**EduBridge**

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

**BACHELOR OF TECHNOLOGY IN**

**COMPUTER SCIENCE**

****

Submitted by

Kashish Tyagi (2100290120094)

Dewansh Gupta (2100290120070)

Kaushiki Verma (2100290120095)

Aryan Singh (2100290120049)

Supervised by

Prof. Vandana

**Session 2024-2025**

**DEPARTMENT OF COMPUTER SCIENCE**

# KIET GROUP OF INSTITUTIONS, GHAZIABAD

**(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India) May 2025**

# DECLARATION

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

Signature: Signature:

Name : Kashish Tyagi Name : Dewansh Gupta

Roll No.: 2100290120094 Roll No.: 2100290120070

Signature: Signature:

Name : Aryan Singh Name : Kaushiki Verma

Roll No.: 2100290120049 Roll No.: 2100290120095

# CERTIFICATE

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

.

##### Date:

**Supervisor** Ms.Vandana (Professor)

# ACKNOWLEDGEMENT

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

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

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

Date :

Signature: Signature:

Name : Dewansh Gupta Name : Kashish Tyagi

Roll No.: 2100290120070 Roll No.: 2100290120094

Signature: Signature:

Name : Aryan Singh Name : Kaushiki Verma

Roll No.: 2100290120049 Roll No.: 2100290120095

# ABSTRACT

In today's fast-paced and ever growing learning ecosystem, the teachers and learners are subjected to a variety of academic prospect — including seminar, competition, collaborative projects, and important alerts. Yet, the very abundance of these resources often leads to confusion. Information is scattered across platforms, difficult to track, and easy to miss.

This is where Edubridge comes in, it was born from a simple yet powerful idea: to bring everything academic into one unified, accessible platform. It aims to eliminate the chaos of fragmented information and instead create a smooth, streamlined experience for students, educators, and institutions.

Despite having so many opportunities available online, there’s a gap — a missing link that connects students to the right activities at the right time. Currently, the lack of a centralized system often means lost opportunities, mismatched priorities, and wasted time.

Edubridge addresses this by offering a one-stop solution where all academic activities can be easily discovered, tracked, and managed.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **Page No.** |
| DECLARATION……………………………………………………………………. | ii |
| CERTIFICATE……………………………………………………………………… | iii |
| ACKNOWLEDGEMENTS…………………………………………………………. | iv |
| ABSTRACT……………………………………………………………………….... | v |
| LIST OF FIGURES…………………………………………………………………. | ix |
| LIST OF TABLES…………………………………………………………………… | xi |
| LIST OF ABBREVIATIONS………………………………………………………. | xii |
| SDG MAPPING WITH JUSTIFICATION ………………………………………. | xii |
| CHAPTER 1 INTRODUCTION | 1-4 |
| 1.1 Introduction to Project | 1 |
| 1.2 Project Category | 2 |
| 1.3 Objectives | 2 |
| 1.4 Structure of Report | 4 |
| CHAPTER 2 LITERATURE REVIEW | 5-9 |
| 2.1 Literature Review | 5 |
| 2.2 Research Gaps | 7 |
| 2.3 Problem Formulation | 8 |
| CHAPTER 3 PROPOSED SYSTEM | 9-11 |
| 3.1 Proposed System | 9 |
| 3.2 Unique Features of The System | 10 |
| CHAPTER 4 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION | 11-26 |
| 4.1 Feasibility Study (Technical, Economical, Operational) | 11 |
| 4.2 Software Requirement Specification | 14 |
| 4.2.1 Data Requirement | 14 |
| 4.2.2 Functional Requirement | 19 |
| 4.2.3 Performance Requirement | 20 |

|  |  |
| --- | --- |
| * + 1. Maintainability Requirement     2. Security Requirement   1. SDLC Model Used   2. System Design      1. Data Flow Diagrams      2. Use Case Diagrams   4.5 Database Design  CHAPTER 5 IMPLEMENTATION   * 1. Introduction Tools and Technologies Used   2. Technologies and Tools Used   3. Platform Details   CHAPTER 6 TESTING, AND MAINTENANCE   * 1. Testing Techniques and Test Cases Used   2. Bug Tracking and Resolution   CHAPTER 7 RESULTS AND DISCUSSIONS   * 1. Description of Modules with Snapshots   2. Key findings of the project   3. Brief Description of Database with Snapshots   CHAPTER 8 CONCLUSION AND FUTURE SCOPE   * 1. Conclusion   2. Future Scope   REFERENCES  Research Paper Acceptance Proof  Proof of patent publication (Screenshot of Publication) | 21  21  22  22  23  24  25  26-33  26  30  31  34-39  34  37  40-46  40  44  44  47-50  47  48 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 4.1 | Dataflow Diagram | 20 |
| 4.2 | ER Diagram | 21 |
| 7.1 | Welcome | 37 |
| 7.2 | Home Screen | 38 |
| 7.3 | Mentor Module | 39 |
| 7.4 | Event Module | 40 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No.** | **Description** | **Page No.** |
| 5.1 | Technologies and Tools | 27 |
| 6.1 | Sample Test Cases | 34 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| B.Tech | Bachelor of Technology |
| UI | User Interface |
| UX | User Experience |
| IDE | Integrated Development Environment |
| DB | Database |
| JSON | JavaScript Object Notation |
| API | Application Programming Interface |
| SDLC | Software Development Life Cycle |
| FCM | Firebase Cloud Messaging |
| SRS | Software Requirements Specification |
| DFD | Data Flow Diagram |
| AI | Artificial Intelligence |
| ERP | Enterprise Resource Planning |
| CSC | Common Service Center |
| IEEE | Institute of Electrical and Electronics Engineers |
| APK | Android Package Kit |
| CRUD | Create, Read, Update, Delete |

**CHAPTER 1 INTRODUCTION**

#### Introduction

In today’s fast-moving digital world, education is changing faster than ever before. Online platforms, virtual classrooms, and digital tools are no longer optional — they’ve become essential parts of how we learn and teach. Schools and colleges everywhere are using technology to make learning easier, manage academic processes more efficiently, and keep students engaged.

But even with all this progress, one big problem still stands out — there’s no single place where students and educators can easily access and manage all their academic activities.Right now, important updates like seminar invitations, event details, competition announcements, and learning resources are scattered across multiple platforms — from emails and websites to WhatsApp groups and notice boards. This scattered information makes it hard for students to stay updated, difficult for mentors to guide effectively, and challenging for institutions to manage everything in an organized way. Because of this, there are so many lubricative opportunities that go untapped. Time is vanished, communication becomes muddled, and students lose out on experiencing things that can affect their academic life. Simply, there is a good and lubricative solution that is needed, thats is where Edubridge comes in.

Edubridge is trying its best to solve these many problems by putting all the student needed content on one single platform. With an easy-to-use dashboard, smart suggestion based on user interest, and collaborative tool that make collaboration smooth and friction-free, it will allow students, mentors, and colleges to be in a single cycle. The app is user-friendly, with even the most difficult information being easy to grab and follow. The platform has been crafted with atmost care, starting from the original reallife problems of teachers and learners, and drafting a solution which is tap on the money for their needs. Every feature has been drafted with the end-user in mind.

This initial brief just touches the surface of what Edubridge plans to do. It set the stage for a closer look at the platform vision, structure design, and the helpful impact it is making in simplifying academic life and building tighter roots and connection in education.

#### Project Category

In today’s fast-moving digital world, education is changing faster than ever before. Online platforms, virtual classrooms, and digital tools are no longer optional — they’ve become essential parts of how we learn and teach. Schools and colleges everywhere are using technology to make learning easier, manage academic processes more efficiently, and keep students engage The Edubridge platform falls within the realm of Educational Technology (EdTech) with a specific emphasis on scholarly engagement, centralized communication, and student lifecycle management. Edubridge is an initiative to solve a large and growing problem within modern education — the fragmented and typically overwhelming tidal wave of academic information dispersed across channels. Students and teachers within today's education system struggle with emails, group messages, notice boards, and various portals simply to stay updated about routine academic tasks and opportunities. Edubridge mitigates this chaos by creating an integrated platform that compiles all the associated information into one place. It offers a smart, intuitive dashboard that allows students to see seminar invitations, competition updates, learning material, and other relative updates at their own convenience — all tailored to their own interest. Mentors and institutions gain equally, as the platform provides collaborative tools that promote seamless communication and simplicity of academic life. What sets Edubridge apart from the rest is its inherent dedication towards solving the real issues of educators and learners, while building features that actually simplify their lives. This classification fits the nature of the project perfectly, as Edubridge is more than a tool — it is a well-thought solution to the day-to-day problems of academic life, designed to create an integrated, accessible, and efficient learning environment for all stakeholders.

#### Objectives

The prime objective of the EduBridge program is to create an all-embracing and inclusive electronic platform that acts as a hub for scholarly interaction, mentoring, and monitoring within an educational institution. Sensing the growing demand for effective communication, access, and collaborative work among the academic community, EduBridge attempts to unite students, faculty, mentors, and institutional resources.

This platform is meant to create an organized environment in which students can easily access academic resources, such as internships, scholarships, and events, in addition to enabling real- time communication between mentors and students. It complements the all-round growth of students by providing an organized framework for tracking academic performance, managing participation in events, receiving personalized advice, and being informed of institutional announcements and opportunities.

In addition, EduBridge is focused on reducing administrative complexities by automating mundane processes and ensuring streamlined information flow among the schools. With the integration of features like event management, profile management, opportunity cataloging, and feedback systems, the platform ensures access to a more comprehensive and integrated learning experience.

The main goal is to empower students through technology, increase transparency and effectiveness in institutional activities, and create a dynamic learning culture that fosters teamwork, mentoring, and academic excellence.

#### Structure of Report

Chapter 1: Introduction

Provides an overall description of the project, including its background, categorization, principal objectives, and the overall structure of the report.

Chapter 2: Literature Review

Evaluates the current body of literature and comparable platforms within the Educational Technology domain, determines gaps in current frameworks, and develops the problem statement from real-world issues.

Chapter 3: Proposed Framework

Explains the suggested solution — Edubridge — in detail as how it caters to the gaps identified and enumerates its distinguishing, user-centered features.

Chapter 4: System Specification and Requirement Analysis

Includes the technical, economic, and operational feasibility study, and comprehensive software requirements such as data, functional, performance, maintainability, and security

requirements. Also includes the selected SDLC model, system design (data flow and use case diagrams), and database design.

Chapter 5: Implementation

Stresses the development stage, describing the tools, technologies, and infrastructures utilized in the building and deployment of the platform.

