart = parts[1];


    // Apply Indian format to the integer part

    char[] digits = intPart.toCharArray();

    StringBuilder result = new StringBuilder();

    int len = digits.length;

    int count = 0;

    for (int i = len - 1; i >= 0; i--) {
```

```
        result.append(digits[i]);

        count++;

        if (i > 0) {

            if ((count == 3) || (count > 3 && (count - 3) % 2 == 0)) {

                result.append(",");

            }

        }

    }

    // Reverse to get final formatted value

    result.reverse().append(".").append(decPart);

    return result.toString();

}

}
```

3.Service

Project Service:

```
package com.techm.rpm.SERVICE;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.techm.rpm.MODEL.Faculty;
import com.techm.rpm.MODEL.Project;
import com.techm.rpm.REPOSITORY.ProjectRepository;
```

```
import jakarta.servlet.http.HttpSession;

@Service

public class ProjectService {

    @Autowired

    private ProjectRepository repo;

    @Autowired

    private HttpSession session;

    public List<Project> getAllProjects() {

        return repo.findAll();

    }

    public Project saveProject(Project project) {

        return repo.save(project);

    }

    public void deleteProject(String pid) {

        repo.deleteById(pid);

    }

    public Project getProjectById(String pid) {

        return repo.findById(pid).orElse(null);

    }

}
```

```
public List<Project> fetchFacultyProjects(int facultyId) {  
    return repo.findByFaculty_FacultyId(facultyId);  
}
```

```
public long getTotalProjects() {  
    return repo.getTotalProjects();  
}
```

```
public long getCompletedProjects() {  
    return repo.getCompletedProjects();  
}
```

```
public long getFundedProjects() {  
    return repo.getFundedProjects();  
}
```

```
public double getTotalFundedAmount() {  
    Double amount = repo.getTotalFundedAmount();  
    return amount != null ? amount : 0.0;  
}
```

```
public long countCseProjects() {  
    return repo.countCseProjects();  
}
```

```
public long countEceProjects() {  
    return repo.countEceProjects();  
}
```

```
public long countEeProjects() {  
    return repo.countEeProjects();  
}
```

```
public long countMeProjects() {  
    return repo.countMeProjects();  
}
```

```
public long countCeProjects() {  
    return repo.countCeProjects();  
}
```

```
public long countInhouseProjects() {  
    return repo.countInhouseProjects();  
}
```

```
private int getFacultyIdFromSession() {  
    Object facultyObj = session.getAttribute("faculty");  
    if (facultyObj instanceof Faculty) {  
        return ((Faculty) facultyObj).getFacultyId();  
    } else {
```

```
        throw new IllegalStateException("Faculty not found in session");
    }
}

public long getTotalfacultyProjects() {
    return repo.getTotalProjectsByFaculty(getFacultyIdFromSession());
}

public long getCompletedfacultyProjects() {
    return repo.getCompletedProjectsByFaculty(getFacultyIdFromSession());
}

public long getFundedfacultyProjects() {
    return repo.getFundedProjectsByFaculty(getFacultyIdFromSession());
}

public Double getTotalfacultyFundedAmount() {
    return repo.getTotalFundedAmountByFaculty(getFacultyIdFromSession());
}

public long countfacultyCseProjects() {
    return repo.countCseProjectsByFaculty(getFacultyIdFromSession());
}

public long countfacultyEceProjects() {
    return repo.countEceProjectsByFaculty(getFacultyIdFromSession());
}
```

```
}
```

```
public long countfacultyEeProjects() {  
    return repo.countEeProjectsByFaculty(getFacultyIdFromSession());  
}
```

```
public long countfacultyMeProjects() {  
    return repo.countMeProjectsByFaculty(getFacultyIdFromSession());  
}
```

```
public long countfacultyCeProjects() {  
    return repo.countCeProjectsByFaculty(getFacultyIdFromSession());  
}
```

```
public long countfacultyInhouseProjects() {  
    return repo.countInhouseProjectsByFaculty(getFacultyIdFromSession());  
}  
}
```

Admin Service:

