TECHCONNECT: A PEER TUTORING PLATFORM FOR THE COLLEGE OF INDUSTRIAL TECHNOLOGY

Hannah Jade E. Gariguez

Renier Bong F. Remorin

Southern Luzon State University

College of Industrial Technology

August 2025

Chapter I

INTRODUCTION

For many students in technical fields such as industrial technology, academic challenges extend beyond understanding complex concepts, they also involve adapting to intensive workloads, varying course structures, and a lack of personalized support. While traditional classroom instruction provides the core foundation of learning, it often falls short in meeting the individual needs of students who may be struggling with specific subjects or balancing academic responsibilities with hands-on technical requirements. In a highly practical and skill-driven environment like the College of Industrial Technology (CIT) at Southern Luzon State University (SLSU), the need for flexible and targeted academic assistance is more important than ever.

Peer tutoring has emerged as a powerful educational strategy that offers a solution to this gap. It enables students to support one another through structured academic assistance, building a culture of collaboration, shared responsibility, and mutual growth. Research has shown that peer tutoring helps reinforce subject mastery, boosts confidence, and fosters meaningful academic relationships among students (Paolillo, 2024). However, while the concept is widely recognized, the way it is implemented, particularly in technical programs, often lacks the structure and specialization needed to truly make an impact.

At SLSU, the Peer Assisted Learning (PAL) program is the existing framework for tutoring. However, it operates on a generalized level, which limits its effectiveness for

CIT students who face specialized challenges across fields. These programs demand more than general academic support, they require a tutoring environment that is tailored, adaptive, and built to match the rigor and diversity of technical coursework.

In response to these challenges, the proponents propose TechConnect, a web-based peer tutoring platform designed specifically for the College of Industrial Technology. The system aims to centralize and simplify the peer tutoring process by offering intelligent features such as tutor–tutee matching based on subject compatibility, flexible scheduling, and differentiated roles like “Study Buddy” for casual assistance and “Learning Buddy” for more formal support. In addition, the platform includes a dashboard for activity tracking, a resource hub for study materials, and tools that promote tutor participation through incentives like certificates and resume-building features.

The development of TechConnect is also guided by established quality standards such as ISO 25010, ensuring that the platform meets essential criteria for functionality, usability, performance efficiency, reliability, security, and maintainability. By integrating these variables into the design and evaluation of the system, the project not only addresses current limitations in peer tutoring within CIT but also lays the groundwork for future scalability and institutional integration.

More than just a technical solution, TechConnect represents a step toward reimagining academic support at CIT, making it more student-centered, efficient, and aligned with the college’s unique learning environment.

Background of the Study

Students in technical programs such as those offered by the College of Industrial Technology (CIT) at Southern Luzon State University (SLSU) face unique academic challenges that go beyond traditional classroom instruction. With hands-on subjects, evolving curricula, and rigorous skill-based requirements, many CIT students struggle to find the personalized academic support they need to succeed. Although peer tutoring is widely acknowledged as an effective way to reinforce learning, boost academic confidence, and promote collaboration among students (Paolillo, 2024), the current support structures within the university fall short of meeting these specialized needs.

The university's Peer Assisted Learning (PAL) program, facilitated by the Learning Development Center, aims to provide general academic support to all students. However, its broad and non-specialized scope often results in low visibility and limited relevance for CIT students. As a result, students in technical disciplines are left without structured, subject-specific assistance tailored to their coursework.

Several key problems highlight the need for a more focused and responsive academic support system within the College of Industrial Technology. One major issue is the lack of tailored academic support. The existing Peer Assisted Learning (PAL) program does not adequately reflect the technical complexity of CIT's diverse programs. Students enrolled in both legacy and newly implemented curricula often have distinct academic needs that are not sufficiently addressed by the current generalized support system.

Another pressing concern involves tutor recruitment and engagement. Encouraging qualified students to participate as peer tutors remains a challenge, largely due to the absence of meaningful incentives. Participation tends to be inconsistent and unsustainable without formal recognition. Feedback from the CIT Student Council suggests that incentives such as certificates, resume-building opportunities, and official recognition are crucial to motivating and retaining student tutors.

In addition, there is a clear problem with unstructured resource and session management. The lack of a centralized platform to organize tutoring sessions, share learning materials, or document academic progress creates inefficiencies. Tutors have limited tools for uploading content or tracking tutee development, while students struggle to find accessible review resources.

Peer tutoring within CIT is also fragmented and informal, often relying on personal arrangements rather than a coordinated system. This disorganization diminishes the potential impact of tutoring and weakens accountability among both tutors and tutees. Finally, the absence of a digital platform results in limited administrative oversight and analytics. Faculty and department heads currently lack effective tools to monitor session activities, evaluate tutor performance, or analyze academic trends, all of which are essential for informed academic planning and support.

These gaps hinder the academic success and engagement of CIT students. Studies have shown that unstructured peer mentoring programs are less effective in improving student productivity and well-being (Murrell & Blake-Beard, 2021; Nwaesei & Liao, 2023). A structured, technology-driven system is therefore critical to addressing these inefficiencies.

In response to these challenges, this capstone project proposes the development of TechConnect, a web-based peer tutoring platform exclusively for the College of Industrial Technology. The system is designed to streamline the peer tutoring process by offering dynamic tutor–tutee matching, subject tagging, flexible scheduling, progress tracking, and centralized access to academic resources. It also incorporates tutor incentives and role differentiation to support engagement and structure.

By aligning with the academic environment of CIT and using modern web technologies, TechConnect offers a tailored solution that empowers students, supports tutors, and enhances institutional coordination of peer tutoring activities.

Objectives of the Study

The main objective of the project is to provide an academic support solution tailored to the needs of the College of Industrial Technology by creating a structured, accessible, and efficient peer tutoring environment through a web-based system. Specifically, it aims to:

1. Design and develop a web-based platform called TechConnect: A Peer Tutoring Platform for the College of Industrial Technology, which will be capable of:
    a. Providing an efficient peer tutoring platform that allows students to securely create profiles, indicate academic needs or tutoring expertise,

browse available tutors or tutees using fuzzy search and rule-based filtering (enabling approximate matches for easier tutor discovery), and schedule tutoring sessions through integration with Google Calendar for scheduling and Google Meet for video conferencing.

b. Implementing a filtering system that finds tutors based on subject specialization, availability, learning preferences, and minimum rating thresholds.

c. Allowing tutors to upload review materials and session notes, and enabling tutees to download resources.

d. Providing a dashboard that visualizes tutoring activity, session history, and feedback summaries to support self-improvement and data-driven decisions.

e. Enabling administrators to manage user accounts, monitor tutoring sessions, and handle tutor recognition or reporting concerns.

f. Allowing the department to post announcements, upcoming workshops, or academic support events relevant to the College of Industrial Technology.

g. Addressing tutor recruitment challenges by incorporating incentives such as certificates, resume-building opportunities, and formal recognition to encourage participation and sustained engagement.

2. Evaluate the TechConnect system if it complies with the ISO 25010 standards in terms of:

- Functional Suitability

- Performance Efficiency
- Compatibility
- Interaction Capability
- Reliability
- Security
- Maintainability
- Flexibility
- Safety

3. Prepare an implementation plan for the deployment of TechConnect.

Significance of the Study

The capstone project titled TechConnect: A Peer Tutoring Platform for the College of Industrial Technology is designed to improve access to academic support by offering a structured and technology-enabled environment for peer-assisted learning. By addressing existing limitations in tutoring coordination and student engagement, the system aims to enhance educational outcomes and foster a collaborative learning culture within CIT. The project is expected to benefit the following:

University Leaders will benefit from the project as it aligns with Southern Luzon State University's broader mission of academic innovation and student success. By introducing a scalable, data-informed tutoring framework, the platform provides a model for academic support that can extend beyond CIT to other departments.

CIT Administrators are equipped with a centralized system to oversee tutoring operations, monitor performance metrics, and ensure the effective delivery of peer-led academic services. This strengthens the college's institutional support mechanisms and reinforces its commitment to high-quality technical education.

Faculty Members gain a complementary tool that helps address basic academic concerns, allowing them to focus on advanced instruction. With access to session feedback and analytics, faculty can identify learning trends and common student challenges, improving overall teaching strategies.

CIT Students, as the primary users, benefit from accessible and personalized academic assistance tailored to their subject needs. Tutees receive structured support to help them succeed, while tutors reinforce their understanding of course material, sharpen leadership abilities, and engage in meaningful academic service.

CIT Tutors are empowered through formal recognition, skill-building opportunities, and resume-enhancing incentives such as certificates and recommendation letters. The platform also provides them with professional tools for session management and performance tracking.

Researchers gain valuable experience in project development, system analysis, documentation, and academic writing, while deepening their understanding of how peer tutoring can positively influence student outcomes.

Future Researchers may use this study as a reference for further exploration of web-based academic platforms, peer-assisted learning, or educational technology. The structure and findings of this project may guide similar initiatives in other colleges and institutions.

Scope and Limitation

This project is bounded by several practical constraints that shaped both design and implementation. A key challenge lies in sustaining consistent tutor availability, since participation is voluntary despite incentive mechanisms such as certificates, recognition badges, and system acknowledgments. During peak academic periods, tutor activity is expected to decline as students prioritize their own coursework, which may lead to temporary service gaps. The platform must also address the complexities of CIT's dual curriculum system, where legacy and updated program structures coexist, requiring ongoing adjustments to subject tags and tutor expertise mappings.

Technical limitations arise from the system's reliance on external services, particularly Supabase for authentication, database management, storage, and real-time notifications, as well as Google Meet and Google Calendar APIs for session scheduling and video conferencing. These dependencies impose restrictions such as API quotas, service availability, and potential scalability challenges if platform usage increases significantly. Hosting on Hostinger, while affordable and practical for a student-led initiative, may result in slower response times under heavy traffic compared to cloud-native or dedicated hosting solutions.

Certain design trade-offs were also made to ensure timely project delivery. TechConnect was developed as a standalone system rather than integrated with the university's IT infrastructure, requiring users to register separately instead of leveraging institutional single sign-on. While this decision simplified development, it added an extra onboarding step. Similarly, features such as offline access, native mobile applications, and fully automated content moderation were excluded due to resource and time constraints. Notifications are currently limited to in-app delivery through Supabase Realtime, without push alerts outside the web platform. The donation module, though functional via QR code payments, remains basic and does not yet include automated tracking or third-party gateway integration.

Administrative functions such as tutor verification, event management, and learning material approval continue to rely on manual oversight, as full automation was beyond project scope. Despite these limitations, TechConnect successfully delivers a foundational peer tutoring ecosystem for CIT, with a modular architecture that supports future performance improvements, feature enhancements, and potential integration with institutional systems as resources permit.

Definition of Terms

The following terms are defined conceptually and operationally to support clear understanding of the variables and technical elements used in this study:

Administrator is a user role within TechConnect responsible for managing accounts, overseeing sessions, posting announcements, and monitoring platform activity.

Deployment is the process of launching the finalized platform online, making it accessible to users through Hostinger's web hosting service.

Fuse.js is a lightweight JavaScript library used in the platform to implement fuzzy search functionality, allowing users to retrieve relevant results even with typographical errors or partial input matches. It enhances search accuracy and user experience by ranking results based on similarity scores.

Implementation is the execution phase where the system is developed, tested, and prepared for actual use by its intended users.

ISO 25010 is a software quality model used to evaluate the platform based on criteria such as functionality, usability, reliability, performance, and maintainability.

Learning Buddy is a platform-assigned user role that represents a formal tutor responsible for structured academic sessions.

Progress Tracking is a system feature that monitors and displays a student's academic development based on session participation and feedback.

Resource Hub is a centralized section of the platform where tutors upload study materials and tutees can access, bookmark, or download them.

Rule-based Matchmaking is a filtering logic used in the platform that connects tutors and tutees based on predefined conditions such as year level, subject expertise, and availability. It follows strict “if-then” logic to ensure accurate and structured pairing, enhancing reliability and relevance in peer tutoring sessions.

Study Buddy is a casual peer-to-peer support role in the platform meant for informal or less structured tutoring sessions.

Tutor is a student user who provides academic assistance to peers by conducting tutoring sessions and sharing learning resources.

Tutor Tagging is a labeling system that identifies a tutor’s subject expertise to improve the accuracy of tutor–tutee matching.

Tutee is a student user who seeks academic support through the platform and books sessions with available tutors.

User Roles are defined access levels and responsibilities within the platform, such as administrator, tutor, tutee, Study Buddy, and Learning Buddy.

Chapter II

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter presents a review of existing literature and studies relevant to the capstone research project. The researchers gathered information from journals, books, and the internet to provide more information about the study.

Digital Learning Support in the Current Generation

The integration of digital technologies into education has fundamentally transformed the pedagogical landscape, creating new opportunities for enhanced accessibility, engagement, and personalization in learning experiences. Zou et al. (2025) emphasize that digital learning, characterized by the integration of information and communication technologies (ICT) into educational practices, has become a cornerstone of modern education. The COVID-19 pandemic served as a catalyst, accelerating the adoption of digital technologies and highlighting their potential to facilitate continuous learning even during unprecedented global challenges. This transformation extends beyond simply replacing traditional teaching methods to fundamentally enhancing the educational experience through more flexible, accessible, and personalized learning environments.

Digital learning platforms have emerged as essential tools for contemporary education, offering flexible and accessible learning opportunities across various educational levels. The implementation of adaptive learning technologies, such as Smart Sparrow and ALEKS, demonstrates the power of data analytics in creating personalized learning journeys responsive to individual student needs (Zou et al., 2025). These systems monitor student progress in real-time and dynamically adjust content complexity, resulting in significantly higher pass rates and greater overall engagement compared to conventional learning environments. Artificial intelligence has become a significant catalyst in enhancing learning outcomes, with AI-driven tools including virtual teaching assistants and automated grading systems alleviating administrative burdens while enabling educators to focus on personalized instruction.

However, the rapid advancement of digital learning technologies also presents significant challenges that must be addressed to ensure equitable and effective implementation. Technical and infrastructure limitations remain pressing concerns, particularly the digital divide between those with access to technology and those without, especially in underdeveloped regions and rural areas (Zou et al., 2025). Additionally, pedagogical challenges emerge as educators must adapt from traditional face-to-face instruction to digital platforms, requiring fundamental changes in instructional design and delivery. The study emphasizes that addressing these challenges requires a multifaceted approach including inclusive policies, infrastructure investment, and continuous professional development for educators to create truly effective digital learning environments.

Southeast Asia has experienced rapid growth in digital technology applications within education, fundamentally changing educational delivery methods across the region's diverse contexts. According to a comprehensive regional report by UNESCO and the Southeast Asia Ministers of Education Organization (SEAMEO, 2023), the region boasts approximately 400 million internet users, with 40 million people going online for the first time in 2020 alone. This expansion has been identified for its potential to transform education and meet regional development aspirations, particularly in areas such as personalized tutoring and testing, learning management systems, language learning, and skills development. However, the report cautions that technology is not a universal solution for addressing major educational challenges, and its potential for improving teaching and learning requires empirical validation.

The benefits of digital learning developments in Southeast Asia have been uneven, varying significantly by socioeconomic level, educational attainment, and teacher preparedness. Students from the wealthiest households in the region are nearly eight times more likely to have home internet connectivity compared to their poorest counterparts, while only half of rural primary schools are connected to the internet (SEAMEO, 2023). This digital divide has created significant disparities in educational access and outcomes. Furthermore, in countries including Cambodia, Lao PDR, Malaysia, Myanmar, the Philippines, and Vietnam, nearly one-third of primary school teachers reported feeling 'not very' or 'not at all' confident in using ICT in 2019, highlighting the critical need for enhanced teacher training and support.