Chapter 6: Testing and Maintenance

Explain the test procedures used, test cases run, and maintenance policies in place to provide a stable and secure system.

Chapter 7: Findings and Analysis

The modules developed are shown through screen shots, important results are examined, and a summary of the database schema is provided along with the respective screen shots.

Chapter 8: Overview and Future Directions Summarizes the achievements of the project, determines its relevance, and proposes potential areas of future development and improvement.

References and Support Materials Includes all scholarly sources utilized, as well as evidence of research paper acceptance and screenshots of patent publication to authenticate the novelty and contribution of the project.

# CHAPTER 2

## Literature Review

##### Literature Review

With the fast pace of the present-day education landscape, digital technology has evolved from a luxury to a necessity. Digital invention are not only restructuring the manner in which we teach in the classroom but also the manner in which we manage colleges. With the growing acclaim of hybrid and pliable learning spaces, there is a growing necessity for systems that are able to cohesively bring together all the constituents—learner, teachers, and admins—into a harmonized experience.

Most of the available literature focuses on the application of Enterprise Resource Planning (ERP) systems in schools. ERP systems have proven to be effective in the management of admin tasks, such as admissions, taking attendance, administering exams, and distributing resources. But there is this huge gap; most of the ERP system overlook most of the parts of the intellectual process that are aimed at engaging the leaners, such as attend seminars, activities, mentorship, or working with fellow learners.

It was concluded in a study conducted by Tella and Bashorun (2011) that the handiness and user-friendliness of virtual academic portals play a pivotal role in shaping the academic satisfaction and performance of students. The authors promoted platforms that transcend the storage of information; these platforms should also support active learning and interaction.

Saxena (2013) provided a twist by studying India's Common Service Centers (CSCs) under the e-Governance initiative. While the centers were not set up with the only goal of research, CSCs demonstrated how digital center can enable rural and far off area through the delivery of major services, e.g., education. The main solution is that with effective implementation, central digital platforms can dismantle barriers and promote inclusivity.

Jun and Chen (2011) presented a solid argument in their paper regarding the incorporation of employment services into government systems. Although they were most concerned with job placement and alumni relations, their study proved that centralization of resources can streamline processes and make them more available—lessons that can be applied to education

platforms too. Another significant input was from Ambrose et al. (2008) in favor of the adoption of Free and Open Source Software (FOSS) for e-learning. In their opinion, open- source systems present cost benefits and flexibility, especially if they are built upon modular, community-oriented designs. They did mention, however, that proprietary systems are not typically equipped with personalization capabilities — something that is essential to keeping students interested.

Recent studies by Chatterjee (2022) and Sharma & Verma (2020) indicate a change in student requirements. Students today do not only require functionality but also platforms that enable social interaction, are mobile, and are customized to their current needs. The function of user experience has become increasingly vital in this context.

Even with all the advancements, many educational tools available today remain subpar. The tools tend to operate in silos, addressing either administrative or academic functions without developing interlinkages. They are not cross-functionally integrated, a failing that prevents correlating activities such as attending events with learning accomplishments or mentorship opportunities. Such a failure of interlink between the services offered by schools and the actual needs of students is precisely what motivated the development of EduBridge.

EduBridge is seen as an integrated learning environment — not just an electronic bulletin board, but rather an active and interactive community where students can discover opportunities, mentors can discover mentees, and institutions can build an environment that is supportive of participation and collaboration. Leveraging insights from existing research and embracing available technology, EduBridge aims to bridge gaps and integrate learning experiences into the very fabric of one.

##### Research Gaps

While scholarly research has examined a wide range of educational technology and their roles within schools, there exist significant and pressing gaps in the execution of these systems— and, more importantly, how well they are performing for the people who rely upon them, especially students. It is necessary to take close examination of some of the most significant challenges:

##### Scattered Systems, Scattered Focus

Most schools employ different platforms to fulfill different purposes—one for management of course content, one for administrative functions, and maybe a few more for events or clubs. The availability of the combined space that harmonizes all of these is not prevalent. This segmented setup forces students to work with multiple programs, and this leads to lost opportunities and more of an impression of confusion than of sufficient support.

##### Limited Room to Connect

Most education platforms function properly; however, they do not have a human touch. These platforms usually never have real-time features like live chat functionality, event registration functionality, feedback mechanisms, or even simple mechanisms whereby students can interact with mentors. This constraint is very common in big institutions, which can lead to alienation, where the students will view themselves as just another identifier in the system.

##### One Size Fits None

Contemporary systems usually fail to include artificial intelligence or behavioral information for the purpose of suggesting customized content—like events that match a student's interest or mentors with similar career aspirations. Without this customization, sites usually appear bland and generic, thus limiting user participation and encouragement.

##### Built for Yesterday, Not Tomorrow

When educational institutions grow and reach across campuses, many platforms struggle to keep pace with the changes. The platforms were never built for scalability or support of today's collaborative needs—like shared project spaces, group discussions, or team collaboration. This makes it difficult for students and guides to collaborate in meaningful ways.

##### Still Waiting on That Email?

Today's students are mobile-first. They expect to receive immediate notifications pushed directly to their phones. But most higher ed systems still use outdated strategies such as email or bulletin boards — which are simple to miss and usually too sluggish to be of much value.

All these challenges point to a shared root issue: the essential requirement for a smart, centralized, and human-focused system — a system that not only manages the academic drudgery but also provides an experience of visibility, aid, and real connection to the students.

EduBridge is not just another learning resource; it is, instead, a carefully thought-out solution to a relevant real-world issue. Its architecture facilitates integrating multiple aspects into one platform, thus augmenting the accessibility, collaboration, and customization of campus life for all stakeholders.

##### Problem Formulation

Based on the challenges and findings of the previous studies discovered, that there is an urgent and critical necessity for an integrated academic system, a system that effectively integrates students, teachers, and institutions into an active and symbiotic environment.

Contemporary ERP systems are highly effective in carrying out administrative tasks, like tracking attendance and test scheduling; yet, they can only be effective in meeting rudimentary operational needs. ERP systems are not adequately supportive of interaction and lack meaningful relationships. On the other hand, while social networking websites and simple learning environments can be interactive, they are unlikely to have the academic intensity or orderly engagement that can be expected within formal education.

This vacuum — lack of a system of mixed-mode academic participation — separates us all. Students miss the advantage of learning that would make them more professional and well- rounded. Faculty cannot mentor and contribute constructively. Institutions are not able to feel a sense of community across their campuses.

It is here that the basic question exists. This takes us to the ultimate reason for this project:

The purpose is to design and deploy a robust academic engagement platform that addresses every dimension of scholarly work, enables real-time collaboration, and offers customized experiences for students, mentors, and institutions. The initiative is not a collection of palliative solutions to current issues but a radical paradigm change in our way of approaching scholarly activities.

EduBridge is our forward-thinking response to this challenge: a carefully crafted and forward- thinking solution that seeks to improve academic participation, promote more integration, enable more sharing, and ultimately make it more human-centered.

# CHAPTER 3

## Proposed System

##### Proposed System

If we're going to transform the student experience of higher education itself, we can't be satisfied with one more admin portal or notice board. We need something that feels like it's been built by and for the people who will be using it on a daily basis. That's where EduBridge steps in—a single, interactive mobile app that aims to unify all the stakeholders of the academic world: learner, mentors, societies, and institutions. EduBridge isn't sharing information—it's creating a space where learning is effortless, tailored, and accessible.

Now, precious opportunities such as seminars, workshops, competitions, and club activities are lost in WhatsApp messages or remain stuck on notice boards. Students are therefore being deprived, mentors are losing touch, and administrators are finding it hard to track participation. EduBridge consolidates all of this within one, mobile-focused platform—where finding, entering, and organizing scholarly work is at your fingertips.

##### What Makes EduBridge Work

At its core, EduBridge is built on a modular, scalable architecture, ensuring it evolves with an institution’s growing needs. Technically, it’s developed using Java and XML on Android Studio, supported by Firebase Realtime Database for fast, cloud-based syncing and Firebase Auth for secure logins and user identity protection.

##### Here's how the app is structured:

* + - 1. User Registration & Role-Based Logins

Simple sign-ups with customized dashboards for students, mentors, and admins.

* + - 1. Personalized Activity Feeds

A centralized feed showing relevant events, announcements, and curated opportunities.

* + - 1. Smart Mentorship Matching

An intelligent pairing system that connects students with mentors based on shared interests and career goals.

* + - 1. Event & Club Management Tools

Tools for organizing, joining, and managing campus events and society activities with ease.

* + - 1. Push Notifications

Timely updates that actually reach users—no more missed announcements or forgotten opportunities. To ensure usability—especially for first- and second-year students—we’ve prioritized intuitive UI design, familiar navigation patterns, and guided onboarding screens that support users from the moment they log in.

##### Unique Features of the System

EduBridge isn’t just another campus tool—it’s a whole new way for students to experience academic life. Designed with the real needs of students and educators in mind, it goes beyond simply delivering information. EduBridge helps make academic life more connected, easier to navigate, and a whole lot more engaging.

Here’s what makes EduBridge stand out:

##### One Place for Everything

Tired of digging through WhatsApp groups or random notice boards for updates? EduBridge brings all your academic notices, events, club activities, and mentorship opportunities into one clean, organized space. It saves time and ensures no one misses out—whether you're a newcomer or an introvert trying to stay involved.

##### Smart Mentor Suggestions

Finding the right mentor shouldn’t feel like a guessing game. EduBridge uses a smart recommendation engine that considers your interests, past activities, and subject focus to match you with mentors who actually align with your goals and learning style.

##### Dashboards That Actually Make Sense

No clutter. No confusion. Whether you're a student, mentor, or admin, your dashboard is tailored to show exactly what you need—like upcoming events, mentorship requests, or quick- access tools relevant to your role.

##### Instant Alerts When It Matters

With Firebase Cloud Messaging, EduBridge keeps you in the loop in real time.Event coming up? Registration closing soon? New mentorship slots open?You’ll know instantly—no more lost emails or forgotten updates.

##### Easy Event Discovery & Sign-Ups

Browsing events is simple and even fun.Filter by category—technical, cultural, or social—and register with just a few taps.

##### Built-in Student Communities

EduBridge goes beyond academics. With group chats, Q&A boards, and post-event discussions, it’s easy to collaborate, get feedback, and feel part of something bigger.

##### Clubs Get the Spotlight Too

Clubs and societies aren’t sidelined—they’re at the center. Each club gets its own profile to post updates, manage memberships, and engage with students directly. It’s like giving every club its own mini-app inside EduBridge.

##### Safe, Fast, and Ready to Grow

Powered by Firebase, EduBridge is not just secure—it’s built to scale. Whether your college has 500 students or 5,000, the app runs smoothly with encrypted data, role-based access, and real-time syncing.