```
package com.techm.rpm.SERVICE;  
  
import java.util.Optional;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import com.techm.rpm.MODEL.Admin;  
import com.techm.rpm.REPOSITORY.AdminRepository;
```


@Service

```
public class AdminService {
```

@Autowired

```
    private AdminRepository adminRepository;
```

```
    // Method to authenticate a admin using email and password
```

```
    public Admin authenticateAdmin(String email, String password) {
```

```
        Optional<Admin>
```

```
        adminOptional
```

```
=
```

```
    adminRepository.findByAdminEmailAndAdminPassword(email, password);
```

```
        if (adminOptional.isPresent()) {
```

```
            return adminOptional.get(); // Return the admin if found
```

```
        } else {
```

```
            // Return null or throw a custom exception (for failed authentication)
```

```
            return null;
```

```
        }
```

```
    }
```

```
}
```

Faculty Service:

```
package com.techm.rpm.SERVICE;
```

```
import java.util.Optional;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Service;
```

```
import com.techm.rpm.MODEL.Faculty;
```

```
import com.techm.rpm.REPOSITORY.FacultyRepository;
```

@Service

```
public class FacultyService {
```

@Autowired

```
private FacultyRepository facultyRepository;

// Method to authenticate a faculty using email and password
public Faculty authenticateFaculty(String email, String password) {
    Optional<Faculty> facultyOptional =
facultyRepository.findByFacultyEmailAndFacultyPassword(email, password);
    if (facultyOptional.isPresent()) {
        return facultyOptional.get(); // Return the faculty if found
    } else {
        // Return null or throw a custom exception (for failed authentication)
        return null;
    }
}
```

4.REPOSITORY

Project Repository:

```
package com.techm.rpm.REPOSITORY;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

import com.techm.rpm.MODEL.Project;

public interface ProjectRepository extends JpaRepository<Project, String> {

    List<Project> findByFaculty_FacultyId(int facultyId);

    @Query("SELECT COUNT(p) FROM Project p")
    long getTotalProjects();
}
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.status = 'Completed'")
long getCompletedProjects();
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.projectType = 'Funded'")
long getFundedProjects();
```

```
@Query("SELECT SUM(p.fundedAmount) FROM Project p WHERE p.projectType =
'Funded'")
Double getTotalFundedAmount();
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'CSE'")
long countCseProjects();
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'ECE'")
long countEceProjects();
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'EE'")
long countEeProjects();
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'ME'")
long countMeProjects();
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'CE'")
long countCeProjects();
```

```
@Query(value = "SELECT COUNT(*) FROM researchproject WHERE
LOWER(projectType) = 'inhouse'", nativeQuery = true)
long countInhouseProjects();
```

```
// Total projects by faculty
```

```
@Query("SELECT COUNT(p) FROM Project p WHERE p.faculty.facultyId = :facultyId")
long getTotalProjectsByFaculty(@Param("facultyId") int facultyId);
```

```
// Completed projects by faculty
@Query("SELECT COUNT(p) FROM Project p WHERE p.status = 'Completed' AND
p.faculty.facultyId = :facultyId")
long getCompletedProjectsByFaculty(@Param("facultyId") int facultyId);

// Funded projects by faculty
@Query("SELECT COUNT(p) FROM Project p WHERE p.projectType = 'Funded' AND
p.faculty.facultyId = :facultyId")
long getFundedProjectsByFaculty(@Param("facultyId") int facultyId);

// Total funded amount by faculty
@Query("SELECT SUM(p.fundedAmount) FROM Project p WHERE p.projectType =
'Funded' AND p.faculty.facultyId = :facultyId")
Double getTotalFundedAmountByFaculty(@Param("facultyId") int facultyId);

// CSE projects by faculty
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'CSE' AND
p.faculty.facultyId = :facultyId")
long countCseProjectsByFaculty(@Param("facultyId") int facultyId);

// ECE projects by faculty
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'ECE' AND
p.faculty.facultyId = :facultyId")
long countEceProjectsByFaculty(@Param("facultyId") int facultyId);