Three essential conditions have been identified as necessary for realizing the full potential of educational technology in Southeast Asia: equitable access to technology, appropriate governance and regulation, and sufficient teacher capacity. The report emphasizes that regulations are particularly crucial for protecting children's wellbeing, noting that 70% of surveyed adolescents in Cambodia, Indonesia, Malaysia, and Thailand reported upsetting experiences associated with online activities (SEAMEO, 2023). Additionally, protection of children's data is needed, as less than one-third of countries currently guarantee data privacy in education through law or policy. These findings underscore the importance of comprehensive approaches that address not only technological infrastructure but also governance frameworks and human capacity development to ensure effective and ethical digital learning implementation.

The Philippines has experienced significant transformation in its educational landscape through digital learning adoption, marking a critical milestone in the country's educational evolution. This transition involved comprehensive deployment of various online platforms, e-learning tools, and digital resources to facilitate remote education, driven by both technological advancement and educational necessity (MyPrivateTutor.ph, 2024). The Department of Education's initiatives, including the Learning Resources Portal, provide centralized platforms for digital content accessible to teachers and students nationwide. Additionally, mobile learning has gained substantial traction, capitalizing on the Philippines' high mobile phone penetration rate, though challenges persist in ensuring equitable access in remote and underprivileged areas where internet connectivity and digital device availability remain limited.

Educational technology integration in Philippine higher education has shown remarkable progress, with 87% of private and 67% of public higher education institutions incorporating Learning Management Systems (LMS) into their classrooms by 2022 according to World Bank findings. The adoption of cloud-based educational technologies has enabled institutions to provide more cohesive and flexible learning environments, supporting both in-person and hybrid learning models that became essential during the pandemic (BusinessWorld, 2024). Significantly, 50% of Filipino students access their institution's LMS via mobile phones, highlighting the critical importance of mobile-optimized learning experiences. This mobile-first approach reflects the practical reality of educational access in the Philippines and demonstrates the adaptability of educational institutions to local technological contexts.

Despite technological advances, accessibility gaps and digital literacy disparities remain significant challenges in Philippine digital education. The digital divide disproportionately affects students from lower-income families who may lack reliable internet access, personal computers, or appropriate learning environments for online education. An overreliance on free educational materials—utilized by 74% of educators and 64% of students—highlights ongoing inequality in resource availability and distribution (Inquirer Business, 2024). Furthermore, institutions must prioritize comprehensive training and guidance for both students and educators in effectively using digital and physical resources. The expansion of educational technology in the Philippines has opened new possibilities for student empowerment, but realizing these benefits requires continued efforts to bridge accessibility gaps and ensure that technological integration supports all learners regardless of their socioeconomic circumstances.

Peer Tutoring Systems in Modern Education

Recent research has demonstrated the transformative impact of structured pedagogical training on peer tutoring effectiveness, revealing significant improvements in both tutor teaching capabilities and tutee learning outcomes. Lin et al. (2025) conducted a comprehensive mixed-methods study at a Sino-American university, examining how peer tutors trained in the BOPPPS (Bridge-in, Objective, Pre-assessment, Participatory Learning, Post-assessment, Summary) pedagogical framework compared to untrained tutors. Their findings revealed a notable reduction in controlling teaching styles among trained tutors, with pedagogically trained tutors developing clearer teaching personas, enhanced communication skills, and more effective engagement strategies. The study involved 174 valid data points across three time periods, demonstrating that tutors who completed formal pedagogical training showed profound improvements not only in their instructional methods but also in their ability to create supportive learning environments that foster tutee academic confidence and satisfaction.

The application of structured pedagogical frameworks in peer tutoring contexts has proven particularly effective in enhancing the quality of tutor-tutee interactions and learning outcomes. Lin et al. (2025) found that tutees working with pedagogically trained tutors reported significantly higher levels of self-efficacy, intrinsic motivation, and overall satisfaction with their learning experiences. The BOPPPS framework enabled tutors to implement systematic scaffolding strategies that supported tutee engagement and comprehension, resulting in more meaningful academic interactions. Semi-structured interviews revealed that tutees showed a clear preference for sessions utilizing scaffolding strategies, which tutors had acquired through their pedagogical training. This preference was manifested in improved academic confidence and more positive attitudes toward peer tutoring programs, suggesting that formal pedagogical training addresses critical gaps in traditional peer tutoring approaches that rely solely on subject matter expertise.

However, the study also identified important limitations and areas for improvement in pedagogical training implementation. Not all tutees experienced uniform improvements in academic confidence, with some students—particularly those with lower initial motivation or learning style mismatches—showing less pronounced benefits (Lin et al., 2025). Language barriers emerged as a significant factor, especially when tutors over-relied on students' first language for explanations, potentially limiting the development of academic English skills. These findings underscore the importance of comprehensive training programs that address not only pedagogical techniques but also cultural sensitivity, adaptive teaching strategies, and assessment of tutor-tutee compatibility. The research suggests that future peer tutoring programs should incorporate more sophisticated matching processes and ongoing training support to maximize the benefits of pedagogical preparation while addressing individual learner variability and needs.

Recent meta-analytical research has provided compelling evidence for the effectiveness of peer tutoring across diverse educational contexts, with particular strength demonstrated in STEM disciplines. A comprehensive meta-analysis involving 24 studies with 3,311 participants revealed that peer tutoring has a significant positive effect on academic achievement (effect size = 1.23, 95% CI [0.75, 1.70], $p < 0.001$) and other learning outcomes (effect size = 0.40) (ScienceDirect, 2025). The analysis showed substantial variations in effect sizes between different subjects, though no significant differences were found between different types of peer tutoring approaches. These findings suggest that peer tutoring's potential to significantly improve academic performance is particularly pronounced in STEM fields, supporting the expansion of peer tutoring programs in these critical educational areas where student success has broad implications for career preparation and societal development.

The benefits of peer tutoring extend beyond academic achievement to encompass enhanced student confidence, improved critical thinking skills, and increased motivation for both tutors and tutees. Research by Arco-Tirado et al. (2019) and Murtonen et al. (2023) has documented that peer tutoring enhances academic performance while developing critical thinking skills among participants. Furthermore, peer tutoring has been shown to reduce dropout rates and increase student confidence across various educational levels. The effectiveness of peer tutoring appears to be enhanced when tutors receive structured training that equips them to handle diverse learning needs and foster meaningful academic interactions with tutees. Cross-year small-group tutoring, where senior students serve as tutors for junior students, has been particularly effective in creating supportive learning environments that benefit both parties through knowledge reinforcement and skill development.

However, contemporary research also identifies important factors that influence peer tutoring effectiveness and require careful consideration in program design and implementation. Studies have revealed that tutor-tutee mismatches can hinder the benefits of peer tutoring, emphasizing the importance of appropriate pairing processes that consider learning styles, personality compatibility, and communication preferences (Topping, 2005). Additionally, the quality of training provided to peer tutors significantly impacts program outcomes, with structured pedagogical preparation showing superior results compared to programs relying solely on academic qualification. Modern peer tutoring programs increasingly incorporate technology-enhanced elements, online platforms, and digital collaboration tools that expand access and flexibility while maintaining the essential interpersonal connections that drive peer learning effectiveness. These developments suggest that successful contemporary peer tutoring requires thoughtful integration of pedagogical training, careful matching processes, and appropriate technological support to maximize benefits for all participants.

API Integration in Learning Platforms

The integration of Application Programming Interfaces (APIs) in educational platforms has emerged as a critical enabler for seamless learning experiences and automated administrative processes. Hyperlink InfoSystem (2025) demonstrates how educational institutions leverage APIs to standardize processes and automate routine reporting tasks, citing the University of British Columbia's implementation where API integration reduced student graduation list processing from a week to same-day delivery. This transformation highlights the profound impact of API standardization on administrative efficiency and data accuracy in educational environments.

API-driven scheduling automation represents a particularly significant advancement in educational technology, with platforms like Cronofy and Nylas providing comprehensive calendar integration solutions. Cronofy (2024) reports that their scheduling automation platform reduced interview scheduling time from 6 days to a median of 90 minutes for recruitment teams, demonstrating the transformative potential of automated scheduling in educational contexts. The Nylas Calendar API offers end-to-end scheduling solutions that eliminate double bookings and enable customizable workflows, providing educational institutions with the flexibility to configure scheduling processes according to their specific operational requirements.

The technical implementation of educational APIs extends beyond basic calendar integration to encompass comprehensive learning management system interoperability. Binariks (2024) emphasizes that educational data interoperability through API management platforms eliminates the need for expensive custom development while enabling seamless data sharing between multiple e-learning solutions. Their analysis reveals that 71% of K-12 edtech leaders identify insufficient expertise as a primary challenge in implementing educational data interoperability, highlighting the need for robust API management platforms that can bridge the technical knowledge gap while maintaining system security and performance standards.

The Southeast Asian educational technology landscape demonstrates unique approaches to API integration, particularly in response to regional connectivity challenges and diverse linguistic requirements. Indonesian Cloud (2024) highlights how Learning Management Systems in the region prioritize mobile-first API designs that accommodate varying internet connectivity speeds and device capabilities. Their platform architecture emphasizes real-time synchronization capabilities that enable seamless access across mobile applications and desktop platforms, reflecting the region's mobile-centric digital infrastructure.

Philippine educational institutions have adopted sophisticated API integration strategies to address scalability challenges in higher education enrollment management. The University of Computer Indonesia (UNIKOM) LMS (2024) exemplifies this approach through automated synchronization systems that perform data integration every 4 hours during business hours, ensuring real-time availability of course materials and student progress tracking. Their implementation demonstrates how educational APIs can be architected to handle high-volume student data while maintaining system performance and data integrity across multiple academic systems.

Regional Learning Management System providers like Paradiso LMS and Ruangkerja (2024) have developed comprehensive API ecosystems that support multi-tenancy, AI-powered personalization, and seamless third-party integrations tailored to Southeast Asian educational contexts. These platforms demonstrate advanced API integration capabilities including Salesforce and Microsoft Dynamics connections, gamification engines, and multi-language support systems. The success of Ruangkerja's implementation across major Indonesian corporations like Pertamina and Tokopedia, achieving 93-98% completion rates, illustrates the effectiveness of well-designed API integration in supporting large-scale corporate learning initiatives.

Rule-Based Filtering and Matchmaking in Peer Platforms

Rule-based expert systems have emerged as fundamental components in intelligent tutoring systems, providing transparent and maintainable frameworks for tutor-tutee matching in educational platforms. Lin et al. (2023) conducted a comprehensive systematic review of AI implementation in intelligent tutoring systems, identifying rule-based approaches as critical for creating transparent matching algorithms that educators can understand and modify. Their analysis of 29 research papers reveals that rule-based systems enable educational platforms to implement adaptive learning frameworks while maintaining explainability, addressing the "black box" concern often associated with machine learning approaches in education.

The technical architecture of rule-based matching systems in educational contexts requires sophisticated weighting mechanisms that can evaluate multiple criteria simultaneously. US Patent 20020013836A1 describes a comprehensive tutor matching system that utilizes predetermined criteria rankings combined with proficiency measurements to compute qualifiers for each potential tutor. The system incorporates dynamic weight adjustment capabilities and multi-tiered rules that can compensate for subjective student descriptions through objective weights developed by education experts. This approach demonstrates how rule-based systems can achieve both flexibility and objectivity in educational matching scenarios.

Contemporary implementations of rule-based tutoring systems show particular effectiveness in mathematical problem-solving contexts, where step-by-step guidance can be systematically encoded into rule structures. Shih et al. (2023) developed a mathematics intelligent tutoring system for fraction operations that employs block-based matching methods with decision trees built from collected expectation and misconception response patterns. Their system achieved significant learning improvements through rule-based error identification and personalized feedback delivery, demonstrating the practical effectiveness of transparent rule-based approaches in educational applications where explainability is paramount for both students and educators.

Educational recommender systems increasingly implement rule-based filtering mechanisms to ensure transparency and maintainability in matching algorithms, addressing growing concerns about algorithmic accountability in learning platforms. Ventionai (2024) emphasizes how rule-based cognitive models serve as the intelligence behind AI tutoring systems, enabling platforms to construct transparent student knowledge models essential for flexible learning environments. These systems predict student actions and anticipate potential errors through explicit rule structures that educators can examine, modify, and validate, ensuring that matching decisions remain interpretable and pedagogically sound.

The implementation of rule-based filtering in peer tutoring platforms requires sophisticated handling of multi-dimensional matching criteria while maintaining system performance and scalability. TutorCruncher (2024) demonstrates this through their tutor filtering system that enables administrators to filter tutor profiles by teaching skills, location, and niche subject proficiencies in real-time. Their rule-based approach supports both automated matching notifications and manual override capabilities, providing flexibility for different institutional preferences while maintaining transparent decision-making processes that administrators can understand and adjust based on educational outcomes.

Research in educational data mining reveals that rule-based systems offer superior maintainability compared to purely machine learning approaches, particularly in dynamic educational environments where matching criteria frequently evolve. The systematic review by Lin et al. (2023) identifies that educational institutions prefer rule-based approaches for their ability to incorporate pedagogical theories and expert knowledge directly into matching algorithms. This preference stems from the educational sector's emphasis on evidence-based practices and the need for matching systems that can be validated against established educational principles and adapted to changing institutional requirements.

Fuzzy Logic and Approximate Matching in Educational Tools

Fuzzy string matching has emerged as a critical technology for handling unstructured course inputs and approximate matching in educational platforms, addressing the inherent ambiguity in how students and educators describe learning objectives and course content. Nath (2022) provides a comprehensive analysis of fuzzy matching algorithms, demonstrating how Levenshtein distance, Damerau-Levenshtein, and n-gram approaches can effectively handle course description variations, misspellings, and abbreviated formats commonly encountered in educational databases. These algorithms enable educational platforms to achieve high matching accuracy even when dealing with inconsistent user inputs, making them essential for robust course recommendation and enrollment systems.

The technical implementation of fuzzy string matching in educational contexts requires careful consideration of algorithm selection based on the specific characteristics of educational data. The DataCamp tutorial (2024) illustrates how different fuzzy matching approaches excel in various educational scenarios: Levenshtein distance for handling simple typographical errors in course codes, Jaro-Winkler distance for matching course names with different word orders, and phonetic algorithms like Soundex for handling pronunciation-based variations in course titles. The tutorial emphasizes that educational platforms benefit from hybrid approaches that combine multiple algorithms to achieve optimal matching performance across diverse content types.

Performance optimization remains a critical consideration in educational fuzzy matching systems, particularly when dealing with large course catalogs and real-time matching requirements. The Towards Data Science analysis (2024) reveals that traditional FuzzyWuzzy implementations can require up to 337 hours for processing large educational datasets, while optimized approaches like FuzzyCouple using TF-IDF vectorization and cosine similarity can complete the same tasks in 21 minutes. This dramatic performance improvement demonstrates the importance of algorithm selection and implementation strategy in creating scalable educational matching systems that can handle institutional-scale course catalogs efficiently.

Advanced educational platforms increasingly incorporate fuzzy logic mechanisms to enhance the flexibility and contextual appropriateness of automated question-answering systems. Chen et al. (2024) developed the Chaotic LLM-based Educational Q&A System (CHAQS) that integrates fuzzy logic with large language models to dynamically adjust response parameters based on question characteristics. Their system implements fuzzy membership functions that categorize question lengths into short, medium, and long categories, with corresponding parameter adjustments for top_p and temperature settings that optimize response diversity and coherence based on the information content of student queries.