# CHAPTER 4

## Requirement Analysis and System Specifications

##### Feasibility Study

Before building EduBridge, we undertook a comprehensive feasibility study to determine if the platform was viable and achievable. We reviewed technical aspects, where we determined that development and scalability were possible with tools like Java, Android Studio, and Firebase. Economically, we determined that the project was feasible within the budget via cheap cloud services. Operationally, user, mentor, and admin feedback determined that EduBridge would effectively address actual problems like disjointed communication and low participation. Overall, the study determined that EduBridge is viable and worth the effort to users.

##### Technical Feasibility

The Technical feasibility of EduBridge was assessed by checking the availability and compatibility of development tools, deployment platforms, back-end infrastructure, and integration capabilities. The system is built using the following major technologies:

* **Java and XML:** They are the default programming language options for developing Android applications, offering mature libraries, cross-device compatibility, and enormous community support.
* **Android Studio:** A powerful and well-documented IDE with the capability for real- time debugging, graphical layout editing, and built-in emulator testing.
* **Firebase:** As the backend-as-a-service (BaaS) platform, Firebase offers a strong suite of cloud tools that consists of:
  + Realtime Database for real-time data syncing,
  + Firebase Authentication for secure and role-based user login,
  + Cloud Messaging (FCM) for push notifications,
  + Crashlytics for fast-time error reporting and stability tracking.

This tech stack is lightweight, mobile-optimized, and allows seamless real-time interaction between users and the database. The development process was further enhanced by the use of Figma for UI/UX design and prototyping, which allowed early visualization and feedback collection.

The platform was tested across various Android API levels to ensure device compatibility, and it performed well on devices with moderate to high processing capabilities. Firebase’s auto- scaling infrastructure ensures that EduBridge can grow alongside institutional demand, handling thousands of simultaneous reads/writes with minimal latency.

In conclusion, the tools and technologies selected are technically sound, scalable, and support all the functional requirements of the platform, thereby validating its technical feasibility.

##### Economic Feasibility

The economic feasibility of EduBridge was a key consideration, especially in the context of educational institutions that often operate under limited technology budgets. Fortunately, the platform was developed using cost-efficient, open-source, and freemium technologies, which drastically reduced both development and deployment costs.

Major economic determinants are:

1. **Zero Licensing Cost:** All the key softwares (Android Studio, Firebase, Figma) are either open-source or offer generous free tiers suitable for academic applications.
2. **No Server Overhead:** Firebase handles all backend hosting and auto-scaling, eliminating the need for purchasing and maintaining physical servers or dedicated hosting services.
3. **Minimal Hardware Requirement:** The application runs on standard Android smartphones, which are already in use by most students and faculty.
4. **Remote Development and Testing:** Cloud and local machines were utilized by developers emulators, which remove the need for expensive development hardware or spacelabs.

Overall, EduBridge was developed with minimal financial input.

##### 2.1.3 Operational Feasibility

The operational viability of EduBridge lies in the fact that it can be integrated into current student, faculty, and administrative user habits without disruption and massive retraining.

1. **User-foccused Interface:** The application is intended for first- and second-year students and features a minimalist, natural-fitting, and user-friendly UI in line with mobile app design standards.
2. **Minimum Training Requirements:** Onboarding guides and manuals allow users, for example, club coordinators and mentors, to set up profiles and events with no or minimal outside training.
3. **Role-Based Access:** Student, mentor, and admin role-specific dashboards and permissions simplify workflows and deliver context-oriented task management.
4. **Accessibility:** Android compatible, with future inclusion of features like large font and high contrast themes for greater accessibility.
5. **Real-Time Communication:** Immediate sending of alerts, event notices, and announcements strengthens information flow across the institution.

By aligning its modules with actual academic activities—mentoring, event planning, and club management—EduBridge enhances institutional effectiveness and socialization. Its seamless, functional, and scalable fit into a daily academic routine justifies its excellent operational feasibility.

#### Software Requirement Specification

Software Requirements Specification (SRS) defines all the functional and non-functional requirements of the EduBridge platform. This is a full guide throughout the design, development, and testing stages to place the system into compliance with user specifications and institutional goals.

##### Data Requirement

EduBridge demands a very scalable, performance-driven, and fault-tolerant data storage, retrieval, and updating mechanism for various types of structured data including academic activity, user behavior, and institution administration. The data must be safe, readily accessible, and support real-time synchronisation across devices and interfaces. The platform

structure must support multiple data types, provide high-frequency interactions, and determine data integrity for all services.

Some of the most critical components of the EduBridge model include user data, event data, notification, feedback, and authentication data. These are the basic building blocks of the platform's functionality to ensure seamless transactions among students, mentors, and administrators. Effective and proper data modeling is required to manage maximum performance and scalability for long-term development.

##### User Profiles

User profiles are at the core of the data model to the platform. User profiles contain the different participants, including students, mentors, and administrators, and contain required data in order to tailor the EduBridge experience. User profiles contain a number of critical information fields:

* + - * 1. **Personal Information:** These are the basic user information such as name, age, gender, school, contact information, and cover picture. In the case of students, this will also include their year of study, course of study, and interests. Professional qualifications, years of teaching experience, and areas of expertise can be included for mentors.
        2. **Role Identifiers:** You are assigned a particular role (student, mentor, admin),It determines the level of access, functionality, and permissions within the platform. Role-based access control (RBAC) allows only permitted users to accesssensitive data or carry out prohibited actions.
        3. **Event History:** The event history field captures the user's history in past events,both hosted and attended. For students, these could be a seminar,workshops, hackathons, and culture events. Mentors shall keep a record ofactivities that they have organized or led.

Mentors will have a record of the events that they have created or mentored. Admins can monitor all event interactions so that event information is up-to-date and that it reflects the user's actions.

* + - * 1. **Preferences:** User preferences regarding notifications, event topic, mentor recommendations, and other platform settings are saved. This enables the platform to suggest customized content to the user based on interest, boosting engagement and satisfaction.

These profiles are stored in a NoSQL structure, where each user profile is a document in a collection. This allows for flexibility in coping with varied data formats and attributes in different user roles. The platform will be storing this data in the Firebase Realtime Database, which facilitates real-time synchronization and optimal updates. The platform will be storing this data in the Firebase Realtime Database, which facilitates real-time synchronization and optimal updates.

##### Event Details

EduBridge EduBridge takes campus life to your doorstep by hosting a wide variety of events ranging from academic workshops and seminars to friendly contests and cultural events. In order for such events to be easy to locate, sign up for, and administer, the site tracks some event data that makes everything run smoothly:

* + - * 1. **Event Title:** The name of the event shown in listings. It’s designed to be short, clear, and informative so students can quickly understand what the event is about.
        2. **Event Description:** This is the event’s “about” section, providing a full overview including the goals, topics covered, speakers or hosts, and any other important details students need before deciding to attend.
        3. **Event Date and Time:** Each event has a scheduled date and time. EduBridge handles time zones to ensure you never miss a beat whether attending virtually or in person.
        4. **Event Category:** Events are sorted into types such as technical, cultural, social, academic, and more. This categorization helps users quickly filter and find events matching their interests.
        5. **Event Location:** Events are also categorized by location or format (in-person, virtual), making it easier for students to filter through and find events that suit their preferences.
        6. **Associated Club or Department:** Each event may be affiliated with a specific academic department, student club, or faculty member. This data helps in tracking

event ownership and connecting the event to relevant academic or extracurricular units within the institution.

This data is structured in a hierarchical format, with parent entities (such as the event) containing nested sub-entities (such as the event’s categories, speakers, and resources). The event data is stored in Firebase Realtime Database, ensuring smooth synchronization of changes across all active users.

##### Notifications and Announcements

The notifications and announcements system plays a critical role in keeping users informed about upcoming events, platform updates, deadlines, and other important information. Notifications are triggered in real-time or scheduled to be pushed to users based on various factors. The notification system will include:

* + - * 1. **Message Content:** This includes the text, images, or links sent to users, providing updates about events, promotions, deadlines, etc. Content must be clear, concise, and formatted for easy reading.
        2. **Target Audience:** Notifications can be targeted to specific user groups based on roles (students, mentors, or admins), event participation, interests, or even geographic location. Admins will have the ability to send broad announcements or customized messages to specific user segments.
        3. **Real-Time Triggers:** Notifications can be sent based on real-time events, such as when a new event is added, a user registers for an event, or a change occurs to a user’s profile (like event cancellation or venue change). These triggers should be efficient and timely to ensure that users are always updated.
        4. **Manual Publishing:** Admins and authorized personnel can manually publish announcements, which may include upcoming events, holidays, policy changes, or other institutional information. These should be most visibly and prominently shown in user feeds.

The real-time synchronization feature provided by Firebase ensures the notifications are delivered in real-time to users irrespective of the device and location.

##### Feedback and Ratings

User feedback is a critical component of the EduBridge platform, enabling administrators and mentors to improve the quality of events and user experience through feedback. The feedback is collected from students to mentors, events, and platform usage as a whole. The feedback system collects the following details:

* + - * 1. **Event Feedback:** Students can post reviews and ratings for events they have attended, giving useful feedback on the relevance, effectiveness, and quality of the event. Ratings can be in the form of numbers (1-5 stars) with explanatory written comments. This feedback assists subsequent event organizers in refining their provision.
        2. **Mentor Feedback:** Students' feedback about their mentors may be considered to monitor the effectiveness of their teaching, communication skills, and overall performance. The feedback may impact the mentor's involvement in future events and programs.
        3. **User Satisfaction Indicators:** Beyond event-specific feedback, the platform will also collect general user satisfaction indicators. These indicators can be in the form of surveys, Net Promoter Score (NPS) measures, or direct feedback on the usability and functionality of the platform.

All the feedback data is stored with security and processed with predefined procedures to generate insights that contribute to the quality of education, mentorship, and the overall user experience, respectively.

##### Authentication Data

The authentication information is essential in safeguarding the user accounts and ensuring that sensitive information on the EduBridge platform is secure. The user credentials are stored securely through the assistance of Firebase Authentication, which offers strong mechanisms for handling user authentication and session. The authentication data items that are required are:

* + - * 1. **Credentials:** The username, email, and password are included. Passwords and other sensitive data are hashed and stored securely to avoid unauthorized access.
        2. **Session Tokens:** Every active user session has a unique session token, which is utilized to handle sessions. Session tokens are reliably stored so that the users remain authenticated during their use of the platform.
        3. **Two-Factor Authentication (2FA):** To integrate security features, EduBridge may adopt 2FA mechanisms that require users to verify themselves using a second method (e.g., a tel. no. or email address).