// EE projects by faculty
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'EE' AND
p.faculty.facultyId = :facultyId")
long countEeProjectsByFaculty(@Param("facultyId") int facultyId);

// ME projects by faculty
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'ME' AND
p.faculty.facultyId = :facultyId")
```

```
long countMeProjectsByFaculty(@Param("facultyId") int facultyId);

// CE projects by faculty
@Query("SELECT COUNT(p) FROM Project p WHERE p.department = 'CE' AND
p.faculty.facultyId = :facultyId")
long countCeProjectsByFaculty(@Param("facultyId") int facultyId);

// Inhouse projects by faculty (Native query)
@Query(value = "SELECT COUNT(*) FROM researchproject WHERE
LOWER(projectType) = 'inhouse' AND facultyId = :facultyId", nativeQuery = true)
long countInhouseProjectsByFaculty(@Param("facultyId") int facultyId);
}
```

Admin Repository:

```
package com.techm.rpm.REPOSITORY;

import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;

import com.techm.rpm.MODEL.Admin;

public interface AdminRepository extends JpaRepository<Admin, Integer> {
    Optional<Admin>    findByAdminEmailAndAdminPassword(String email, String
password);
}
```

Faculty Repository:

```
package com.techm.rpm.REPOSITORY;

import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;

import com.techm.rpm.MODEL.Faculty;
```

```
public interface FacultyRepository extends JpaRepository<Faculty, Integer> {  
    Optional<Faculty> findByFacultyEmailAndFacultyPassword(String email, String  
password);  
}
```

APPENDIX-B

(SCREENSHOTS)

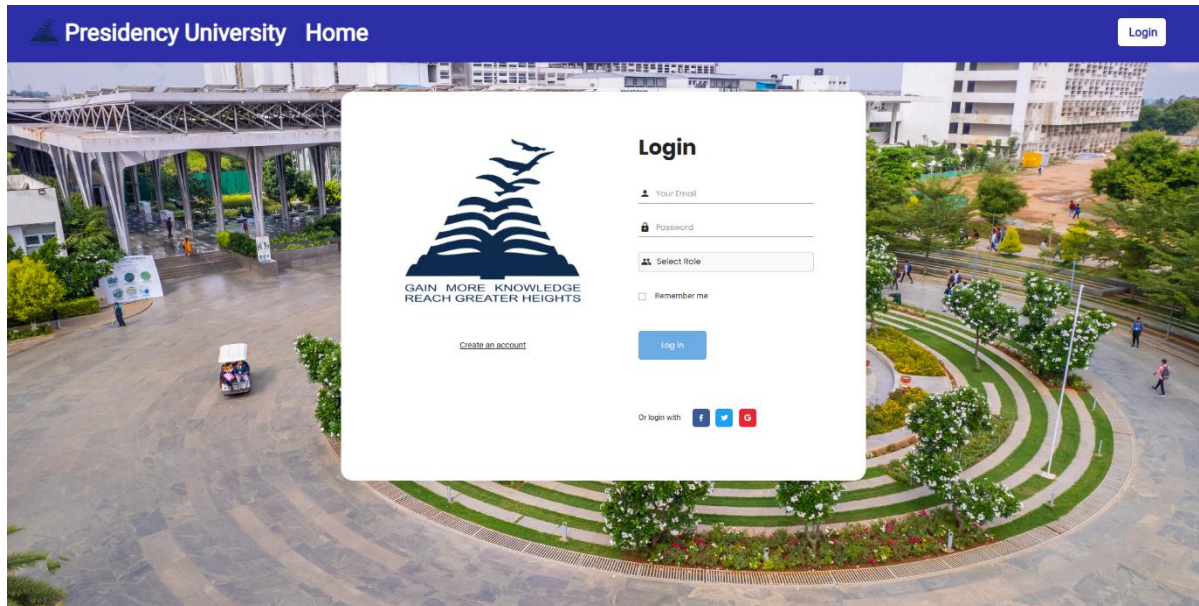


Figure 1: Login Page

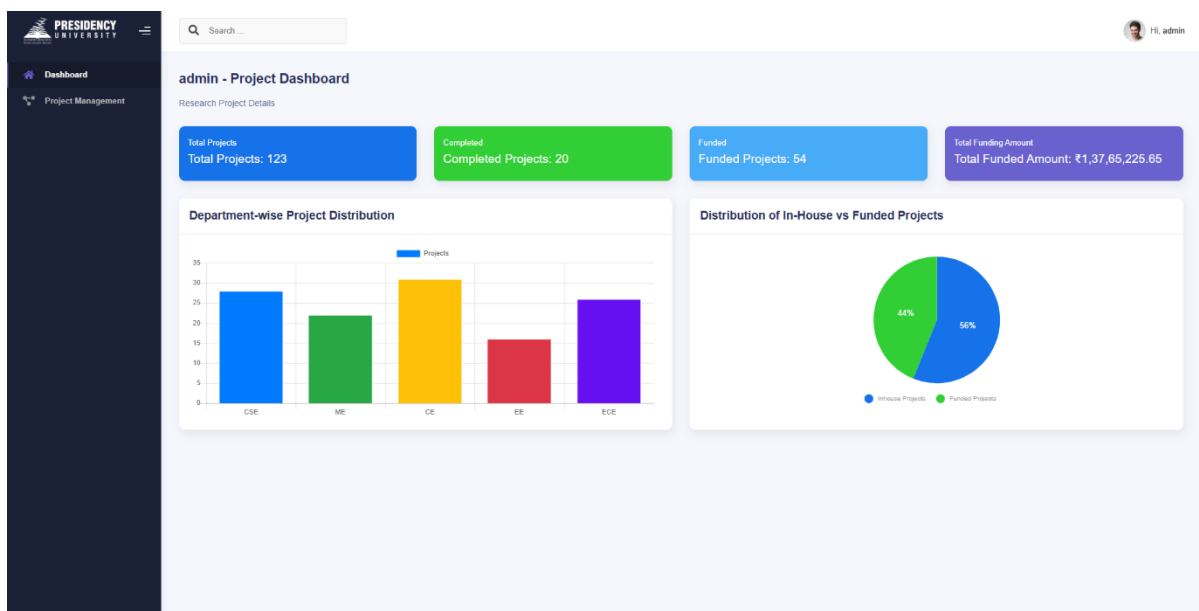


Figure 2: Admin And Faculty Project Dashboard

The screenshot displays the 'Project Management' dashboard for Presidency University. The interface includes a sidebar with 'Dashboard' and 'Project Management' options. The main content area is titled 'Project Management' and features a search bar, a '+ Add Project' button, and a table of 'Research Projects'. The table lists five projects with columns for ID, Title, Description, Status, Start Date, End Date, Faculty ID, Department, Project Type, Funded Amount, Funding Agency, and Action. The projects are: P001 (IoT for Smart Cities, Paused), P002 (Renewable Energy Sources, Paused), P003 (Robotics for Disaster Management, Ongoing), P004 (Blockchain in Finance, Ongoing), and P005 (IoT for Smart Cities, Proposed). A pagination bar at the bottom shows 'Showing 1 to 5 of 123 entries'.