The technical architecture of fuzzy logic educational systems requires sophisticated rule definition and parameter tuning to achieve optimal performance across diverse educational contexts. The CHAQS implementation demonstrates how fuzzy rules can be designed to handle varying levels of question complexity: short questions receive high top_p values with low temperature settings to introduce diversity while maintaining coherence, while longer questions receive low top_p values with high temperature settings to focus vocabulary selection while introducing creative elements. This adaptive parameter adjustment achieved 5.12% improvement in precision, 11% increase in recall, and 8% improvement in F1 scores compared to baseline models.

Empirical evaluation of fuzzy logic educational systems reveals significant advantages in handling the inherent ambiguity and context-sensitivity of educational interactions. The comprehensive evaluation by Chen et al. (2024) using BERT F1, precision, and recall scores, alongside human evaluation across coherence, consistency, fluency, relevance, and accuracy dimensions, demonstrates that fuzzy logic augmentation enables educational systems to provide more contextually appropriate responses. Their comparison with state-of-the-art models including Qwen-7B-Chat, Baichuan2-7B-Chat, and Llama-2-7B-Chat confirms the effectiveness of fuzzy logic integration in educational applications where nuanced understanding of student intent is crucial for providing effective learning support.

Modern Web Frameworks in E-Learning

Modern educational platforms increasingly rely on dynamic, responsive user interfaces to deliver engaging learning experiences. React.js has emerged as a leading framework for building educational applications due to its component-based architecture and virtual DOM optimization (Redvike, 2023). Educational technology companies recognize React.js's ability to create immersive learning experiences through reusable components that can track student performance, provide real-time feedback, and adapt to individual learning paths. The framework's declarative nature allows developers to build complex educational interfaces while maintaining code readability and performance optimization.

The adoption of React.js in prominent educational platforms demonstrates its practical effectiveness in real-world scenarios. Khan Academy underwent a major migration to React.js, which provided smooth element transitions, eliminated unnecessary re-renders, and enabled efficient development cycles (Redvike, 2023). Similarly, platforms like Codecademy, Skillshare, and Duolingo have leveraged React.js to create highly responsive user interfaces that support interactive learning features such as live coding environments, collaborative tools, and real-time progress tracking. The framework's compatibility with legacy systems and search engine optimization capabilities make it particularly valuable for educational institutions transitioning from traditional learning management systems.

React.js's strength in educational applications lies in its ability to handle concurrent user interactions and complex state management, which are critical requirements for modern e-learning platforms. The framework's extensive ecosystem of libraries and community support enables rapid development of educational features such as gamification elements, multimedia content delivery, and social learning tools (Redvike, 2023). As educational technology continues to evolve toward more personalized and interactive learning experiences, React.js provides the technical foundation necessary to support these innovations while ensuring scalable, maintainable code architecture that can adapt to changing pedagogical requirements.

Next.js has revolutionized e-learning platform development by addressing the unique performance challenges faced by educational applications that serve diverse global audiences. The framework's hybrid rendering model, combining Server-Side Rendering (SSR) and Static Site Generation (SSG), enables educational platforms to deliver both static content like course catalogs and dynamic user-specific data such as progress tracking with optimal performance (PagePro, 2024). This architecture is particularly valuable for educational platforms that must handle varying content types, from high-quality videos and interactive exercises to personalized learning paths and real-time collaboration features.

The implementation of Next.js in educational platforms demonstrates significant measurable improvements in user experience and business outcomes. Learn Squared, an e-learning platform serving over 20,000 students worldwide, experienced a 24.35% increase in revenue within 26 days of migrating from Drupal to Next.js (PagePro, 2024). The transition resolved critical performance bottlenecks, eliminated server-side rendering delays during high-traffic periods, and enabled seamless integration with third-party educational tools including JWPlayer for video content, Shopify for e-commerce functionality, and Discord for community interactions. The platform's improved load times and reduced bounce rates directly contributed to enhanced user engagement and retention.

Next.js addresses several critical requirements specific to educational platforms, including multilingual content delivery, real-time collaborative features, and secure handling of sensitive educational data. The framework's built-in internationalization capabilities enable global educational platforms to serve localized content efficiently, while its support for WebSockets and integration with real-time services facilitates live discussions, collaborative exercises, and instant feedback mechanisms (PagePro, 2024). Additionally, Next.js's security features and modular architecture support GDPR compliance and enterprise-grade data protection, making it suitable for educational institutions that handle sensitive student information and must meet stringent regulatory requirements.

Database Technologies in Educational Platforms

Supabase has emerged as a compelling database solution for educational platforms by combining the reliability of PostgreSQL with real-time capabilities essential for modern learning environments. The platform's architecture leverages Postgres's three-decade development history while adding real-time subscriptions, instant APIs, and built-in authentication systems that are particularly valuable for educational applications requiring live collaboration features (Supabase, 2024). Educational developers can implement features such as real-time discussion forums, collaborative document editing, and instant progress updates without managing complex infrastructure, making Supabase an attractive option for institutions seeking to modernize their learning management systems while maintaining data reliability and security.

The real-time capabilities of Supabase address critical requirements in contemporary educational technology, including live chat functionality, collaborative learning sessions, and instant feedback mechanisms. The platform's Realtime Server utilizes Postgres's built-in replication functionality to convert database changes to JSON and broadcast updates via WebSockets to authorized clients, enabling seamless real-time experiences across educational applications (Supabase, 2024). This architecture supports educational use cases such as live peer tutoring sessions, real-time quiz participation, group project collaboration, and instructor-student communication without requiring educational institutions to develop and maintain complex real-time infrastructure from scratch.

Supabase's comprehensive feature set, including Row Level Security, Edge Functions, and vector embeddings support, provides educational platforms with enterprise-grade capabilities while maintaining development simplicity. The platform's integration with modern web frameworks and its extensive API coverage enable educational developers to build sophisticated learning management systems that can scale from small classroom applications to institution-wide deployments serving thousands of students (Supabase, 2024). The combination of PostgreSQL's ACID compliance, Supabase's real-time features, and built-in authentication makes it particularly suitable for educational applications that must handle sensitive student data while providing engaging, interactive learning experiences that meet contemporary educational technology expectations.

PostgreSQL has established itself as a preferred database solution for educational institutions due to its robust feature set, proven reliability, and comprehensive support for complex educational data relationships. The database's advanced capabilities including JSON support, full-text search, and geospatial data handling make it particularly suitable for modern educational applications that must manage diverse content types ranging from traditional student records to multimedia learning resources, location-based services for campus management, and complex assessment data structures (Supabase, 2024). Educational institutions benefit from PostgreSQL's ACID compliance and strong consistency guarantees when handling sensitive student information, financial records, and academic transcripts that require absolute data integrity and regulatory compliance.

The extensibility of PostgreSQL through its comprehensive ecosystem of extensions provides educational developers with powerful tools for implementing specialized educational features without requiring additional database systems. Extensions such as pg_vector enable sophisticated educational analytics including semantic search capabilities for course content discovery, student performance clustering analysis, and personalized learning recommendation systems (Supabase, 2024). The database's support for stored procedures, triggers, and custom functions allows educational institutions to implement complex business logic directly at the database level, including automated grading calculations, prerequisite validation, and audit trail maintenance that are essential for academic record keeping and institutional compliance requirements.

PostgreSQL's performance characteristics and scaling capabilities make it well-suited for the variable workload patterns typical in educational environments, where usage may spike dramatically during enrollment periods, examination times, and semester transitions. The database's support for read replicas, connection pooling, and partitioning strategies enables educational institutions to maintain consistent performance across different usage scenarios while managing costs effectively (Supabase, 2024). Additionally, PostgreSQL's strong community support and extensive documentation provide educational IT teams with reliable resources for ongoing database administration, troubleshooting, and optimization, making it a sustainable long-term choice for institutions that must balance technical capabilities with administrative overhead and resource constraints.

UI/UX Design and Responsive Frameworks

Tailwind CSS has gained recognition for its utility-first approach that enables developers to create accessible educational interfaces while maintaining design flexibility and development efficiency. The framework's comprehensive accessibility utilities, particularly the sr-only and not-sr-only classes, provide essential tools for creating inclusive educational platforms that serve students with diverse accessibility needs (Accreditly, 2024). Educational developers can implement screen reader-friendly interfaces using Tailwind's semantic HTML support, ensuring that educational content remains accessible to students using assistive technologies while maintaining visually appealing designs that enhance learning engagement for all users.

The implementation of WCAG compliance in educational platforms using Tailwind CSS requires careful attention to color contrast ratios, keyboard navigation support, and semantic markup structures that preserve the inherent accessibility features of HTML elements. Tailwind's extensive color palette includes utilities specifically designed to meet accessibility standards, while its focus utilities ensure that keyboard navigation remains clear and consistent throughout educational interfaces (Accreditly, 2024). Educational platforms benefit from Tailwind's approach of maintaining semantic HTML integrity while providing styling capabilities, enabling developers to create complex educational interfaces including interactive course navigation, accessible form controls, and responsive multimedia content presentation without compromising accessibility requirements.

The responsive design capabilities of Tailwind CSS align particularly well with the diverse device usage patterns common in educational environments, where students access learning materials across smartphones, tablets, and desktop computers. The framework's mobile-first design philosophy ensures that educational content remains accessible and usable across all device types, while its responsive utilities enable developers to create adaptive learning interfaces that optimize content presentation based on screen size and device capabilities (Accreditly, 2024). This responsive accessibility approach is crucial for educational equity, ensuring that students with different technological resources can access educational content effectively while maintaining consistent user experience quality across various access methods and assistive technologies.

Research conducted during the COVID-19 pandemic revealed significant correlations between responsive web design implementation and educational platform usability, with studies showing a 91.5% explanatory relationship ($R^2 = 0.915$) between responsive design features and overall usability measures in academic websites (Parlakkiliç, 2022). Educational institutions that implemented responsive design principles experienced improved user engagement, reduced bounce rates, and enhanced accessibility during the critical transition to remote learning environments. The study's findings demonstrated that university students showed strong preference for educational platforms designed with responsive principles, with 99.2% of participants owning smartphones and 91.3% using mobile devices for internet access, highlighting the critical importance of mobile-optimized educational interfaces.

The technical implementation of responsive design in educational contexts requires sophisticated approaches to content prioritization, navigation optimization, and interactive element adaptation across varying screen sizes and device capabilities. Educational platforms must balance information density requirements with mobile usability constraints, ensuring that complex features such as course catalogs, assignment submission systems, and collaborative learning tools remain functional and accessible across all device types (Parlakkiliç, 2022). The research identified ease of use as the most preferred responsive design feature among educational users, with 73.4% of participants rating this aspect as critically important, indicating that educational responsive design success depends more on functional usability than aesthetic considerations.

The implications of responsive design implementation extend beyond technical considerations to directly impact educational equity and institutional effectiveness, particularly in regions with diverse technological access patterns. Students accessing educational platforms through mobile devices demonstrated equivalent learning engagement and academic performance compared to desktop users when platforms implemented comprehensive responsive design principles (Parlakkiliç, 2022). Educational institutions that prioritized responsive design experienced improved student retention rates, enhanced accessibility compliance, and reduced technical support requirements, demonstrating that investment in responsive educational platform design yields measurable returns in both educational outcomes and operational efficiency while supporting institutional goals of inclusive and accessible education delivery.

Acceptability Testing in Educational Technologies

Lin and Yu (2023) conducted a comprehensive study extending the Technology Acceptance Model (TAM) to investigate higher-education students' acceptance of digital academic reading tools on computers. Their research utilized structural equation modeling with data from 884 participants across Chinese universities, achieving substantial explanatory power with R^2 values ranging from 64.70% to 84.20%. The study introduced six external factors to the traditional TAM model, including lecturer's positive response, academic experience, expectation of academic achievement, ease of access to digital resources, perceived ease for collaborative learning, and self-efficacy. The findings revealed that traditional TAM constructs (perceived usefulness, perceived ease of use, and attitude) remained significant predictors of behavioral intention, with attitude showing the strongest effect on intention to use.

The study's methodology employed rigorous validity and reliability testing, including exploratory and confirmatory factor analysis, with all variables demonstrating excellent internal consistency (Cronbach's $\alpha \geq 0.900$). Particularly noteworthy was the finding that self-efficacy emerged as the strongest predictor of perceived ease of use, while academic experience negatively predicted attitudes toward digital tools. This suggests that students with traditional academic experience may resist transitioning to digital workflows, highlighting the importance of gradual technology integration and comprehensive training programs. The research also revealed that lecturer support significantly influenced both perceived usefulness and student attitudes, emphasizing the critical role of instructor endorsement in technology adoption.

The implications of this research extend to educational technology platform development, particularly for peer tutoring systems where user acceptance is crucial for success. The study's identification of specific factors that enhance technology acceptance provides valuable insights for designing user-centered educational platforms. The substantial explanatory power of the extended model (R^2 = 84.20% for intention to use) demonstrates the importance of considering academic-specific factors rather than relying solely on traditional TAM variables. For peer tutoring platforms like TechConnect, these findings suggest that incorporating features that enhance self-efficacy, providing clear instructor guidance, and addressing the specific academic expectations of users are essential for achieving high acceptance rates and sustained usage.

Kemp (2024) investigated technology acceptance of virtual classrooms through a novel extended educational technology acceptance model, focusing on post-COVID-19 educational contexts. The study examined student attitudes toward virtual learning platforms by integrating cognitive engagement, comfort and well-being factors, and social interaction components into the traditional TAM framework. Using data from university students who experienced the transition to online learning during the pandemic, the research identified that user satisfaction and system attributes were critical determinants of acceptance. The study revealed significant gender differences in self-learning adoption, with female students showing more positive attitudes toward online self-learning compared to their male counterparts.

The research methodology incorporated both quantitative and qualitative approaches, utilizing survey data and user feedback to develop a comprehensive understanding of virtual classroom acceptance. The study found that cognitive engagement, defined as users being absorbed, focused, and entertained during learning activities, strongly influenced perceived usefulness of educational technology. Additionally, the research highlighted the importance of comfort and well-being factors, which emerged as significant predictors of technology acceptance in educational contexts. These findings are particularly relevant given the sustained changes in educational delivery methods following the pandemic, where hybrid and online learning have become permanent fixtures in higher education.

For peer tutoring platforms operating in contemporary educational environments, these findings emphasize the need to address both functional and emotional aspects of user experience. The identification of gender differences in technology acceptance suggests that platform design should consider diverse user needs and preferences. The emphasis on cognitive engagement indicates that peer tutoring systems should incorporate interactive elements, gamification, and engaging content to maintain user interest and participation. Furthermore, the study's focus on comfort and well-being factors suggests that successful educational technology platforms must create psychologically safe environments where users feel confident and supported in their learning interactions.

Agile-Kanban in Software Development for Education

Dong et al. (2024) conducted a comprehensive systematic literature review to develop a new definition of Agile Project Management, analyzing its expansion beyond software development into various sectors including education. Their research revealed that agile practices have been increasingly adopted in educational institutions for managing complex projects, developing educational technologies, and improving institutional processes. The study identified that while agile methodologies originated in software development, their application in education has evolved to address the unique challenges of academic environments, including uncertain requirements, diverse stakeholder needs, and the necessity for continuous adaptation to changing educational demands. The research highlighted that educational institutions face particular challenges in implementing agile practices due to traditional organizational structures and resistance to change among faculty and administrative staff.

The systematic review methodology employed by Dong et al. encompassed analysis of empirical studies, case studies, and theoretical frameworks across multiple databases, ensuring comprehensive coverage of agile implementation in educational contexts. The findings revealed that successful agile adoption in education requires careful consideration of institutional culture, stakeholder buy-in, and gradual implementation strategies. The study identified key success factors for agile implementation in educational settings, including leadership support, cross-functional team formation, and iterative approach to project delivery. Additionally, the research emphasized the importance of adapting agile principles to accommodate the unique constraints of educational environments, such as academic calendars, regulatory requirements, and diverse user populations.