Firebase Authentication offers a safe and scalable system for handling authentication data, including email verification, password reset, and secure session storage.

##### Data Storage Architecture

All user data that backs EduBridge—events, notifications, and user profiles—is securely stored in the Firebase Realtime Database. Such a structure is especially ideal for applications such as EduBridge, which need speed, real-time data, and seamless synchronization across multiple devices. Firebase's NoSQL configuration, flexible and JSON-based, has the data structure that can expand and develop comfortably. New features or modification of existing features can be achieved without affecting other aspects of the app.

The real magic lies in the real-time synchronization features of Firebase: whenever any change—such as creating a new event or updating a user profile—happens, it gets reflected on all the connected devices instantly, without having to wait or refresh. This setup keeps EduBridge responsive and fast even with a growing number of users. Its hierarchical data model also facilitates the aggregation of correlated data, thus ensuring that users always get to experience a seamless and streamlined affair.

##### Functional Requirement

The EduBridge core functions encompass all the basic activities related to student engagement and platform management. The main functional requirements are:

##### User Registration and Login

Secure registration using Firebase Auth via email-password or institutional IDs. Assigning role at registration (student, mentor, admin).

##### Personalized Dashboards

* 1. Personalized dashboards that adapt to individual users' roles and interests, showing relevant events, news, and recommendations.

##### Event Registration and Tracking

Students can see detailed event information, sign up in one click, and monitor their signed-up and finished activities.

##### Mentor Search and Interaction Interface

A listable pool of mentors by subject matter, ratings, and past interactions. Mentees can request mentorship or schedule sessions.

##### Club/Society Profile Management

Clubs are responsible for their own profiles, post event calendars, and post updates viewable by followers or campus-wide users.

##### Push Notifications and Alerts

Immediate alerts for updates on events, deadlines, and official news are sent through Firebase Cloud Messaging (FCM).

##### Admin Panel

A core dashboard to authenticate new users, handle backend data, post notifications, and monitor engagement metrics.

These conditions ensure that EduBridge achieves its core objective of promoting visibility, accessibility, and engagement of organisational activities.

##### Performance Requirement

Performance specifications are intended to ensure responsiveness, availability, and consistency across all user interactions.

1. The application should **load fully within 2–3 seconds** on mid-range Android devices.
2. **Dashboard elements** and event listings should populate in under 1 second using real- time fetch operations.
3. Firebase’s cloud infrastructure enables **high concurrency**, allowing simultaneous access without latency drops.
4. The app should handle **spikes in traffic**, especially around event registration or exam seasons, without performance degradation.

##### Maintainability Requirement

EduBridge is developed with maintainability as a central principle, ensuring that future updates, patches, and feature expansions can be incorporated with minimal disruption.

1. **Modular Codebase**: Components such as user authentication, event modules, and notifications are built independently for easy debugging and upgrades.
2. **Firebase Schema Flexibility**: The JSON structure allows rapid schema modifications and new node additions.
3. **Version Control via GitHub**: Source code is versioned and collaboratively managed for traceability and rollback.
4. **Real-time Logging and Crash Reports**: Firebase Crashlytics enables live monitoring of app stability and helps identify runtime bugs.

##### Security Requirement

Given the sensitivity of user data and academic records, EduBridge implements robust security measures:

1. **Secure Authentication**: Firebase Authentication supports hashed password storage and secure session handling.
2. **Role-Based Access Control (RBAC)**: Ensures that each user type (student, mentor, admin) can only access permitted resources.
3. **Database Security Rules**: Firebase’s built-in rule system prevents unauthorized read/write operations at the node level.
4. **Encrypted Communication**: All data transmitted between the client and Firebase is encrypted using HTTPS.
5. **Input Validation and Sanitization**: Prevents injection attacks and enforces data integrity.

##### SDLC Model Used

EduBridge was developed using the **Agile Software Development Life Cycle (SDLC)** model. Agile was selected due to its adaptability to changing requirements, focus on user feedback, and iterative nature.

Key phases included:

##### Requirement Gathering

Initial analysis of target user pain points, institutional workflows, and stakeholder expectations.

##### Iterative Design and Prototyping

Use of Figma for wireframing and interactive mockups. Multiple UI iterations based on test group feedback.

##### Development in Sprints

Features were developed in 2-week sprints using GitHub for source control and sprint planning.

##### Continuous Integration and Testing

Each module was tested independently and integrated gradually to identify and resolve dependency issues.

##### Final Deployment and Maintenance

A stable APK was generated for internal testing, with future plans for Google Play Store deployment.

Agile enabled faster decision-making, real-time stakeholder collaboration, and adaptive feature tuning based on student needs.

#### System Design

##### Data Flow Diagrams

Data Flow Diagrams were designed at multiple levels to represent the flow of data through EduBridge:

##### Level 0 – Context Diagram:

Shows the system as a whole with external entities (users, clubs, Firebase) interacting via defined interfaces (registration, dashboard, events).

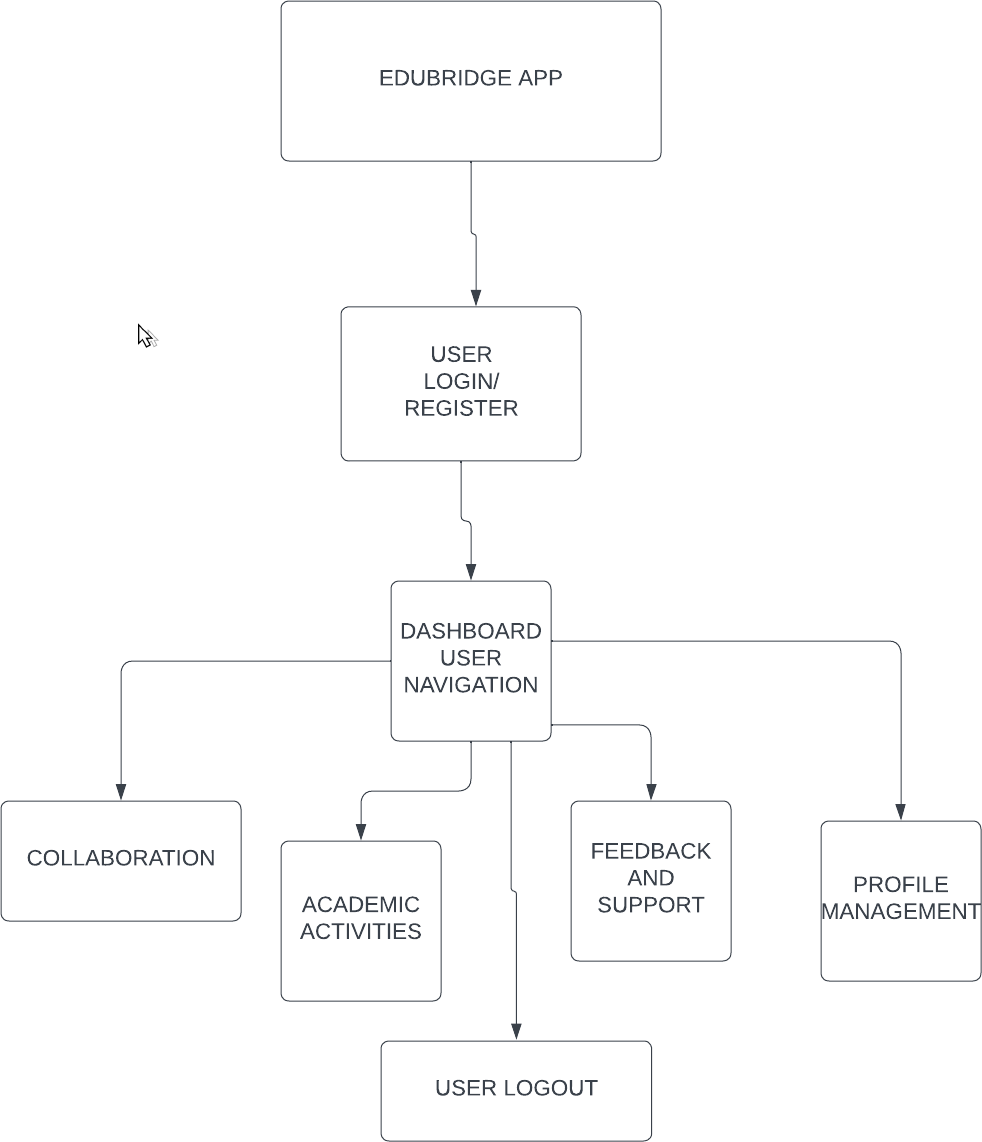
##### Level 1 – Process Breakdown:

* 1. *Process 1*: Registration/Login System
  2. *Process 2*: Event Browsing and Registration
  3. *Process 3*: Mentor Search and Feedback
  4. *Process 4*: Notifications and Updates

##### Level 2 – Detailed Transactions:

* 1. Includes operations like CheckCredentials, FetchUpcomingEvents, SubmitFeedback, UpdateClubDetails.

These diagrams guide developers on module interconnectivity and form the basis for database structuring and logic control.



##### Figure 4.1: Data Flow Diagram

* + 1. **Use Case Diagrams**

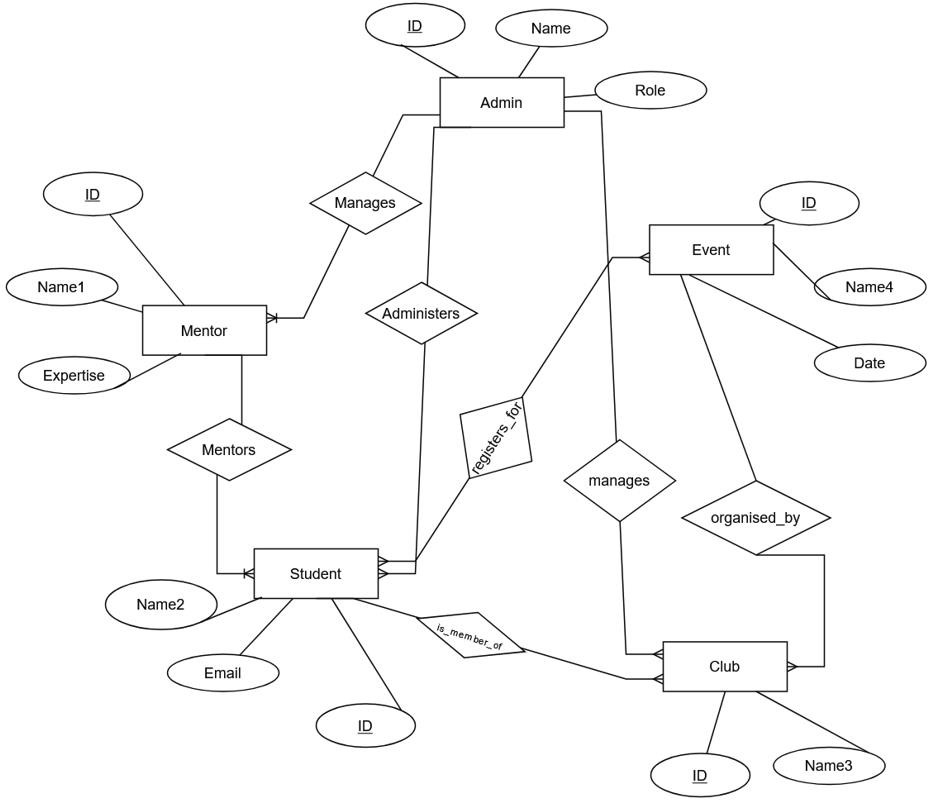
Use case diagrams visually represent the system functionalities and the interactions between actors and the system.