PROJECT ID	TITLE	DESCRIPTION	STATUS	START DATE	END DATE	FACULTY ID	DEPARTMENT	PROJECT TYPE	FUNDED AMOUNT	FUNDING AGENCY	ACTION
P001	IoT for Smart Cities	Research on IoT for smart cities in EE department.	Paused	2020-09-04	2021-09-20	1	EE	Funded	335740.11	UGC	Edit Delete
P002	Renewable Energy Sources	Research on renewable energy sources in EE department.	Paused	2024-06-27	2024-12-34	3	EE	Inhouse	0.0		Edit Delete
P003	Robotics for Disaster Management	Research on robotics for disaster management in CE department.	Ongoing	2021-01-13	2022-09-10	3	CE	Inhouse	0.0		Edit Delete
P004	Blockchain in Finance	Research on blockchain in finance in CSE department.	Ongoing	2020-08-29		2	CSE	Funded	482857.67	DST	Edit Delete
P005	IoT for Smart Cities	Research on IoT for smart cities in ECE department.	Proposed	2022-08-25	2024-06-21	2	ECE	Inhouse	0.0		Edit Delete

Figure 3: Project Management

The screenshot shows the 'New Project Details' form in the Project Management dashboard. The form includes fields for Project ID, Title, Description, Status (with a dropdown menu), Start Date, End Date, Faculty ID, Department (with a dropdown menu), Project Type (with a dropdown menu), Funding Agency, and Funded Amount. At the bottom, there are 'Save' and 'Cancel' buttons.

Figure 4: New Project Details

Presidency University

Search ...

Hi, admin

Edit Project Details

Title: IoT for Smart Cities

Description: Research on IoT for smart cities in EE department.

Status: Paused

Start Date: 2020-09-04

End Date: 2021-09-20

Faculty ID: 1

Department: EE

Project Type: Funded

Funding Agency: UGC

Funded Amount: 336740.11

Save Cancel

Figure 5: Edit Project Details

Presidency University

Search ...

Hi, admin

Project Management

Add, Update or Delete Projects

Research Projects

Show: entities

Search:

PROJECT ID	TITLE	DESCRIPTION	STATUS	START DATE	END DATE	FACULTY ID	DEPARTMENT	PROJECT TYPE	FUNDED AMOUNT	FUNDING AGENCY	ACTION