For educational software development projects, particularly peer tutoring platforms, these findings provide crucial insights into effective project management approaches. The research suggests that educational technology development can benefit significantly from agile methodologies, particularly in managing the iterative nature of educational software design and the need for continuous stakeholder feedback. The study's emphasis on adaptation and flexibility aligns well with the requirements of developing platforms that must serve diverse educational contexts and user needs. Implementation of agile practices in educational software development should consider the specific challenges identified in the research, including the need for extensive stakeholder engagement, accommodation of academic schedules, and integration with existing educational systems and processes.

Research on Kanban methodology implementation in educational and academic contexts has demonstrated significant potential for improving project management efficiency and transparency in software development for educational purposes. Kanban's visual workflow management approach has proven particularly effective in academic environments where multiple stakeholders need visibility into project progress and where requirements may evolve based on pedagogical insights and user feedback. The methodology's emphasis on continuous flow and work-in-progress (WIP) limits aligns well with the iterative nature of educational software development, where features must be tested and refined based on user experience and learning outcomes. Contemporary applications of Kanban in educational technology development have shown improved team collaboration, better resource allocation, and enhanced ability to respond to changing requirements (Asana, 2025).

Implementation of Kanban in educational software development requires careful adaptation to accommodate the unique characteristics of academic environments. Unlike traditional software development, educational technology projects often involve diverse stakeholder groups including educators, students, administrators, and technical staff, each with different perspectives on priorities and requirements. Kanban boards in educational contexts typically require additional customization to reflect the specific workflow stages relevant to educational software, such as pedagogical review, accessibility testing, and compliance validation. The methodology's flexibility in accommodating different team structures and processes makes it particularly suitable for academic environments where team composition may vary based on project phases and where part-time contributors (such as faculty advisors) may be involved (Atlassian, 2025).

The application of Kanban methodology to peer tutoring platform development offers several advantages, including improved transparency in development progress, better management of feature requests from educational stakeholders, and enhanced ability to prioritize tasks based on educational impact. The visual nature of Kanban boards facilitates communication between technical development teams and educational stakeholders who may not be familiar with traditional software development processes. For TechConnect and similar educational platforms, Kanban implementation can support the complex requirements management needed to balance technical feasibility with pedagogical effectiveness, ensuring that development priorities align with educational objectives while maintaining sustainable development practices (Inflectra, 2025).

Implementation Strategies

Scholars and practitioners emphasize that successful peer tutoring platforms require careful technical design coupled with iterative, stakeholder-driven rollout. For example, Kuo, Yao, and Wu (2022) describe a custom web-based peer-tutoring system (for programming) built with standard technologies (PHP and MySQL) that segregates users into tutor/tutee roles with tailored interfaces. Likewise, Reshetov (2025) introduces Slonig, an open-source peer tutoring app purpose-built with algorithmic tutor–tutee matching, structured feedback loops, and integrated tutor training to ensure quality and reduce teacher oversight. In practice, institutions often adopt a phased rollout: piloting the platform in select courses or student groups, then gradually expanding. For instance, the

University of New Hampshire launched a pilot in Fall 2021 focusing on gateway courses and support for first-generation/TRIO students; faculty and administrators co-designed the pilot so that it "complement[s] – not disrupt" existing supports, converting early skeptics into advocates once they saw tangible results. In this phase, active involvement of stakeholders (students, faculty, advisors) is critical. Dedicated outreach and training (for example, setting rigorous tutor qualifications and co-creating onboarding materials) help build trust. Reshetov (2025) specifically notes that peer-led onboarding (experienced students guiding newcomers through the system) enabled rapid adoption across different age groups. In summary, the literature recommends combining sound software engineering (scalable architecture, clear tutor–learner workflows) with participatory planning: engage student leaders, department heads, and IT staff early, pilot in targeted contexts, refine via feedback, and then scale campus-wide (Kuo et al., 2022; Reshetov, 2025). These collaborative, phased strategies help ensu**re** the technical platform fits local needs and gains acceptance before full deployment.

Deployment Practices

Deployment of a peer tutoring platform involves selecting appropriate hosting models, onboarding users effectively, and ensuring ongoing maintenance. Many modern solutions are offered as cloud-based services (Software-as-a-Service) or deployed on institutional servers to ensure reliability and scalability. For example, Schoolhouse.world – a widely used free peer tutoring platform founded by Khan Academy – operates through Zoom video conferencing for its live tutoring sessions, allowing 50,000+ students worldwide to connect with volunteer tutors without each institution building custom video infrastructure. Using cloud videoconferencing or shared LMS tools can dramatically reduce technical barriers. Regardless of hosting choice, administrators must plan for routine system maintenance: applying security patches, scaling server resources during peak demand, and monitoring uptime. Equally important is structured user onboarding and support. Training materials, FAQ guides, and orientation sessions (often delivered by experienced student "tutoring champions") can shorten the learning curve. Slonig's deployment, for instance, featured built-in tutor training modules and peer-led orientation, which "demonstrated rapid adoption and usability" in classrooms. In other cases, platforms have partnered with campus centers to recruit and train tutors, integrate the software into existing help centers, and provide 24/7 chat support for technical issues. Such onboarding fosters confidence: tutors know how to use the tools and abide by academic standards, while tutees learn how to request help. Finally, sustained success depends on an explicit maintenance plan. As one case study noted, outsourcing administrative tasks (like scheduling, payments, and compliance tracking) to the platform can relieve campus staff, but the institution should still allocate IT resources or vendor support for upgrades and troubleshooting. In practice, deployment best practices include choosing a secure, scalable hosting model; designing clear tutor/tutee onboarding workflows (often leveraging peers to mentor new users); and establishing ongoing technical support and data privacy safeguards. These measures help ensure the platform remains accessible and effective over time (Keller, 2024; Stand Together Trust, 2024).

ISO/IEC 25010 Quality Standards

The release of ISO/IEC 25010:2023 marked a significant evolution in software quality standards, introducing nine quality characteristics that are particularly relevant for educational technology platforms. The updated standard replaced the previous eight characteristics with an enhanced model that includes Functional Suitability, Performance Efficiency, Compatibility, Interaction Capability (formerly Usability), Reliability, Security, Maintainability, Flexibility (formerly Portability), and the newly added Safety characteristic. This expansion reflects the growing recognition of safety concerns in software systems, particularly relevant for educational platforms that handle sensitive student data and support critical learning processes. The updated standard provides more comprehensive coverage of quality attributes essential for modern educational technology systems, including enhanced considerations for accessibility, user engagement, and system adaptability (Arc42, 2023).

The 2023 revision introduced several important subcharacteristics that directly impact educational platform design and evaluation. Notable additions include inclusivity and self-descriptiveness under Interaction Capability, resistance under Security, and scalability under Flexibility. These enhancements reflect contemporary requirements for educational technology, where systems must support diverse user populations, provide intuitive interfaces, and scale to accommodate varying institutional needs. The standard's emphasis on user engagement (replacing user interface aesthetics) and faultlessness (replacing maturity) demonstrates a shift toward more user-centered and reliability-focused quality evaluation approaches that align well with the requirements of educational platforms serving critical academic functions (ISO, 2023).

For peer tutoring platforms like TechConnect, the updated ISO/IEC 25010:2023 standard provides a comprehensive framework for ensuring high-quality software that meets the diverse needs of educational environments. The inclusion of safety as a primary characteristic emphasizes the importance of protecting users from harm, whether through data breaches, inappropriate content exposure, or system failures that could impact learning outcomes. The enhanced focus on interaction capability, including inclusivity and self-descriptiveness, aligns with the need for educational platforms to serve diverse user populations with varying technical skills and accessibility requirements. Implementation of these quality characteristics throughout the development lifecycle can help ensure that peer tutoring platforms meet the high standards expected in educational technology while supporting effective learning outcomes (Pacific Certifications, 2024).

Functional Suitability and Performance Efficiency in Educational Software

Functional Suitability, as defined in ISO/IEC 25010:2023, encompasses three critical subcharacteristics: functional completeness, functional correctness, and functional appropriateness, all of which are essential for educational platforms supporting peer tutoring activities. Functional completeness ensures that the platform provides all necessary features to support effective tutoring interactions, including communication tools, resource sharing capabilities, scheduling systems, and progress tracking mechanisms. Functional correctness guarantees that these features operate reliably and produce accurate results, which is crucial for maintaining trust in educational technology systems. Functional appropriateness ensures that platform features are suitable for their intended educational purposes and facilitate rather than hinder the tutoring process (ISO25000, 2024).

Performance Efficiency in educational software contexts involves three key subcharacteristics: time behavior, resource utilization, and capacity. Time behavior is particularly critical for peer tutoring platforms, where delays in communication, file sharing, or system responses can significantly impact the quality of tutoring interactions. Educational platforms must maintain responsive performance even during peak usage periods, such as exam periods or assignment deadlines when tutoring demand typically increases. Resource utilization becomes important for institutions with limited IT infrastructure, requiring platforms to operate efficiently without excessive demands on network bandwidth, server resources, or client device capabilities. Capacity considerations ensure that platforms can scale to accommodate growing user populations and increasing data volumes without performance degradation (Codacy, 2023).

The implementation of Functional Suitability and Performance Efficiency standards in peer tutoring platforms requires comprehensive testing and validation processes throughout the development lifecycle. Performance testing should simulate realistic usage scenarios, including concurrent user sessions, large file transfers, and high-volume communication activities typical in academic environments. Functional testing must validate not only technical correctness but also pedagogical effectiveness, ensuring that platform features genuinely support learning objectives and tutoring effectiveness. For TechConnect, adherence to these quality characteristics means implementing robust performance monitoring, conducting regular capacity planning, and maintaining comprehensive functional testing suites that validate both technical functionality and educational effectiveness (Monterail, 2023).

Interaction Capability and Security in Educational Technology

The Interaction Capability characteristic in ISO/IEC 25010:2023 has evolved significantly from the previous Usability concept to encompass broader user experience considerations essential for educational platforms. This characteristic includes appropriateness, recognizability, learnability, operability, user error protection, user engagement, inclusivity, and self-descriptiveness. For peer tutoring platforms, appropriateness recognizability ensures that users can quickly identify whether the platform suits their tutoring needs, while learnability focuses on how easily new users can become proficient with the system. User engagement, a new addition replacing user interface aesthetics, emphasizes the platform's ability to maintain user interest and motivation throughout tutoring sessions. Inclusivity ensures that the platform can be used effectively by individuals with diverse characteristics and capabilities, including users with disabilities or varying technical expertise (ISO25000, 2024).

Security in educational technology platforms encompasses six critical subcharacteristics: confidentiality, integrity, non-repudiation, accountability, authenticity, and resistance. Confidentiality is paramount in educational contexts where sensitive student information, academic records, and private tutoring conversations must be protected from unauthorized access. Integrity ensures that educational content, user communications, and assessment data remain accurate and unmodified by unauthorized parties. Non-repudiation and accountability provide essential audit trails for academic integrity purposes, enabling institutions to verify the authenticity of tutoring interactions and maintain academic standards. The resistance subcharacteristic, newly added in the 2023 revision, addresses the platform's ability to withstand various forms of attacks and maintain security under adverse conditions (QMII, 2024).

For peer tutoring platforms operating in educational environments, the implementation of robust Interaction Capability and Security measures is essential for ensuring user adoption and institutional compliance. Interaction Capability requirements necessitate user-centered design approaches, comprehensive usability testing with diverse user populations, and continuous refinement based on user feedback and engagement metrics. Security implementation requires comprehensive risk assessment, implementation of appropriate security controls, regular security auditing, and compliance with educational data protection regulations such as FERPA in the United

States or GDPR in European contexts. The integration of these quality characteristics throughout the platform development lifecycle helps ensure that peer tutoring systems provide both excellent user experiences and robust protection of sensitive educational data (Pacific Certifications, 2024).

Conceptual Framework

The conceptual framework for TechConnect illustrates the theoretical foundation and relationships between key components that guide the development of a peer tutoring platform for the College of Industrial Technology. At its core, the framework recognizes three primary stakeholders: administrators who oversee system operations, tutors who provide academic support, and tutees who seek assistance. These stakeholders interact through a technology-mediated environment that facilitates peer learning through structured processes.

The framework is grounded in Vygotsky's Zone of Proximal Development theory, which emphasizes that students learn most effectively when guided by more knowledgeable peers. This theoretical foundation drives the platform's matching mechanism, which connects students with tutors who possess the specific expertise needed. The framework also incorporates principles of self-directed learning, recognizing that while technology can empower student autonomy, structured mentorship remains crucial for effective learning outcomes.

Central to this framework is the concept of intelligent matching, which combines rule-based filtering with fuzzy logic to create meaningful tutor-tutee connections. This matching process considers multiple factors including subject expertise, availability, learning preferences, and performance ratings to optimize learning partnerships. The framework acknowledges that successful peer tutoring requires not just technical matching but also social and pedagogical alignment between participants.

The conceptual framework further recognizes the importance of incentive structures in sustaining peer tutoring programs. By incorporating recognition mechanisms such as certificates, badges, and portfolio-building opportunities, the platform addresses the challenge of maintaining consistent tutor participation. This aligns with motivational theories that emphasize both intrinsic rewards (knowledge reinforcement, leadership development) and extrinsic benefits (formal recognition, resume enhancement).

The framework positions technology as an enabler rather than a replacement for human interaction. While digital tools provide efficiency in scheduling, resource sharing, and communication, the essence of peer tutoring remains the meaningful exchange between students. This human-centered approach ensures that TechConnect enhances rather than diminishes the personal connections that make peer learning effective.

Research Paradigm

A research paradigm shows how the different parts of a study are related. It outlines the research process and how these components interact to ensure consistent findings. The diagram below presents the step-by-step process involved in developing TechConnect: A Peer Tutoring Platform for the College of Industrial Technology.

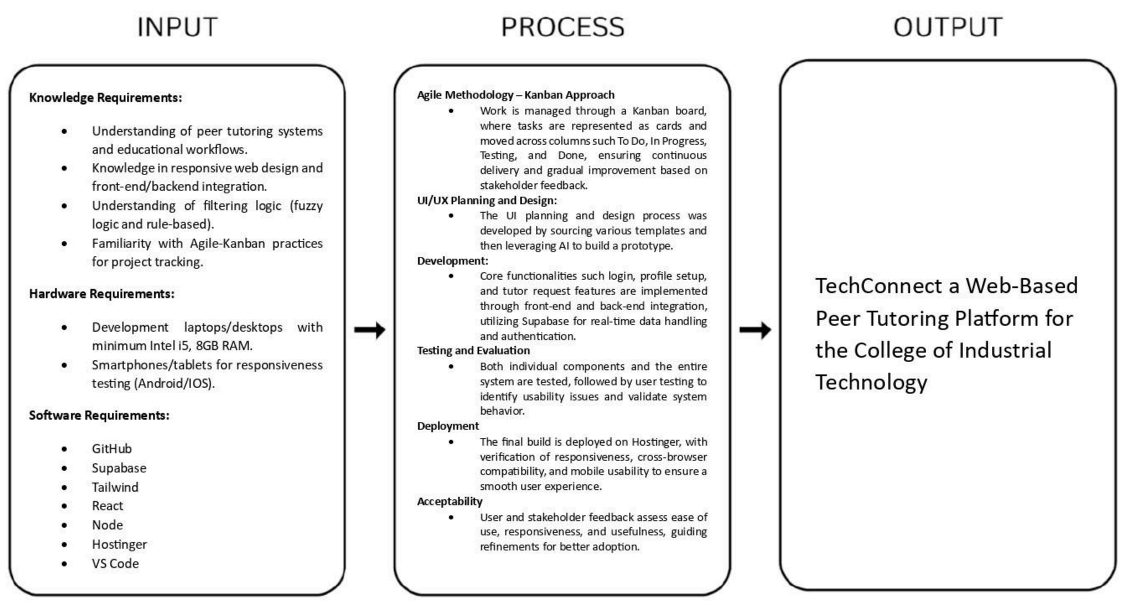

Figure 1. *Input, Process, Output (IPO) Model of the Study*

The development of TechConnect requires a comprehensive set of inputs spanning both technical and conceptual domains. On the knowledge side, the project team must have expertise in peer tutoring systems, educational workflows, and learner support practices. Technical competencies include proficiency in responsive web development, full-stack integration, and intelligent filtering techniques such as fuzzy logic and rule-based matching. Familiarity with Agile–Kanban practices is also essential for efficient project tracking and iterative improvement.