##### Actors:

1. **Student**: Can register, log in, view dashboard, register for events, give feedback, and follow clubs.
2. **Mentor**: Can update profile, manage mentees, and respond to mentorship requests.
3. **Admin**: Can approve events, verify clubs, broadcast announcements, and analyze app usage.

These diagrams help clarify functional boundaries and support role-based user interface design.

##### ER Diagram

****

**Figure 4.2: ER Diagram**

The Entity-Relationship (ER) diagram models the data structure and relationships among entities in the system. Key entities include:

1. User (user\_id, name, email, role)
2. Event (event\_id, title, date, host\_club)
3. Club (club\_id, name, description)
4. Mentorship (mentor\_id, student\_id, rating)
5. Notification (notif\_id, user\_id, message)

##### Relationships:

1. A *user* can register for multiple *events*
2. A *club* can host multiple *events*
3. A *mentor* can be assigned to multiple *students*

This visual structure aids in backend schema creation and real-time data binding.

##### Database Design

EduBridge uses **Firebase Realtime Database**, a cloud-hosted NoSQL database that stores data as one large JSON tree. Key collections (or nodes) include:

##### /users

Stores all user information including names, roles, email addresses, preferences, and history.

##### /events

Contains all active and past event details such as name, category, organizer, and participant lists.

##### /clubs

Includes data about societies or clubs with their events, admin users, and social feeds.

##### /notifications

Houses system-generated and admin-published notifications linked to event reminders and updates.

##### /feedback

User-submitted feedback and star ratings for mentors or events.

Firebase’s real-time sync ensures that all users have the most recent data without needing to refresh or manually update views. Its scalability supports growth across departments, institutions, or even nationwide deployments in the future.

# CHAPTER 5 IMPLEMENTATION

##### Introduction: Tools and Technologies Used

The implementation of **EduBridge** was guided by the objective of building a scalable, efficient, and secure mobile application that enhances academic engagement in educational institutions. To achieve this, a carefully curated **technology stack** was adopted that balances development agility with production-level performance.

The choice of tools and platforms was primarily influenced by the need for:

1. Seamless integration across system components
2. Real-time data synchronization
3. Low development cost
4. Device compatibility
5. Ease of future maintenance and scaling

The entire system was architected using **Google Firebase** as the backend and **Android Studio** as the development environment. Below is a breakdown of the tools and technologies used in the implementation process:

##### Android Studio (Integrated Development Environment)

**Purpose:** Primary IDE for building, testing, and deploying the mobile application

##### Reason for Selection:

* + - 1. Official support for Android development
      2. Rich set of features including emulator testing, code suggestions, and version control integration
      3. Built-in Gradle support for managing dependencies and building APKs

##### Java (Programming Language)

**Purpose:** Backend logic, app structure, and interaction with the Firebase backend

##### Reason for Selection:

* + - 1. High compatibility with Android SDK
      2. Extensive library support
      3. Strong community and documentation
      4. Supports object-oriented programming, ensuring clean and modular code

##### XML (User Interface Markup Language)

**Purpose:** Designing static UI elements such as layouts, widgets, buttons, and input forms

##### Reason for Selection:

* + - 1. Standard language for Android layout design
      2. Separates design from logic for better maintainability
      3. Highly customizable and compatible with design tools like Figma

##### Firebase Realtime Database

**Purpose:** Backend database for storing and syncing data in real time

##### Reason for Selection:

* + - 1. NoSQL, JSON-based cloud-hosted database
      2. Enables instant updates across devices without polling
      3. Ideal for mobile-first applications requiring live data sync

##### Firebase Authentication

**Purpose:** User authentication and role-based access control

##### Reason for Selection:

* + - 1. Secure and easy-to-integrate login system using email/password, phone number, or federated identity providers
      2. Built-in support for session management
      3. Reduces time spent on custom authentication logic

##### Firebase Cloud Messaging (FCM)

**Purpose:** Sending push notifications for reminders, alerts, and announcements

##### Reason for Selection:

* + - 1. Enables timely communication with users even when the app is closed
      2. Supports topic-based or user-specific notifications
      3. Integrates seamlessly with Android apps

##### Firebase Crashlytics

**Purpose:** Real-time crash reporting and issue tracking

##### Reason for Selection:

* + - 1. Provides stack traces and user metrics to identify bugs quickly
      2. Helps improve stability by resolving high-impact issues based on priority
      3. Integrates with Google Analytics for deeper insights

##### Figma (UI/UX Design Tool)

**Purpose:** Wireframing, prototyping, and collaborative UI design

##### Reason for Selection:

* + - 1. Web-based design platform for creating responsive and modern UI mockups
      2. Facilitates real-time collaboration and feedback during the design process
      3. Output can be easily translated into Android XML layouts

##### Git and GitHub (Version Control System)

**Purpose:** Code versioning, team collaboration, and issue tracking

##### Reason for Selection:

* + - 1. Enables collaborative coding and feature branching
      2. Facilitates safe rollbacks and code reviews
      3. Helps in maintaining a clean, versioned development workflow

##### Android Virtual Device (AVD) Emulator

**Purpose:** Simulated environment for testing the app on various Android devices

##### Reason for Selection:

* + - 1. Allows testing across different screen sizes and OS versions
      2. Helps debug app behavior without needing physical devices
      3. Integrated directly with Android Studio

##### Platform Compatibility & Deployment Target

* + - 1. **Target OS:** Android 5.0 (API Level 21) and above
      2. **Target Devices:** Smartphones and tablets used by students, mentors, and faculty
      3. **Current Deployment Mode:** Internal testing via APK installation
      4. **Future Deployment:** Planned release on Google Play Store with institutional onboarding

This technology stack not only empowered the team to develop the application rapidly but also ensured a strong foundation for future updates, scalability, and real-time performance. The use of cloud-based infrastructure (Firebase) combined with native Android development practices allowed EduBridge to meet both technical expectations and user experience goals.

* 1. **Technologies and Tools Used:**

**Table 5.1: Technologies and Tools Used**

|  |  |  |
| --- | --- | --- |
| **Category** | **Tool / Technology** | **Purpose** |
| Programming  Language | **Java** | Backend logic and integration in Android  development |
| Markup Language | **XML** | Designing UI layouts and app structure |
| IDE / Development  Tool | **Android Studio** | Main environment for building and  testing the mobile app |
| Backend Service | **Firebase Realtime**  **Database** | Real-time data storage and retrieval |

|  |  |  |
| --- | --- | --- |
| Authentication | **Firebase Authentication** | Secure login and user management |
| Cloud Messaging | **Firebase Cloud**  **Messaging (FCM)** | Sending real-time notifications to users |
| UI/UX Design | **Figma** | Designing wireframes and interactive  user interface prototypes |
| Testing &  Debugging | **Robolectric, Firebase**  **Crashlytics** | Unit testing and crash/error reporting |
| Version Control | **Git & GitHub** | Managing source code and collaborative  development |

* 1. **Platform Details:**

The implementation of EduBridge was specifically tailored for mobile devices used in academic environments. By aligning the development platform with user behavior—where students and faculty predominantly interact through smartphones—the project ensures accessibility, usability, and scalability from the outset.

The key platform specifications are as follows:

##### Target Platform: Android (API Level 21 and Above)

EduBridge is developed for the Android operating system, beginning with API Level 21 (Lollipop), which ensures compatibility with a wide range of devices still in use by students. This decision balances modern UI/UX capabilities with broad accessibility, including support for low-to-mid-tier devices typically used in campus settings.

Android was chosen for its:

* 1. Extensive user base in the Indian education ecosystem
  2. Open-source foundation
  3. Flexible UI design system
  4. Tight integration with Google Cloud services, especially Firebase

##### Device Compatibility: Mobile Phones with Android OS

The application is optimized for Android smartphones and tablets, focusing on devices with at least 1.5 GB RAM and Android 5.0+. The UI is responsive and designed using constraint layouts and flexible components to support different screen sizes and orientations.

This ensures that:

* 1. Students can access EduBridge on personal devices
  2. Faculty and mentors can use the same interface for event updates and feedback
  3. Institutions need not invest in dedicated hardware for app access

##### Database Format: JSON-based Firebase Structure

EduBridge uses Firebase Realtime Database, which organizes all data in a hierarchical JSON tree format. This design allows for:

* Real-time data syncing across all users
* Easy data querying and filtering
* Efficient bandwidth usage
* Flat learning curve for future developers and admins

Data such as user profiles, event listings, announcements, and feedback are stored in structured JSON nodes, enabling precise role-based access through Firebase security rules.

##### Deployment: Debug APK for Internal Testing, Play Store-ready Package for Release

During the development and testing phases, the app was distributed in the form of a debug APK to selected student testers and institutional reviewers. This facilitated iterative feedback collection and usability testing.

Future deployment goals include:

* 1. Preparing a signed APK for production use
  2. Compliance with Google Play Store standards
  3. Institutional distribution via QR codes or admin-assisted onboarding

The app is modular and adaptable, allowing for multi-institution deployment through configuration without altering core logic.

# CHAPTER 6 TESTING AND MAINTENANCE

##### Testing Techniques and Test Cases Used

To ensure that **EduBridge** functions smoothly, securely, and as intended, a **comprehensive testing strategy** was implemented throughout the development life cycle. The testing approach followed a multi-layered methodology, combining **manual**, **automated**, and **real- time monitoring** techniques to uncover bugs, validate functionalities, and optimize the user experience.

The testing process covered every stage of the system—from individual functions (unit tests) to fully integrated workflows (integration tests), as well as the user interface and real-time system behavior in live environments.