P001	IoT for Smart Cities	Research on IoT for smart cities in EE department.	Paused	2020-09-04	2021-09-20	1	EE	Funded	336740.11	UGC	Edit Delete
P002	Renewable Energy Sources	Research on renewable energy sources in EE department.	Paused	2024-06-27	2024-12-24	3	EE	Inhouse	0.0		Edit Delete
P003	Robotics for Disaster Management	Research on robotics for disaster management in CE department.	Ongoing	2021-01-13	2022-09-10	3	CE	Inhouse	0.0		Edit Delete
P004	Blockchain in Finance	Research on blockchain in finance in CSE department.	Ongoing	2020-08-29		2	CSE	Funded	482857.67	DST	Edit Delete
P005	IoT for Smart Cities	Research on IoT for smart cities in ECE department.	Proposed	2022-08-25	2024-06-21	2	ECE	Inhouse	0.0		Edit Delete

Showing 1 to 5 of 123 entries

Previous 1 2 3 4 5 25 Next

Figure 6: Delete Project

APPENDIX-C

ENCLOSURES

1. Journal publication/Conference Paper Presented Certificates of all students.

Following the Guide's recommendation, we did not do any
Research Paper Publication.

Dr Jayanthi Kamalasekaran - Group 16
Project_Report_updated (1).pdf

ORIGINALITY REPORT

17%

SIMILARITY INDEX

14%

INTERNET SOURCES

12%

PUBLICATIONS

13%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Northampton Student Paper	1%
2	Submitted to University of Greenwich Student Paper	1%
3	Submitted to University of Central Lancashire Student Paper	1%
4	Submitted to Southern Arkansas University (Blackboard LTI 1.3) Student Paper	1%
5	www.coursehero.com Internet Source	1%
6	www.joiningtracks.org Internet Source	1%
7	Submitted to University of Huddersfield Student Paper	1%
8	vbook.pub Internet Source	1%
9	Submitted to Wilmington University Student Paper	1%
10	its203.com Internet Source	1%
11	Submitted to IUBH - Internationale Hochschule Bad Honnef-Bonn Student Paper	1%

SUSTAINABLE DEVELOPMENT GOALS



The Project Work Carried out here is mapped to SDG-04 Quality Education. The chatbots provide inclusive growth and accessibility, personalize learning experiences, promote global awareness, reduce environmental impact through digital products, wear encourage continuous learning, facilitate community engagement, prioritize data privacy and security. Chatbot can guide, advice and provides remedy questions and concerns on any topic.