Hardware requirements include development laptops or desktops with at least Intel i5 processors and 8GB of RAM, along with smartphones or tablets (Android/iOS) for responsiveness and usability testing. Software requirements cover modern development tools such as React with Next.js for front-end implementation and routing, Node.js for server-side logic, Supabase for real-time database and authentication, and Tailwind CSS for responsive UI styling. Additional tools include GitHub for version control, Visual Studio Code as the primary IDE, and Hostinger for deployment and web hosting.

The development process follows the Agile–Kanban methodology, with tasks visualized on Kanban boards categorized into "To Do," "In Progress," "Testing," and "Done." UI/UX design is developed through sourced templates enhanced with AI-assisted prototyping. Core functionalities—including login and authentication, tutor profile setup, and request management—are implemented through the integration of Next.js front-end logic with a Supabase backend for data handling and secure authentication. Testing and evaluation involve unit, integration, and user testing to validate reliability and usability. Deployment ensures cross-browser compatibility, HTTPS compliance, responsiveness, and overall system stability. Acceptability testing through user and stakeholder feedback informs refinements and supports long-term adoption.

The output of this process is TechConnect, a web-based peer tutoring platform for the College of Industrial Technology. Built with a modern stack of React, Next.js, Node.js, and Supabase, and guided by Agile–Kanban principles, the platform delivers intelligent tutor–tutee matching and responsive design, ensuring accessible, reliable, and effective peer-assisted learning.

Chapter III

METHODOLOGY

This chapter covers the Research Locale, Research Design, Requirement Analysis, Requirement Documentation, Software Design, System, Product, and Process Design, as well as the Development and Testing Procedures. It also includes a Description of the Prototype, Implementation Plan, Research Instruments, Study Respondents, and Statistical Treatment. Additionally, this chapter outlines the necessary materials and methods used to develop the software product, including UML diagrams that offer a clearer understanding of the system.

Research Locale

The study was conducted at the College of Industrial Technology (CIT) of Southern Luzon State University (SLSU). TechConnect is specifically tailored to the needs of CIT students and faculty. Since SLSU already has a general academic support program, this project focuses on a specialized departmental peer tutoring solution. All development, testing, and evaluation activities for the platform are centered in this academic environment.

Research Design

This study employed a developmental research design, as the primary goal is to design, build, and evaluate TechConnect, a web-based peer tutoring platform tailored to the needs of the College of Industrial Technology (CIT) at Southern Luzon State University (SLSU). Developmental research is appropriate because it focuses on producing a functional system that directly addresses real-world academic challenges and validating its effectiveness through iterative design, testing, and refinement. The development process is guided by an Agile–Kanban approach, which supports flexibility, continuous improvement, and responsiveness to user feedback, ensuring that each development cycle delivers incremental progress while allowing for adjustments based on the needs of CIT students, tutors, and administrators.

TechConnect is designed to resolve the limitations of the existing Peer Assisted Learning (PAL) program by providing a structured and specialized academic support platform. Its features include tutor–tutee matching based on subject tagging, expertise, and availability; flexible scheduling and integration with Google Calendar and Google Meet; and centralized resource management through a digital library for study materials and session notes. The system also defines distinct roles such as “Study Buddy” for informal peer support and “Learning Buddy” for structured tutoring sessions, while incorporating incentive mechanisms like certificates, resume-building opportunities, and formal recognition to encourage sustained tutor participation. Furthermore, the platform equips administrators with tools for monitoring sessions, posting announcements, and managing reports, thereby enhancing institutional oversight and coordination.

To ensure quality and reliability, the platform will be evaluated using the ISO 25010 software quality model, focusing on functional suitability, performance efficiency, compatibility, interaction capability, reliability, security, maintainability, flexibility, and safety. The evaluation process will involve feedback from stakeholders—CIT students, tutors, faculty members, and administrators—during pilot implementation. Their input will guide refinements to the system's features, user interface, and performance. By integrating modern web technologies such as React, Node.js, Supabase, and related APIs with the specific academic requirements of CIT, TechConnect not only strengthens peer-assisted learning but also establishes a scalable model for academic support initiatives within SLSU.

Requirements Analysis

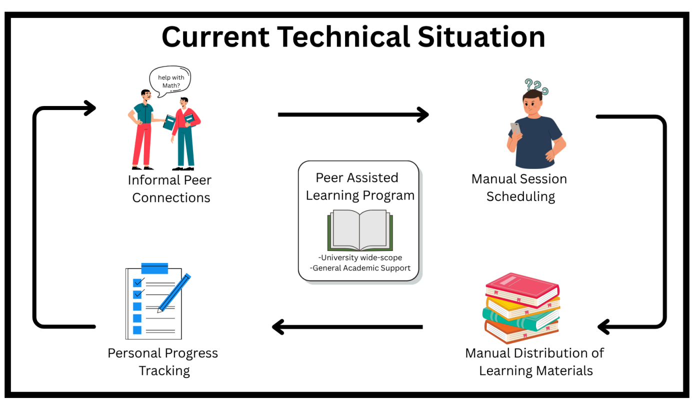

Figure 2. *Current Technical Situation of Peer Tutoring at Southern Luzon State University*

The peer-tutoring landscape at Southern Luzon State University (SLSU) rests almost entirely on informal networks and manual coordination, creating challenges that ripple through student support. Learners who need assistance typically wait for a classmate's recommendation or chance upon a bulletin-board post before they can secure a tutor. While the Learning Development Center's Peer-Assisted Learning (PAL) programme does provide general academic help, it operates without a central calendar, automated sign-up form, or notification system. Sessions are announced sporadically and filled on a first-come, first-served basis, leaving many students uncertain about availability and next steps. This ad-hoc model may work for small study circles, but it cannot guarantee consistent tutor quality, subject-specific matching, or reliable record-keeping as the university's enrolment grows. Incorporating a smart-matching platform that pairs tutees with tutors according to verified expertise and course requirements would streamline operations, improve service quality, and allow SLSU to track outcomes and refine its support strategies.

Requirement Documentation

The functional requirements of TechConnect describe the system's capabilities and expected behavior, focusing on the delivery of an accessible and efficient peer tutoring environment for the College of Industrial Technology. At its core, the platform provides user account management, tutor–tutee matching, a centralized resource hub, dashboards for activity tracking, communication and notification services, announcements and events, incentive mechanisms, and administrative oversight features.

- **User Account Management**

  For user account management, enrolled CIT students are able to register using either a university-issued or personal email address, secured by a password and verified through a confirmation link. During registration, students select a role as either a tutor—classified as a "Study Buddy" for informal support or a "Learning Buddy" for structured sessions—or as a tutee. Users may later edit their personal details, while tutors specify subject expertise through tags and set availability on a calendar interface. Tutees, in turn, identify the subjects they need help with and preferred time slots. Administrator accounts are created manually, granting them the ability to manage users, reset accounts, and access dashboards summarizing overall system activity.

- **Tutor & Tutee Matching**

  Tutor–tutee matching is accomplished through a two-stage process that combines rule-based and fuzzy search logic. Initially, the system pairs users based on subject tags, availability, and preferred tutoring type. In cases where no exact match exists, fuzzy string matching is applied to identify close alternatives, ranking results by similarity and availability. Tutees are presented with a list of potential tutors along with their subject expertise, availability, and a brief description, from which they may request sessions. Tutors receive in-app and email notifications of these requests and may accept or decline within a specified timeframe. Confirmed sessions are added to the calendar, while completed sessions are followed by tutee feedback, which contributes to tutor performance metrics.

- **Resource Hub & Materials**

  The resource hub serves as a centralized repository where tutors may upload study materials, including documents, presentations, images, and compressed files. Metadata such as uploader, timestamp, subject tags, and visibility scope are stored alongside the resources to facilitate indexing and retrieval. Tutees and tutors alike may access the repository through a searchable interface that supports filters, bookmarking, downloads, and feedback ratings. Engagement data, such as download counts and ratings, are also tracked for quality monitoring.

- **Dashboard & Activity Tracking**

  Dashboards are tailored according to user roles. Tutors access tools to manage availability, view upcoming sessions, track performance ratings, and monitor progress toward earning certificates. Tutees can review pending and confirmed sessions, track learning progress, and access bookmarked materials, while administrators are provided with a consolidated view of overall system usage, session statistics, and resource engagement, with the ability to generate downloadable reports.

- **Communication & Notification**

  Communication is facilitated through both in-app. These include session confirmations, reminders, account verifications, feedback requests, and periodic tutor summaries. A unified notification center ensures that users remain informed about their activities and institutional announcements.

- **Announcements & Events**

  Administrators can further publish announcements or academic events with details such as title, content, validity period, and optional attachments. Events are displayed through a searchable calendar, while pinned announcements remain visible across all dashboards.

- **Incentives & Recognition**

  Tutor incentives and recognition mechanisms are embedded into the system to sustain engagement. Tutors who complete a defined number of sessions with positive ratings automatically receive a downloadable certificate, while badges such as “Top Tutor” or “Resource Contributor” are awarded based on participation milestones. In addition, tutors may generate a credential summary that documents their tutoring history and performance for inclusion in résumés or academic portfolios.

- **Administrative & Departmental Features**

  Administrative and departmental functions provide full oversight of platform activity. Administrators can search and manage user accounts, monitor session outcomes, review uploaded resources, and oversee event postings. Audit logs record administrative actions for transparency, while report generation tools support decision-making by offering analytics on user engagement, tutor performance, and resource utilization.

- **System & Infrastructure Context**

  From a system and infrastructure perspective, TechConnect is built using React with Next.js framework for the front end, with server-side logic handled by Node.js functions hosted in a Hostinger environment. Supabase serves as the primary backend for authentication, database management, file storage, and real-time updates, ensuring secure, reliable, and scalable performance. All data is transmitted over HTTPS and encrypted at rest, while indexing and pagination are implemented to maintain query performance even under high loads. A continuous integration and deployment pipeline is linked to the project's repository, enabling automated deployment of updates and consistent delivery of new features.

Design of Software, System, Product and Process

System Architecture

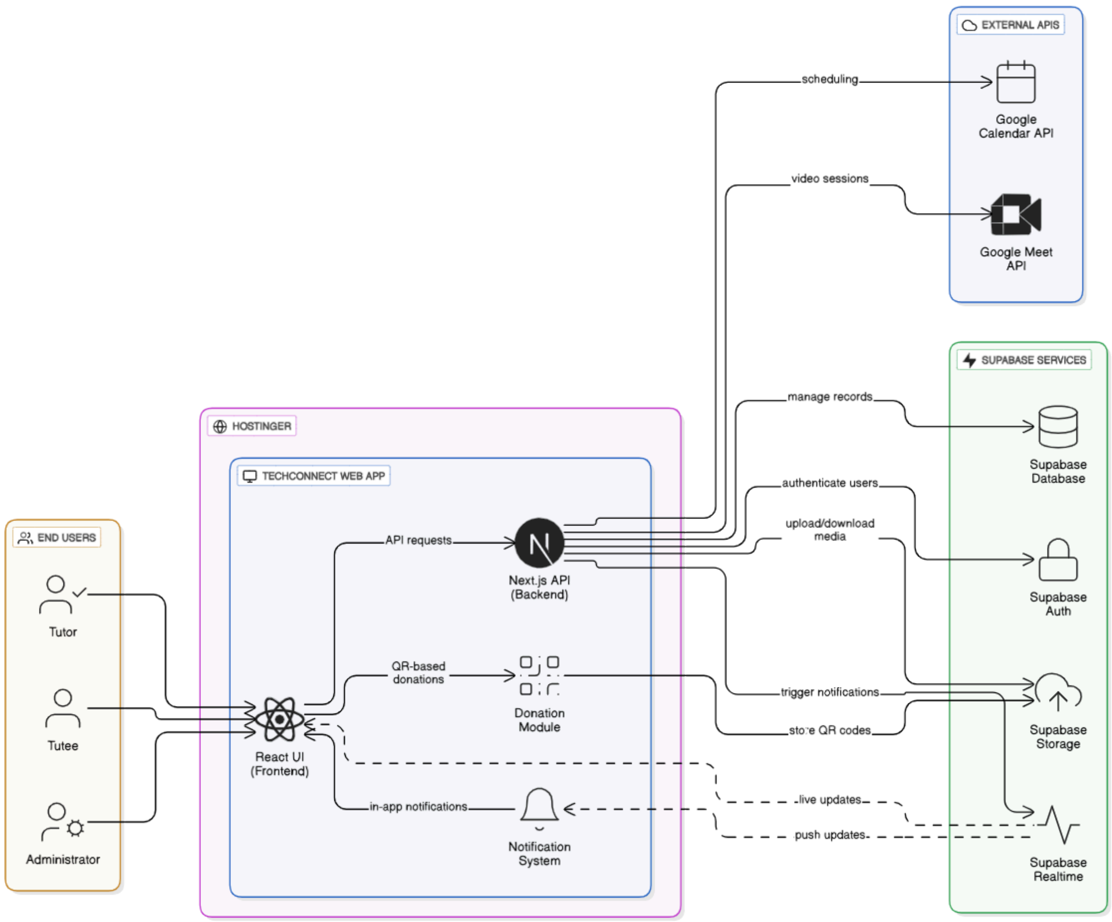

Figure 3. *System Architecture Design of TechConnect*

The system architecture of TechConnect illustrates how the platform connects Administrators, Tutors, and Tutees through a web-based application accessible on both desktop and mobile browsers. The application is hosted on Hostinger and built with a React front end styled with Tailwind CSS, supported by a Next.js backend that processes requests and integrates with external services. Supabase serves as the backend-as-a-service layer, providing authentication, database management, file storage, and real-time updates. The React user interface enables users to manage profiles, browse tutors, schedule tutoring sessions, access learning materials, provide feedback, and view events, while the Next.js backend routes these interactions to Supabase and external APIs. A donation module allows QR-based contributions, and a notification system powered by Supabase Realtime delivers in-app alerts to Tutees, Tutors, and Administrators for session requests, approvals, cancellations, and other important updates. Supabase Auth secures login and account management, the Supabase Database stores user accounts, session records, feedback, learning materials, and donation logs, Supabase Storage handles file uploads such as learning materials and QR codes, and Supabase Realtime ensures live synchronization and notifications. The Next.js backend also connects with Google Meet for video conferencing and Google Calendar for session scheduling and reminders, ensuring smooth communication and time management. Administrators access the platform through the Admin Dashboard to verify tutor accounts, manage users, oversee learning materials, monitor sessions, and update donation QR codes. Tutors set availability, update profiles, manage and upload learning materials, and accept or decline tutoring requests, while Tutees browse tutors, access learning resources, manage sessions, submit feedback, and participate in scheduled events. A typical flow begins when a user interacts with the React interface, which sends a request to the Next.js backend. The backend authenticates the user through Supabase Auth, processes the request by reading or writing to the Supabase Database, and triggers real-time updates through Supabase Realtime. If the request involves a tutoring session, the system generates a Google Calendar event and a Google Meet link, while uploaded files such as learning materials or QR codes are stored in Supabase Storage and served to authorized users. Through this architecture, TechConnect ensures secure authentication, reliable data management, scalable storage, efficient scheduling, and effective in-app communication, providing a streamlined and responsive peer tutoring experience.