##### Testing Techniques Used

* + - 1. **Black Box Testing**

Black Box Testing was applied to assess the system’s outputs without considering its internal code structure. Testers input various valid and invalid data scenarios to evaluate how the system responds from the user’s perspective.

##### Examples of Black Box Testing in EduBridge:

* + - * 1. Logging in with correct vs. incorrect credentials
        2. Registering for an event with and without internet access
        3. Submitting incomplete mentor search queries
        4. Navigating to a dashboard with no upcoming events

This technique was effective in identifying unexpected behaviors in high-traffic functionalities like registration and login.

##### Unit Testing

Unit Testing was used to evaluate individual code modules for accuracy and logic correctness. It was primarily conducted using **Android JUnit** and **Robolectric**, allowing for isolated testing of core components in a simulated Android environment.

##### Key areas tested:

* + - * 1. Event registration logic
        2. User role verification
        3. Recommendation engine output
        4. Firebase database write/read operations
        5. By validating each component in isolation, developers ensured reliability and modularity before combining them into larger workflows.

##### Integration Testing

After unit tests confirmed module-level integrity, **Integration Testing** was performed to ensure that various modules communicate correctly and work cohesively.

##### Examples include:

* + - * 1. A new user signing up, getting role-based dashboard data from Firebase, and registering for an event
        2. Mentor updates triggering dashboard refresh for interested students
        3. Admin posts triggering user-side notifications and backend log updates
        4. These tests helped reveal issues with data dependencies and flow inconsistencies that only arise when modules interact.

##### UI Testing

User Interface (UI) Testing focused on the **visual layout, responsiveness, and usability** of the application. It was performed manually on physical Android devices and emulators, as well as through automated UI tests using Android's Espresso framework (for future implementation).

##### Elements tested:

* + - * 1. Button placements, color contrasts, and font readability
        2. Navigational flow between modules
        3. Layout responsiveness on different screen sizes and resolutions
        4. Error handling messages and feedback alerts
        5. UI testing ensured that EduBridge remained intuitive and visually accessible across devices.

##### Firebase Crashlytics Monitoring

To simulate and handle real-world issues that might escape manual or automated testing, **Firebase Crashlytics** was integrated into the application for real-time crash reporting.

##### Benefits:

* + - * 1. Captured rare crash scenarios with full stack traces
        2. Flagged performance bottlenecks and out-of-memory issues
        3. Highlighted user actions preceding a crash, enabling accurate debugging
        4. Allowed developers to prioritize fixes based on issue frequency and severity
        5. Crashlytics played a crucial role during beta testing by providing quick feedback from early users and identifying device-specific issues not present in emulator environments.

##### Conclusion of Testing Phase

The combined application of these techniques ensured that EduBridge meets the following quality objectives:

* + - * 1. Functional accuracy across modules
        2. Seamless data flow and module interaction
        3. Smooth and intuitive UI/UX
        4. Error-resilient performance in real-world use cases
        5. Scalable and maintainable backend infrastructure
        6. The successful completion of the testing phase laid a strong foundation for stable deployment and future expansion of the platform.
    1. **Sample Test Cases**

**Table 3.1: Sample Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **Case ID** | **Test Scenario** | **Input** | **Expected Result** | **Status** |
| TC001 | Login with valid  credentials | Email, Password | Redirect to dashboard | Pass |
| TC002 | Login with invalid  credentials | Wrong Email, Wrong  Password | Show error message | Pass |
| TC003 | Event registration | Event ID, User ID | Success message,  update in database | Pass |
| TC004 | Push notification  trigger | Event scheduled | Notification delivered  to user device | Pass |
| TC005 | Mentor search and  filter | Subject: Java | Display list of Java  mentors with ratings | Pass |
| TC006 | Club admin creates  new event | Title, Description, Date | Event visible to all  users in feed | Pass |
| TC007 | Firebase database  security rule validation | Direct write attempt by unauthorized user | Access denied | Pass |

* 1. **Bug Tracking and Resolution**

Proper bug tracking and fixing is a foundation of any quality software development process. EduBridge followed a structured and dynamic approach to detect, document, prioritize, and correct defects at each stage of the project lifecycle.

##### Bug Tracking System: GitHub Issues

The development team utilized GitHub Issues very heavily as the main method of documenting tasks and bugs.tracking. All problems were tracked with the following metadata:

* + - 1. **Issue Title**: Concise summary of the bug (e.g., "Event not saving after registration")
      2. **Description**: Detailed explanation of the behavior, expected outcome, and steps to reproduce
      3. **Labels**: Classified as bug, enhancement, UI, backend, etc.
      4. **Priority**: Classified as critical, high, medium, or low based on the level of influence and urgency.
      5. **Assignee**: Developer responsible for resolution
      6. **Status**: Updated as Open, In Progress, Resolved, or Closed

This structure facilitated good communication, effective coordination between team members, and fix traceability throughout sprints.

##### Sprint-Based Resolution Workflow

Bugs were reviewed and addressed iteratively during development sprints, each of which lasted approximately 1–2 weeks. Outstanding bugs from the last iteration were examined and prioritized along with new feature development at the start of each sprint.

Resolution steps usually included:

##### Reproduction and Verification

Confirming the error's occurrence on different devices or environments.

##### Root Cause Analysis

Logging review, stack traces (through Firebase Crashlytics), and affected modules

##### Code Fix and Refactoring

Making corrections while ensuring existing functionality remains intact

##### Peer Review

Submitting a pull request (PR) for review and approval via GitHub

##### Regression Testing

Re-testing to confirm that the bug has been resolved without side effects

##### Firebase Crashlytics for Real-time Error Monitoring

To complement manual and automated testing, **Firebase Crashlytics** was integrated for **live crash tracking and performance monitoring**. This allowed the team to:

* + - 1. Receive instant alerts when the app crashed on any user’s device
      2. Access detailed crash logs, device specs, OS versions, and user sessions
      3. Prioritize high-impact bugs by frequency and user count
      4. Identify patterns such as recurring crashes on specific Android API levels

This proactive system enabled rapid patching of runtime errors and significantly improved post-deployment stability.

##### In-App Feedback Forms

During the internal testing phase, early users (testers from the institution) were encouraged to submit usability feedback and report bugs directly from within the app. A feedback module was included that allowed users to describe the issue, attach screenshots, and rate their experience.

This direct feedback loop helped uncover:

* + - 1. UI inconsistencies on certain devices
      2. Broken links or inaccessible features
      3. Misaligned expectations on functionality

Such qualitative data was invaluable in enhancing user experience and polishing the app prior to launch.

##### Post-Deployment Maintenance Readiness

EduBridge has been structured for **ongoing maintenance** through:

* + - 1. Well-documented code for future handoffs
      2. Modular components for easy debugging and upgrades
      3. Firebase analytics and logs for continuous performance evaluation

To keep development organized and on track, EduBridge uses GitHub project boards for roadmap planning and version control. This ensures the team stays coordinated and responsive as the platform evolves. Through thoughtful planning and the use of flexible tools, EduBridge remains robust, user-centric, and adaptable to future academic needs—ready to grow alongside students and institutions.

# CHAPTER 7 RESULTS AND DISCUSSION

##### Description of Modules with Snapshots

EduBridge has been created with modularity, ease of use, and expandability in consideration. Every feature has been engineered to serve a unique purpose in the academic community, whether it is finding events, establishing mentorship relations, or administrative control. The succeeding modules explain the major modules of the system, with real-time screenshots that will be integrated during the final formatting process.

##### Welcome and Onboarding Screens

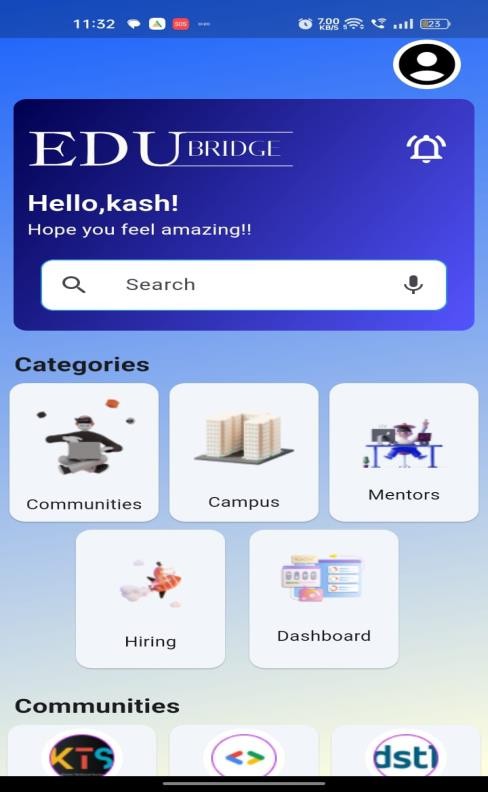
When a user opens EduBridge for the first time, he/she is presented with a set of nice-looking onboarding screens that resemble a warm welcome rather than a typical tutorial. These screens provide users with a quick and welcoming introduction to the application's fundamental features—such as discovering mentors, participating in communities, and registering for events.



##### Figure 7.1: Welcome Screen

* + 1. **User Dashboard/Communities**

After successful authentication, the users are redirected to a customized dashboard based on their own role, whether that of student, mentor, or administrator.



##### Figure 7.2: Home Screen

* + - 1. For students, the dashboard emphasizes
      2. Upcoming events and current registration status
      3. Recommended mentorship opportunities
      4. Club news and announcements
      5. Monitoring and feedback of session attendance progress

The content is presented in an easy-to-understand and user-friendly manner, allowing for student organization and participation without any challenges.

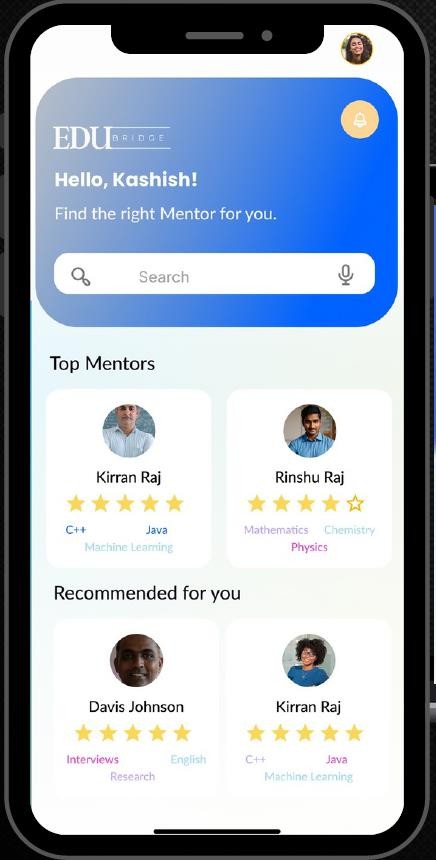
* + 1. **Mentor Discovery Module**

A prominent characteristic of EduBridge is its mentor discovery interface, which enables students to readily:

* + - 1. Search for mentors using subject tags like C++, AI, or Physics
      2. View mentor ratings and areas of expertise
      3. Send mentorship requests or instantly start a session

This function facilitates the development of robust support systems and challenges students to learn from experts from different fields..