Software Methodology

Figure 4. *Agile Kanban Methodology*

The development of TechConnect followed an Agile–Kanban methodology, which provided a flexible and iterative framework for building, testing, deploying, and refining the platform. All development tasks began in a central backlog, where requirements related to peer tutoring practices in the College of Industrial Technology were collected and refined. These requirements were identified through interviews with faculty members and students, a review of Learning Development Center documents, and observations of pilot tutoring arrangements. Each identified need, such as resolving scheduling inefficiencies or defining reporting metrics for administrators, was represented as a backlog card. A card remained in the backlog until it was fully defined and validated by stakeholders and the project adviser.

Once approved, a card moved into the combined design and development flow. At this stage, wireframes and high-fidelity prototypes were created using AI, with emphasis on designing intuitive interfaces tailored to administrators, tutors, and tutees. Visual decisions concerning typography, icons, and color schemes ensured a consistent and professional platform identity. After peer review and approval, cards advanced to the implementation phase, where developers translated the designs into functional features using React components structured within the Next.js framework for the front end and Node.js serverless functions for backend logic. The platform utilized Supabase for authentication, database management, file storage, and real-time updates, ensuring secure and scalable performance. Each feature was implemented as a self-contained user story, such as enabling tutors to define availability slots or developing the algorithm for session booking and tutor–tutee matching. Continuous integration automatically deployed changes to the Hostinger environment, allowing for rapid updates with minimal downtime.

Testing began as soon as new code was deployed in the staging environment. Quality assurance was performed through unit testing of individual components, ensuring that forms rejected invalid input, data persisted correctly, and matching algorithms produced accurate results. Integration testing validated full workflows such as booking a session, notifying the tutor, and recording feedback. Cross-device testing was conducted to confirm responsiveness across desktops, laptops, and mobile browsers. Any issues identified during testing generated new backlog cards categorized as “Bug” or “Enhancement,” which were prioritized alongside new feature development. A card was only marked “Done” once it satisfied all acceptance criteria and passed both functional and usability tests.

Deployment activities, including server configuration updates or Supabase security rule adjustments, were also managed within the same Kanban workflow. Once a deployment card reached completion, updates were released to production immediately. This continuous deployment model allowed small, incremental updates throughout the development cycle, minimizing risks associated with large-scale releases and ensuring that improvements reached users quickly.

Weekly workflow reviews were conducted to evaluate development efficiency and identify potential bottlenecks. Metrics such as cycle time—the duration for a card to move from development to completion—and throughput—the number of completed cards per week—were closely monitored. If delays were observed in testing, the team adjusted priorities or temporarily allocated more resources to restore balance. Stakeholder feedback from pilot users within CIT played a central role in this process. Surveys, usability test results, and reports from administrators were analyzed to generate new backlog cards that addressed evolving requirements.

By integrating requirement analysis, design, development, testing, deployment, and review into a single continuous workflow, the Agile–Kanban methodology ensured that TechConnect remained adaptable, transparent, and efficient. This iterative process not only delivered a functional platform but also enabled continuous improvement guided by user feedback and evaluation against ISO 25010 quality standards.

Software Requirements

The development of TechConnect requires specific software tools and technologies to ensure optimal functionality, performance, and user experience throughout the project lifecycle. These carefully selected software requirements form the technical foundation that enables the researchers to design, develop, test, and deploy a web-based system.

Visual Studio Code. The researchers utilized Visual Studio Code as their integrated development environment because of its extensibility, built-in terminal, and wide range of plugins that support JavaScript and React development.

Node.js and npm. Node.js was used as the JavaScript runtime environment, while npm served as the package manager for installing dependencies, running development servers, and building the React application.

Tailwind CSS. Tailwind CSS was employed to design a responsive and consistent user interface. Its utility-first classes allowed rapid prototyping and streamlined design iterations while keeping the codebase lightweight and maintainable.

React. React was adopted as the front-end framework for building the user interface. Its component-based architecture, virtual DOM optimization, and large community support made it suitable for developing a scalable and efficient platform.

Supabase. Supabase was used as the primary back-end service, providing authentication, database management, file storage, and real-time updates. This ensured secure user sign-in, reliable data handling, and offline persistence while maintaining scalability.

Git and GitHub. Git was employed for version control to track changes and manage collaborative development through feature branches. Integration with GitHub supported centralized repository management, pull requests, and continuous deployment pipelines.

Hostinger. Hostinger served as the cloud hosting provider for both the React front-end and Node.js serverless functions. It provided managed SSL encryption, automatic scaling, and reliable uptime for production deployment.

Google Chrome. Google Chrome was used as the primary web browser for testing and debugging, due to its comprehensive developer tools, performance profiling features, and cross-platform availability.

Hardware Requirements

Personal Computer/Laptop. A 64-bit Microsoft Windows 10 (or later) or macOS workstation with at least 8 GB of RAM, an x86_64 CPU architecture (Intel Core i5 or equivalent AMD processor), and a minimum screen resolution of 1280 × 800. This configuration was necessary to simultaneously run the code editor, development server, Supabase services, and browser testing tools without performance issues.

Stable Internet Connection. A broadband internet connection of at least 10 Mbps download and 5 Mbps upload speed was required to interact with cloud-based services, clone repositories, and deploy updates to Hostinger efficiently.

Mobile Device (Android or iOS). A modern smartphone or tablet running Android 8.0 (or later) or iOS 12 (or later) is recommended to test responsive design, cross-device compatibility, and user experience on smaller screens.

Use Case Diagram

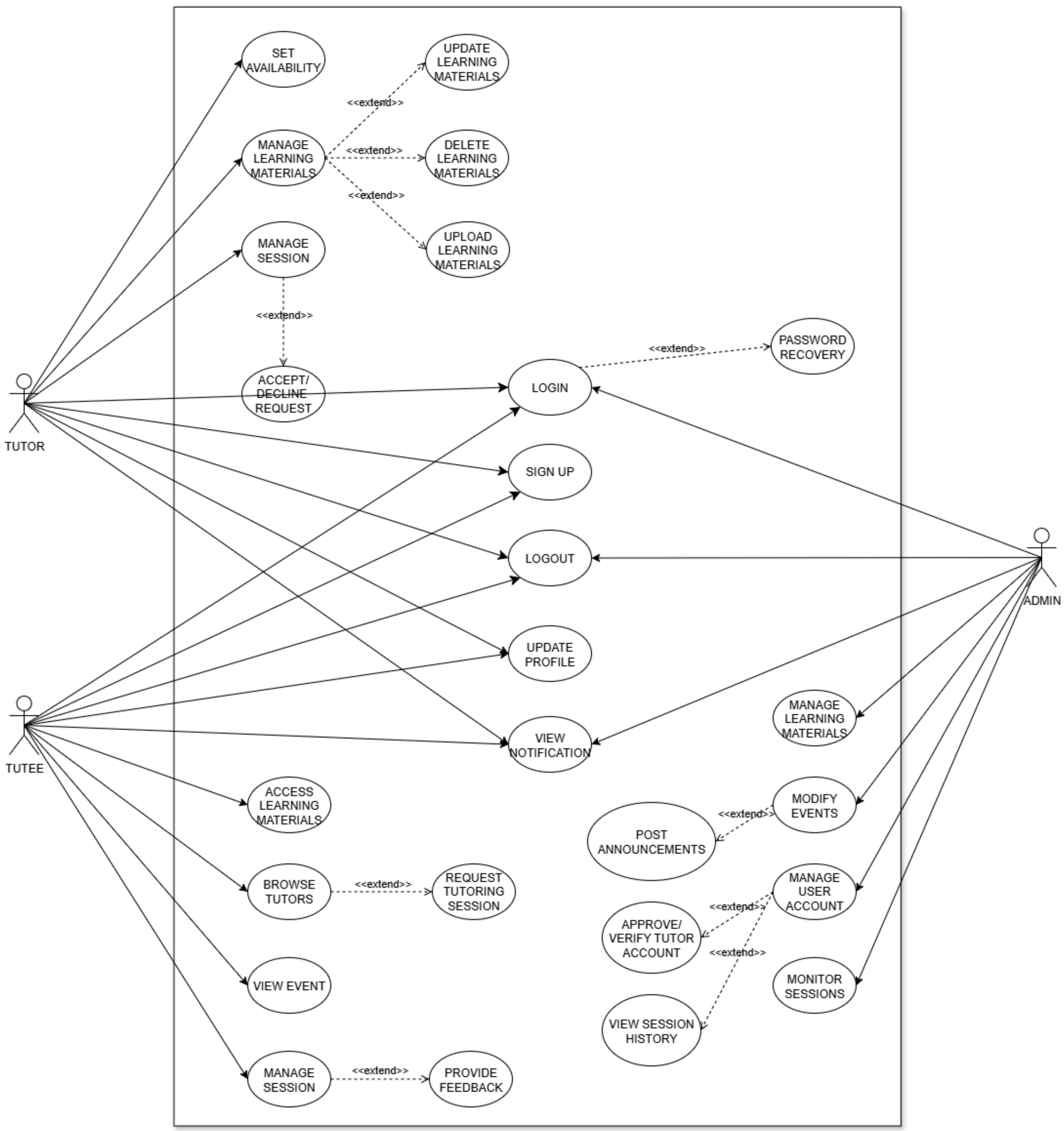

Figure 5. *Use Case Diagram*

The use case diagram presented in Figure 5 illustrates the primary functionalities of the TechConnect platform and delineates how its three user roles—Administrator, Tutor, and Tutee—interact with the system. Serving as a blueprint for the platform's functional requirements, the diagram clarifies the scope of user activities and their relationship to system processes.

The Administrator exercises the highest level of control, with responsibilities encompassing the management of user accounts, verification and approval of tutor profiles, oversight of learning materials, modification of events, posting of official announcements, and monitoring of tutoring sessions. Through these functions, the Administrator ensures that the platform maintains quality, reliability, and operational integrity.

The Tutor role focuses on delivering academic support. Tutors are empowered to manage their personal profiles, set availability, accept or decline tutoring requests, and upload, update, or delete supplementary learning materials. In addition, they receive notifications and feedback from tutees, which provide insights into their performance and contribute to continuous improvement as peer mentors.

The Tutee represents the learner seeking academic assistance. Their interactions include account registration, login, browsing tutor profiles, requesting tutoring sessions, accessing learning resources, managing scheduled sessions, viewing events, and providing feedback after each session. Notifications further support the Tutee by keeping them informed of session updates, announcements, and resource availability.

Activity Diagram

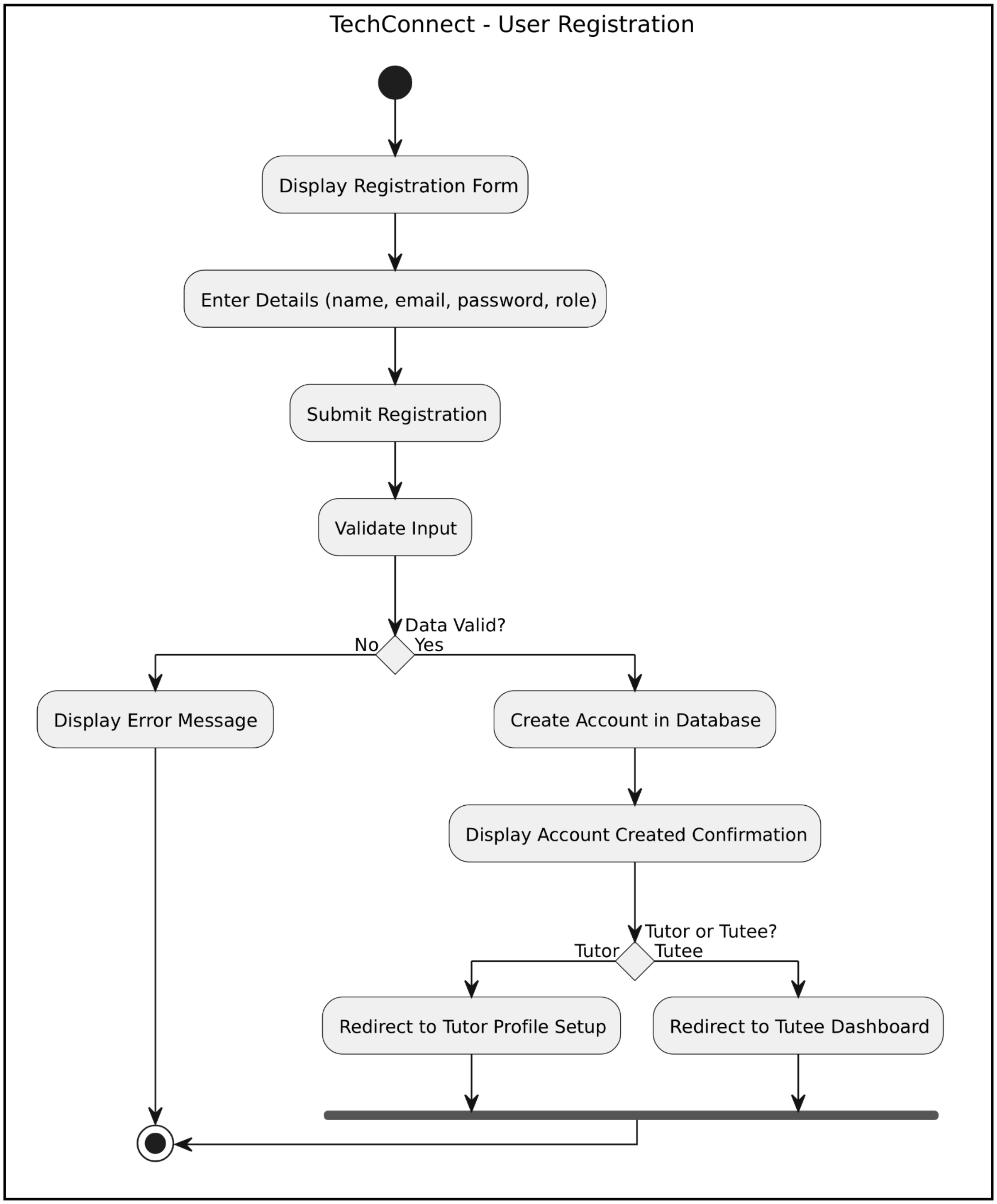

Figure 6. *User Registration*

This activity diagram illustrates how a new user (tutee or tutor) creates an account on the TechConnect platform. It covers entering registration details, system validation, account creation, and directing the user to the next step (dashboard or tutor profile setup).

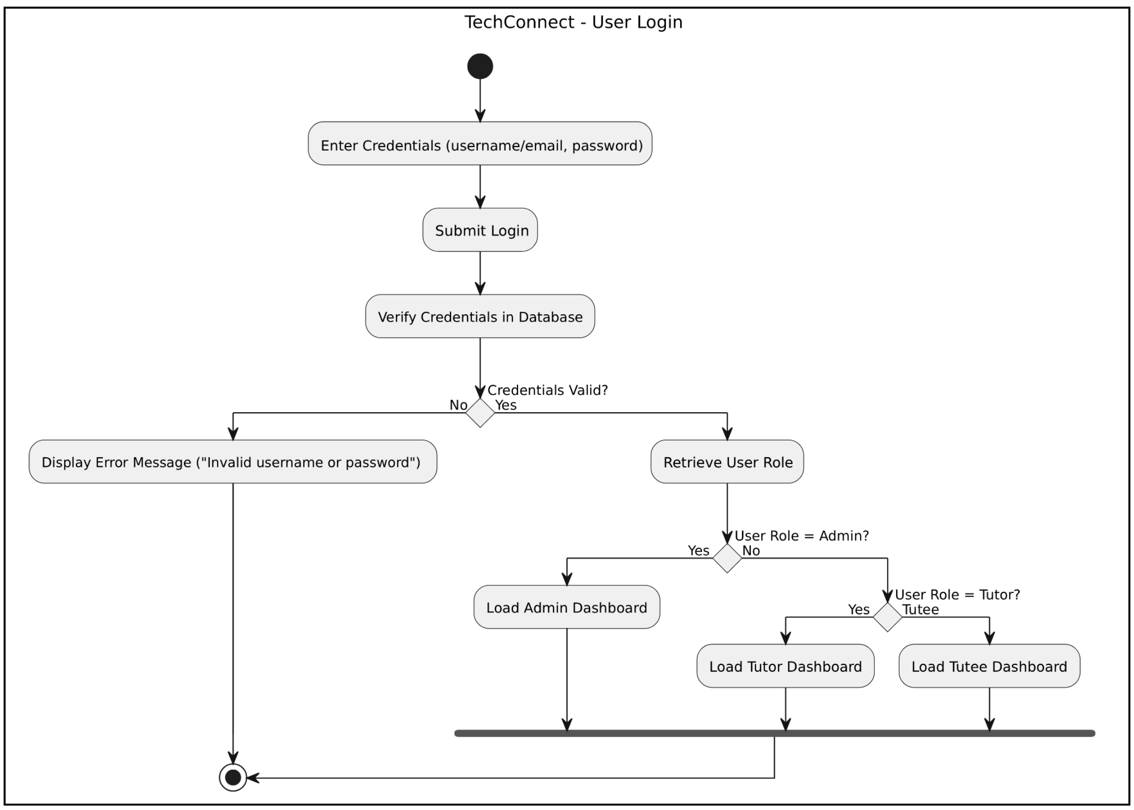

Figure 7. *User Login*

This diagram shows the login procedure for a user (admin, tutor, or tutee) accessing the platform. It includes entering credentials, system authentication, role-based dashboard loading, or an error for invalid login.

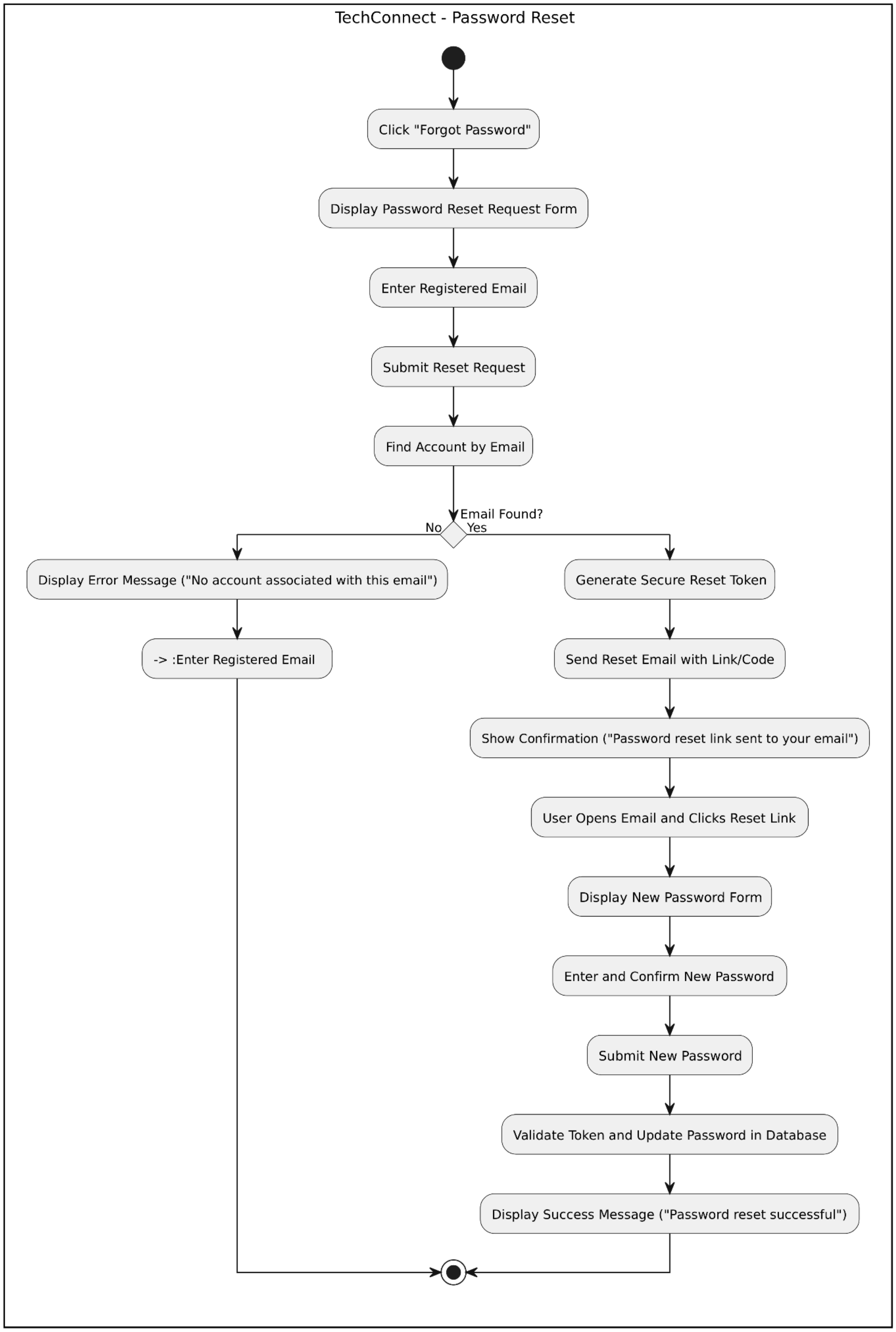

Figure 8. *Password Reset*

This activity diagram details the password recovery process for a user who has forgotten their password. It involves the user requesting a reset, the system sending a reset link via email, and the user setting a new password.

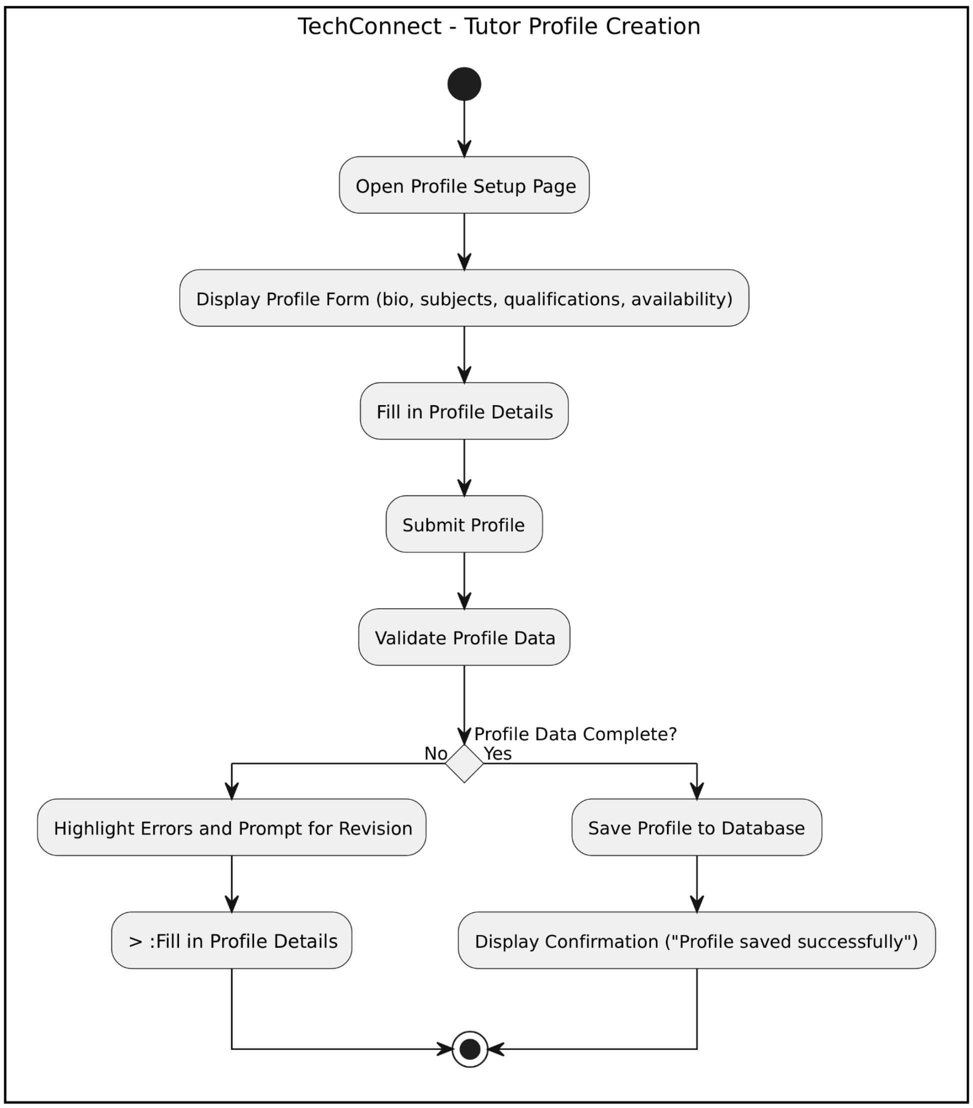

Figure 9. *Tutor Profile Creation*

This diagram covers the process of a tutor setting up their profile after registration. It includes the tutor entering details like qualifications, subjects, and availability, and the system saving the profile.

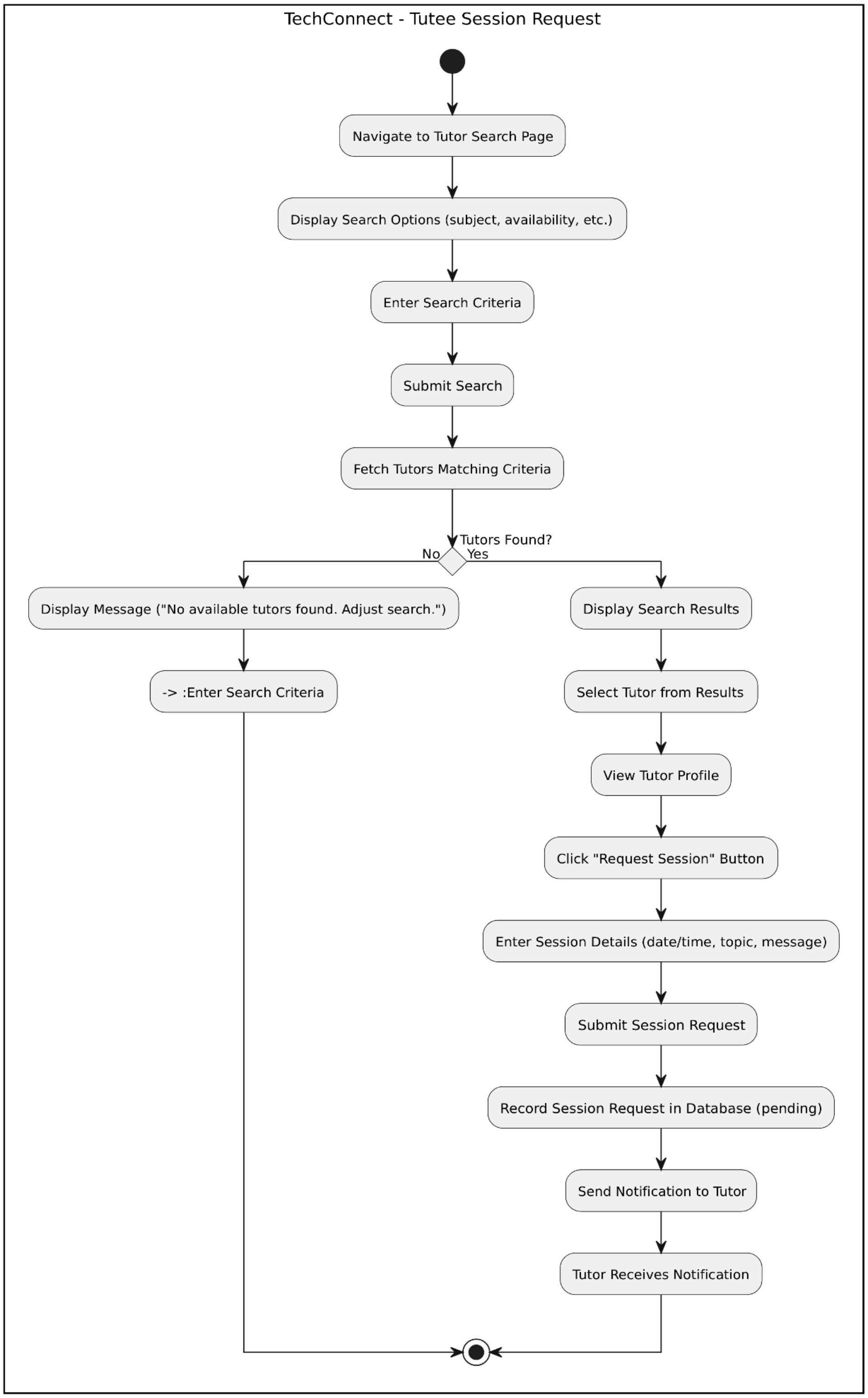

Figure 10. *Tutee Session Request*

This activity diagram outlines how a tutee searches for a tutor and requests a tutoring session. It covers searching by subject, viewing tutor details, selecting a time, and sending a request.