.



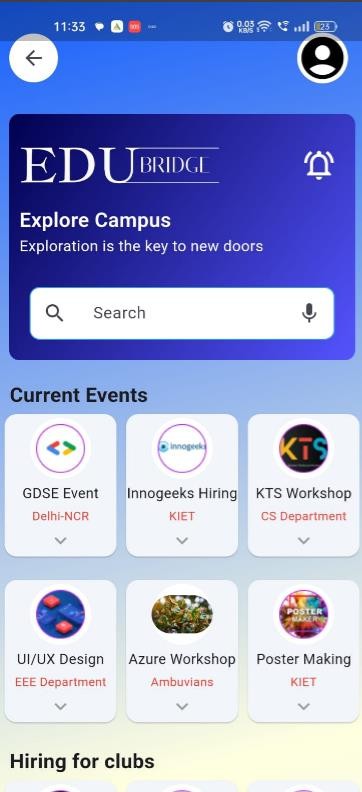
##### Figure 7.3: Mentor Module

* + 1. **Event Management**

Management Students are provided with a well-organized list of academics and extracurricular activities promoted by clubs or faculty. Each example includes:

* + - 1. The event name and a brief description
      2. Host data and calendar
      3. Registration or membership welcoming buttons
      4. Cues and reminders to remain goal-focused

This structure enhances involvement and enables activity planning to be simpler and more organized.



##### Figure 7.4: Event Module

* + 1. **Admin Panel (not shown in user-facing UI)**

Although not accessible to everyone, EduBridge features a secure administration console for club leaders, faculty, and coordinators. Admins here can:

* + - 1. Approve or decline new event submissions
      2. Manage user roles and profiles
      3. Send announcements to the community
      4. Monitor engagement stats and insights

This backend ensures smooth management and keeps the platform organized.

##### Key Findings of the Project

Following several iterations of development, testing, and internal deployment, some significant outcomes were realized. emerged:

1. Greater Involvement – Putting all news and events in one place led to far more participation in university life.
2. Simplified Mentor Connections – Personalized mentor recommendations enabled students to conveniently identify relevant experts, thus increasing the specificity and efficacy of learning support.
3. Reduced Information Overload – Centralized communication relieved the chaos of posters, and individual social media groups..
4. User-Friendly Design Wins – The clean, minimalist design had the ease of use preferred by both students and instructors to begin utilizing the app without training..
5. Smooth Real-Time Updates – With Firebase, updates are performed in real-time with minimal lag—even on standard smartphones.

These results validate the decision-making in developing and illustrate EduBridge's suitability in contemporary virtual academic space.

##### Brief Description of Database with Snapshots

EduBridge takes advantage of Firebase Realtime Database, which is NoSQL cloud-hosted

a database which stores all system data in a structured JSON form. This choice was intentional, thanks to Firebase’s excellent scalability, real-time syncing capabilities, and smooth compatibility with Android platforms. These features are essential to support EduBridge’s mission of providing a seamless, responsive user experience across a wide range of devices and network conditions.

The database is organized into several key collections, each designed to manage specific types of data critical to the platform’s operations:

##### Key Database Collections:

1. **/users**

Holds metadata about student clubs and societies, including:

* 1. Club names and descriptions
  2. Lists of administrators
  3. Associated events
  4. This enables smooth club management and helps maintain an active campus community.

##### /events

This collection holds detailed information about each event, including the event title, hosting organization, date and time, a clear description, relevant tags for easy categorization, and a list of registered participants. It helps users find events, register smoothly, and track attendance.

##### /clubs

This contains data about student clubs and societies, such as their name, description, administrators, and related events. It enables easy management of clubs and keeps the campus community engaged and informed.

##### /notifications

Handles both scheduled and real-time push notifications sent to users’ devices. It ensures:

* 1. Prompt notice of occurrences
  2. Notifications
  3. Updates
  4. Keeps users busy and engaged and informed without overwhelming them.

##### /feedback

Collects:

* 1. Mentor reviews
  2. Session ratings
  3. Summaries presented by students
  4. Such a feedback loop ensures quality mentorship interactions and fostersongoing platform improvement.

# CHAPTER 8 CONCLUSION AND FUTURE SCOPE

##### Conclusion

EduBridge's successful development and in-house implementation are a big step forward. promoting technology adoption in learning environments. The system fills a vast gap in tertiary education through the inclusion of an integrated framework that not only governs scholarly information and also supports active participation, mentoring, and teamwork. EduBridge is designed with maximum care to be student-focused but institution-aligned objectives, consolidating a number of previously separate processes into one single integrated mobile platform.

By providing centralized access to event calendars, directories of mentors, society postings, and tailored dashboards, the app shatters communication barriers and builds a more interconnected academic community. From a technical perspective, EduBridge is constructed with a resilient and adaptable technology framework that includes Android Studio, Java, and Firebase. This trio allowed the team to utilize real-time sync of data, role-based authentication, and cloud-based push notifications — all essential to a modern mobile-first learning experience. Centralized access to event calendars, mentor directories, society announcements, and personalized dashboards, the software diminishes communication barriers and promotes a more integrated academic community. From a technical perspective, EduBridge is constructed utilizing a resilient and adaptable technology framework that includes Android Studio, Java, and Firebase. This combination allowed the team to perform real-time synchronization of data, role-based authentication, and cloud-based push notifications—all essential to mobile-first learning spaces today.

At each developmental stage, special stress was given on:

1. **Usability**: Making sure the interface is intuitive and simple to use, particularly for first-semester students.
2. **Security**: Employing Firebase's database rules and authentication for data integrity and access control.
3. **Performance**: Through the use of real-time database interactions, EduBridge maintains an active and reactive user interface.
4. **Scalability**: Backend architecture and code are designed to allow easy addition of new features, institutions, and users without requiring massive rewrites or systems redesigns in large chunks.

The comments received in the testing process showed that users found the app easy to use, visually appealing, and highly helpful in staying aware and connected in the campus setting. Mentors were pleased with the way the app highlighted their expert knowledge, while students appreciated the ease of finding relevant events and mentorship opportunities brought together under one roof.

In a nutshell, EduBridge transcends the level of a mere academic tool; it is a system that reinvents how students interact with their schools, peers, and educators. Further, it paves the way for mass academic innovation with intelligent, data-driven design and affordable technology solutions.

##### Future Scope

* + 1. **Web Version Deployment**

Enabling access to EduBridge via a web application meant for laptops and desktops will make it more accessible to a broader array of devices. This deployment will allow users to easily access the platform directly through their web browsers, thus offering greater flexibility and a more convenient user experience—especially for those who prefer working with larger screens. The desktop version may also incorporate other features tailored to desktop users like drag-and-drop support, powerful filtering capabilities, and streamlined tools for content creation.

##### Analytics Dashboard

The addition of an integrated analytics dashboard will provide administrators, mentors, and institutions with precious insights into the performance of the platform. Through the aggregation and presentation of event statistics, user behavior, and information

related to mentor-student interactions, EduBridge will facilitate more informed decision-making and the development of customized academic plans. This data-driven approach will allow institutions to monitor patterns of participation, identify areas for enhancement, and optimize the use of resources—leading to a more enriched and effective learning process for all stakeholders.

##### Third-Party Integrations

To enhance the user experience and enable smooth interaction with accessible digital resources, EduBridge aims to integrate widely used third-party features. This includes integration with academic calendars, the inclusion of virtual conferencing tools like Google Meet and Zoom for virtual events, and integration with external Learning Management Systems (LMS) to enrich academic functions. All these integrations should reduce the necessity for manual entry of data, improve accessibility, and enable students and instructors to work within familiar environments—thus making academic management simpler and more effective.

##### Gamification

EduBridge aims to enhance the participation of students and make the learning environment more interactive and competitive through the implementation of a gamification system. By a point-based reward system, students earn points while participating in activities, performing tasks, and interacting with mentors. The earned points can then be traded for rewards, accolades, or academic incentives. By making participation more rewarding and enjoyable, gamification is applied to enhance the levels of motivation and retention, thereby promoting long-term participation among students in their learning community.

##### AI-Based Recommendations

EduBridge will be using artificial intelligence (AI) and machine learning (ML) to drive its personalization feature, making its recommendations smarter and more relevant to its users. Based on user behavior, interests, and interactions data, the platform will be offering personalized recommendations for mentors, events, and learning material. This allows students to discover the most valuable opportunities that align with their academic needs, while also matching educational mentors with students whose goals

and needs are similar to their area. Artificial intelligence-driven recommendations will be increasing the level of personalization and value of the academic experience for every user.

##### 8.1.3 Conclusion:

EduBridge has established a strong foundation for a lively, interactive, and centralized learning community. With future additions such as online deployment, high-end analytics, external service integration, gamification features, and AI-driven personalized features, the platform is poised to grow and develop according to the shifting needs of learners, teachers, and learning institutions. Ongoing development will put EduBridge at the vanguard of the education innovation push, giving users the tools to thrive in the increasingly digital learning environment.

# REFERENCES

[1]. Kenge, Rohit. (2020). A Research Study on the ERP System Implementation and Current Trends in ERP.

[2]. Saxena, A. (2013). Developing entrepreneurship and e-governance in India: Role of common service centers.

[3]. Tella, A., & Bashorun, M.T. (2011). Impact of web portals on e-learning.

[4]. Ambrose, A., Oyelami, O., & Ayo, C. (2008). Development of an E-learning web portal: The FOSS approach.

[5]. Tang, H.L., Chung, S., & Se, C. (2009). Examining the impact of possible antecedents on service usage: An empirical study on Macao e-government.

[6]. Jun, G., & Chen, Y.J. (2011). The integration of college students' employment resources. [7]. Patel, R., & Gupta, S. Node.js: A Platform for Scalable Backend Development.

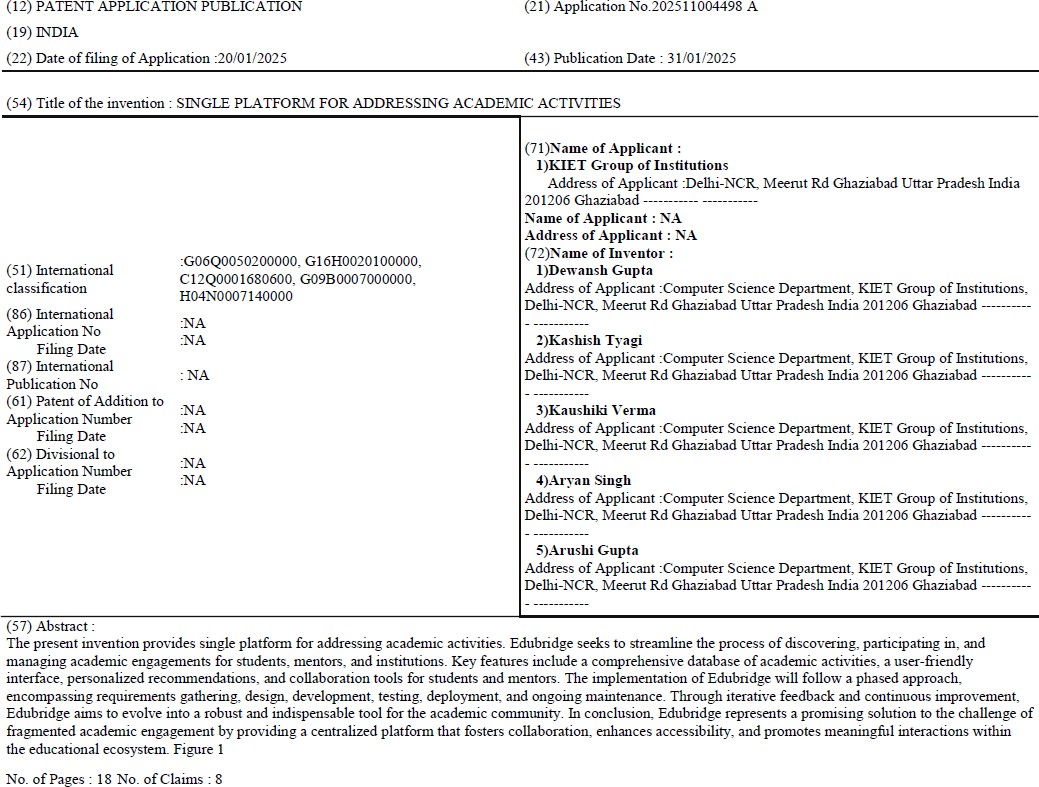
[8]. Al-Mashari, M. (2003). Enterprise Resource Planning (ERP) systems: A research agenda. [9]. Wang, T., & Li, S. (2022). Relationship between employment values and college students' intention of slow employment.

[10]. Gupta, R., & Jain, V. (2021). Design and Development of an Android-based Academic Event Management System. International Journal of Computer Applications, 183(45), 12-18. [11]. Sharma, P., & Verma, S. (2020). Exploring Student Engagement through Educational Mobile Applications. Journal of Educational Technology & Society, 23(3), 67-76. [12]. Singh, K., & Mishra, A. (2019). Enhancing Institutional Communication using Push Notification Services. International Conference on Smart Technologies and Management,2(1),89-95.

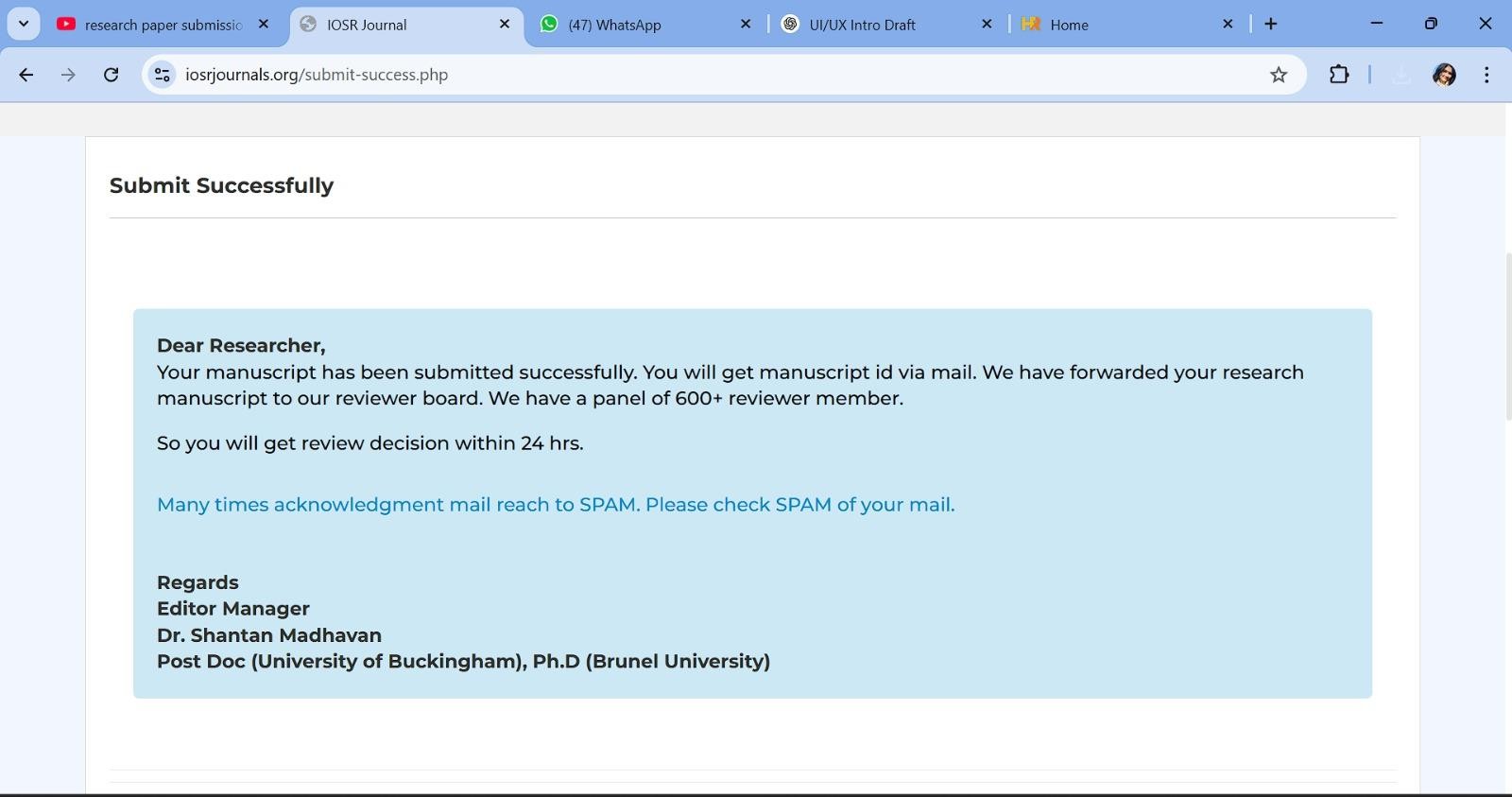
[13]. Chatterjee, P. (2022). A Study on the Impact of Centralized Portals in Indian Higher Education Institutions. Journal of Emerging Educational Technology, 7(2), 33-40. [14]. Dwivedi, A., & Kapoor, M. (2021). Firebase as a Realtime Backend: Benefits and Limitations in Mobile App Development. International Journal of Computer Engineering, 19(1), 110-118.

[15]. Kumar, S., & Raj, A. (2020). Adoption of Agile Methodology in Student-Centric App Development. Proceedings of the 6th IEEE International Conference on Computing, Communication and Automation, 301-307.

## Project Outcome

****

**Paper Submission Proof**

****

CS-25-35-101

ORIGINALITY REPORT

8

%

SIMILARITY INDEX

6%

INTERNET SOURCES

2%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

2

3

4

5

6

7

8

9

10

11

12

### [www.coursehero.com](http://www.coursehero.com/)

Internet Source

### Submitted to KIET Group of Institutions, Ghaziabad

Student Paper

### dspace.dtu.ac.in:8080

Internet Source

### medium.com

Internet Source

### ndltd.ncl.edu.tw

Internet Source

### tudr.thapar.edu:8080

Internet Source

### Submitted to University of Wales Institute, Cardiff

Student Paper

### technodocbox.com

Internet Source

### dspace.daffodilvarsity.edu.bd:8080

Internet Source

### files.eric.ed.gov

Internet Source

### symorg.fon.bg.ac.rs

Internet Source

### [www.frontiersin.org](http://www.frontiersin.org/)

Internet Source

2%

2%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

13

14

15

16

17

18

19

20

21

22

23

24

25

### [www.researchgate.net](http://www.researchgate.net/)

Internet Source

### Submitted to HTM (Haridus- ja Teadusministeerium)

Student Paper

### Submitted to University of Bedfordshire

Student Paper

### Rajan Gupta, Sunil K. Muttoo, Saibal K. Pal. "A comprehensive review of approaches used for solving tele-centre location allocation problem in geographical plane", International Journal of Operational Research, 2021

Publication

### Submitted to CTI Education Group

Student Paper

### Submitted to University of Glamorgan

Student Paper

### Submitted to University of Ulster

Student Paper

### Submitted to Leeds Beckett University

Student Paper

### Submitted to NCC Education

Student Paper

### ir.juit.ac.in:8080

Internet Source

### studentsrepo.um.edu.my

Internet Source

### [www.tandfonline.com](http://www.tandfonline.com/)

Internet Source

### Submitted to University of Greenwich

Student Paper

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

26

27

28

29

30

31

32

33

34

35

36

37

38

### idoc.tips

Internet Source

### myfik.unisza.edu.my

Internet Source

### docslib.org

Internet Source

### Submitted to ABES Engineering College

Student Paper

### hiis.isti.cnr.it

Internet Source

### [www.mdpi.com](http://www.mdpi.com/)

Internet Source

### Seema Shokeen, Pooja Singh, Vinita Tomar, Akshat Jain, Raghav Sharma. "OPTIMIZING EVENT MANAGEMENT IN HIGHER EDUCATION INSTITUTIONS: THE UNFILTERED CONNECT

APPROACH", ShodhKosh: Journal of Visual and Performing Arts, 2024

Publication

### clok.uclan.ac.uk

Internet Source

### eprints.usm.my

Internet Source

### fr.scribd.com

Internet Source

### mail.ijaems.com

Internet Source

### [www.designveloper.com](http://www.designveloper.com/)

Internet Source

### [www.gizmochina.com](http://www.gizmochina.com/)

Internet Source

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

39

### hdl.handle.net

Internet Source

<1%

Exclude quotes Off Exclude bibliography Off

Exclude matches Off