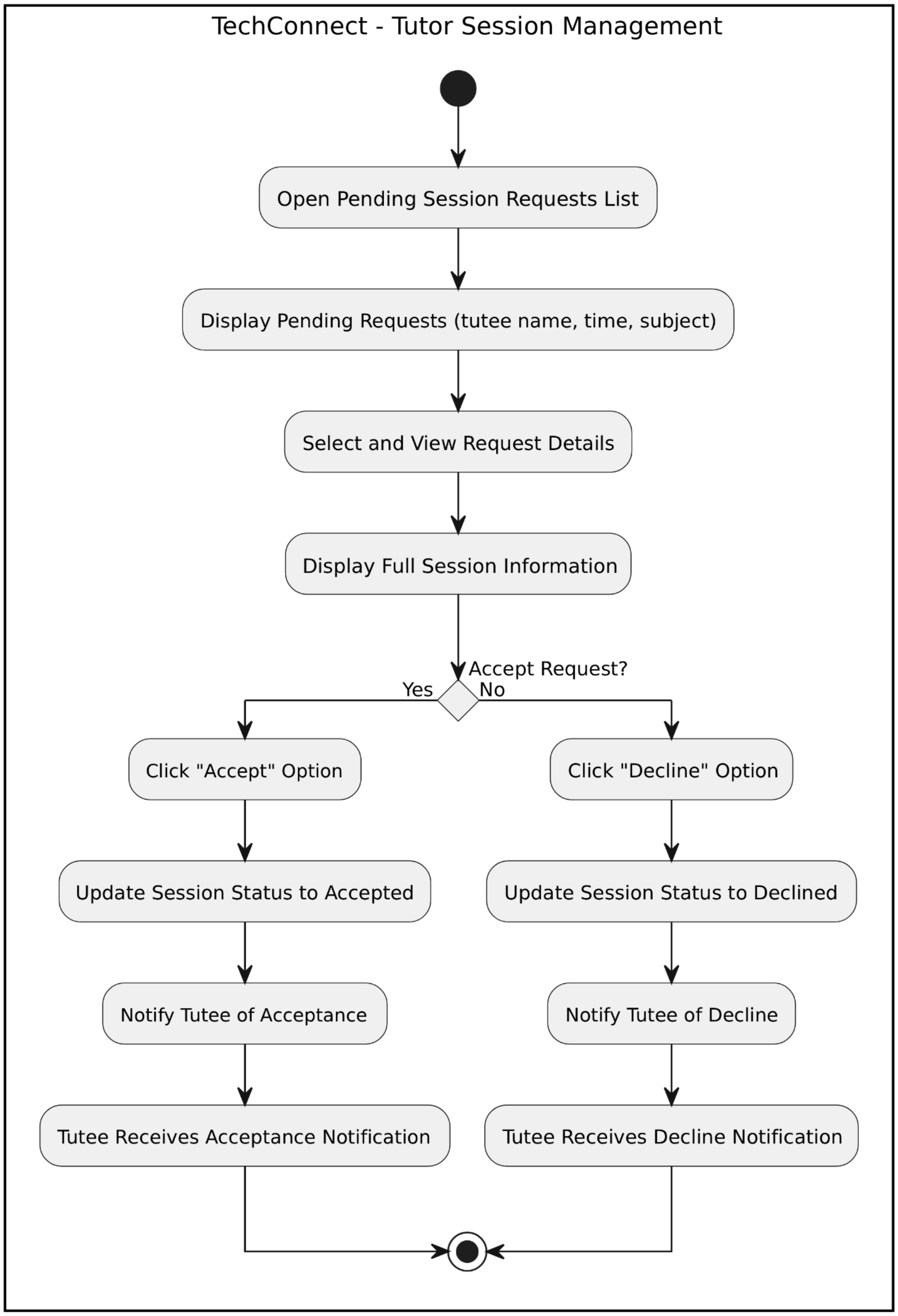

Figure 11. *Tutor Session Management*

This diagram shows how a tutor manages incoming session requests, including viewing request details and choosing to accept or decline the request. It also depicts the system updating the session status and notifying the tutee of the tutor's decision.

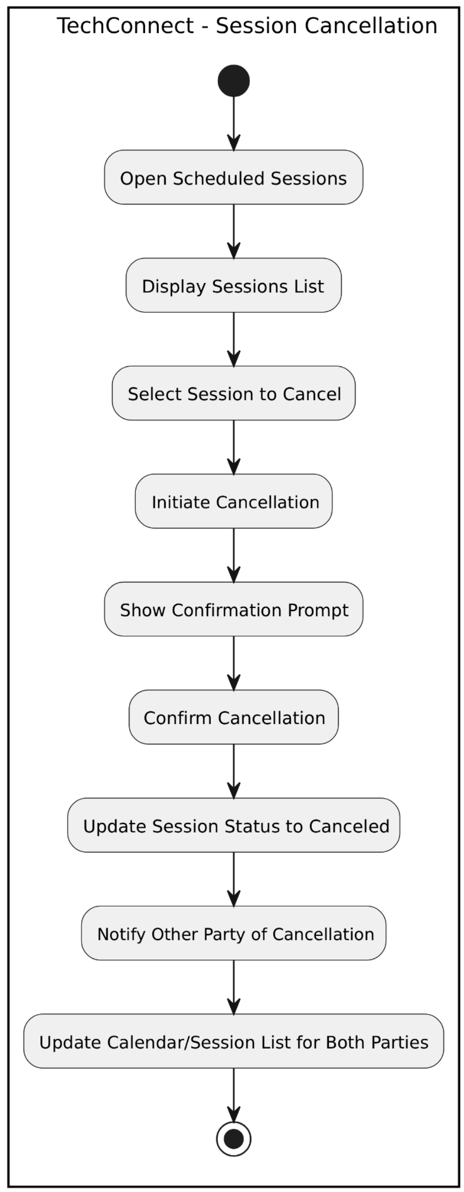

Figure 12. *Session Cancellation*

If a scheduled session needs to be canceled, this diagram illustrates the cancellation process. It shows a tutee initiating a cancellation (tutors could similarly do so), the system updating the session status, and notifications sent to the other party.

A similar flow would occur if the Tutor needs to cancel a session, in that case the process starts in the Tutor lane and the tutee receives the notification. The roles would be reversed, but the system actions (update status, notifications) remain the same.

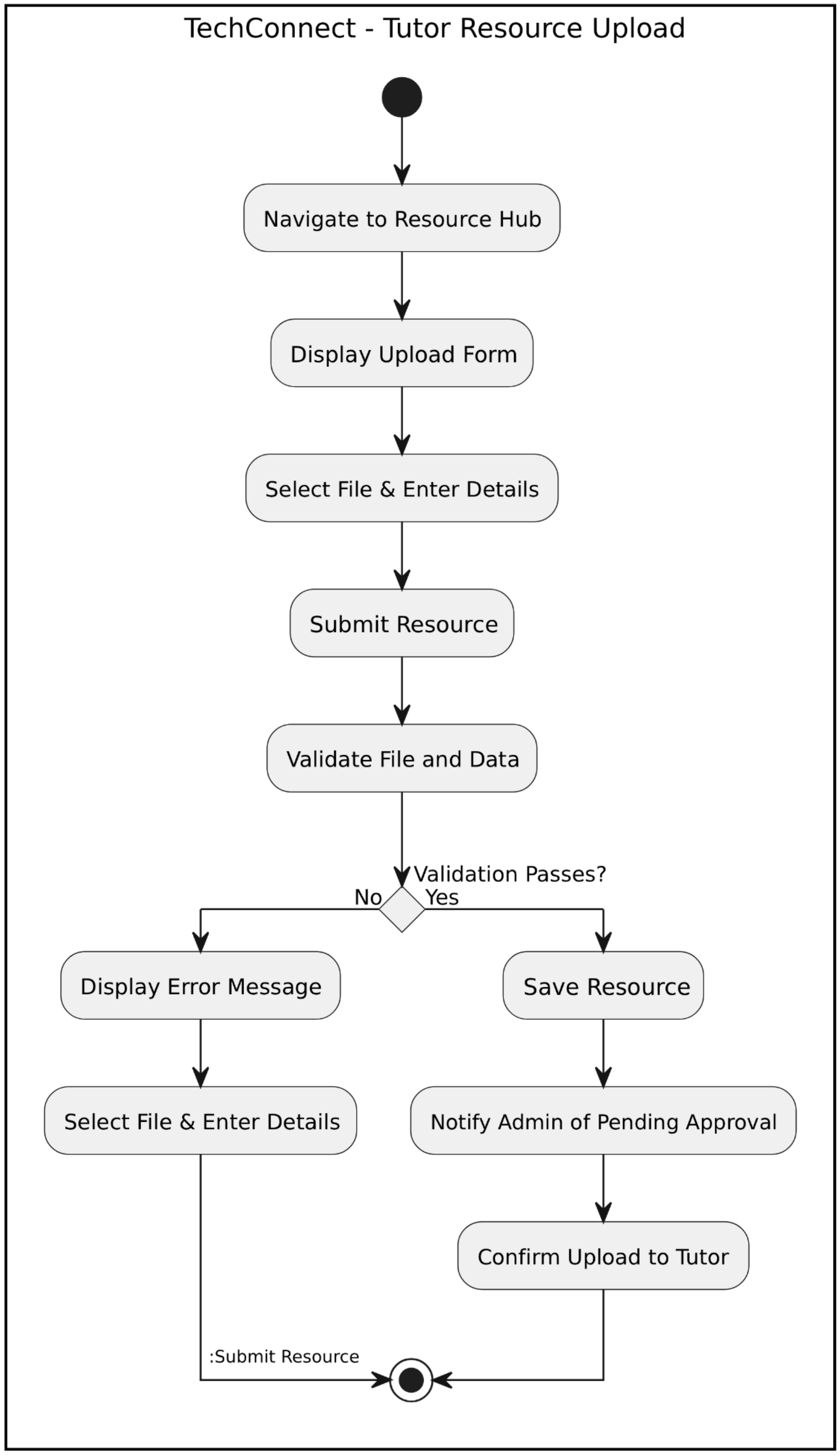

Figure 13. *Tutor Resource Upload*

This activity diagram illustrates the process of a tutor adding a new learning resource, such as notes, slides, or videos, to the Resource Hub. The flow includes several key steps: uploading the file, entering relevant metadata, system validation, and final confirmation.

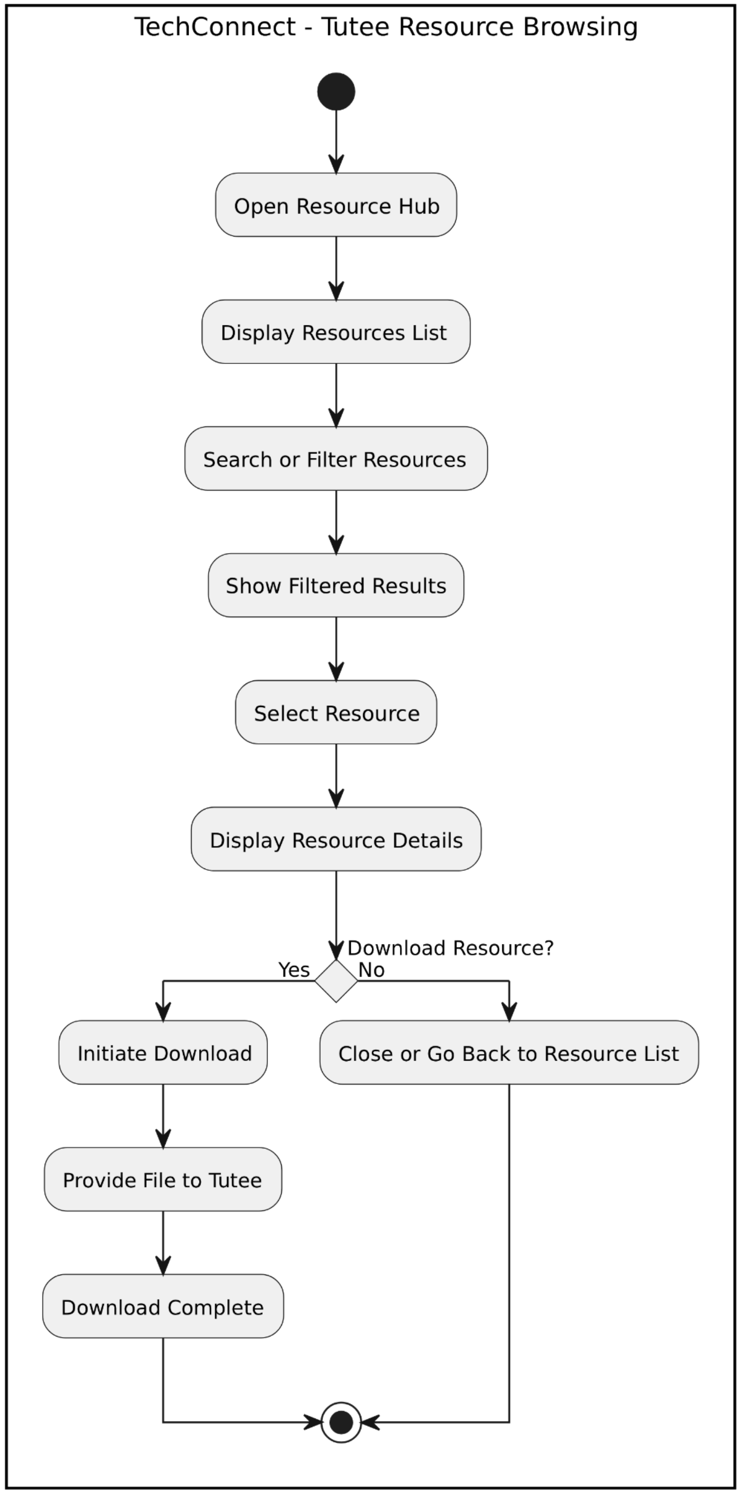

Figure 14. *Tutee Resource Browsing*

This diagram shows how a tutee browses and accesses resources in the Resource Hub. It includes viewing the list of resources, filtering or searching, selecting a resource, and downloading or viewing it.

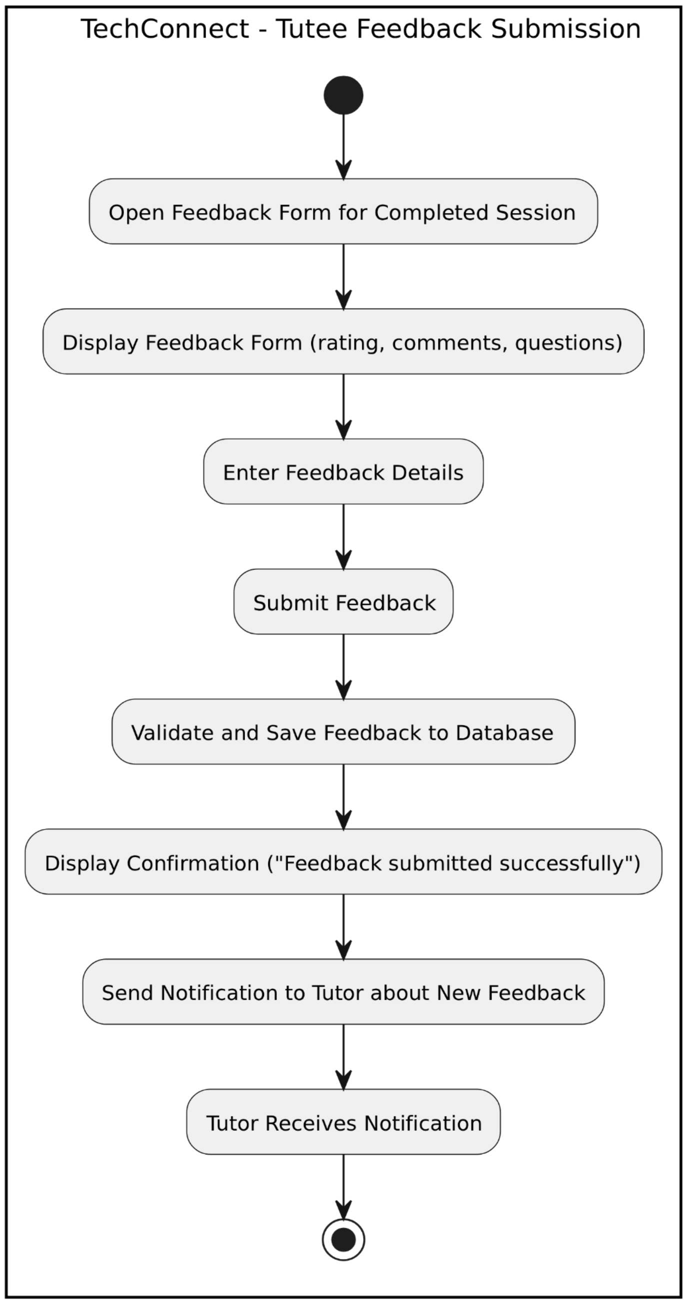

Figure 15. *Tutee Feedback Submission*

After a tutoring session, the tutee can provide feedback or a rating for the tutor. This diagram shows the feedback submission process: the tutee filling out a feedback form, the system saving the feedback, and notifying the tutor about the new feedback.

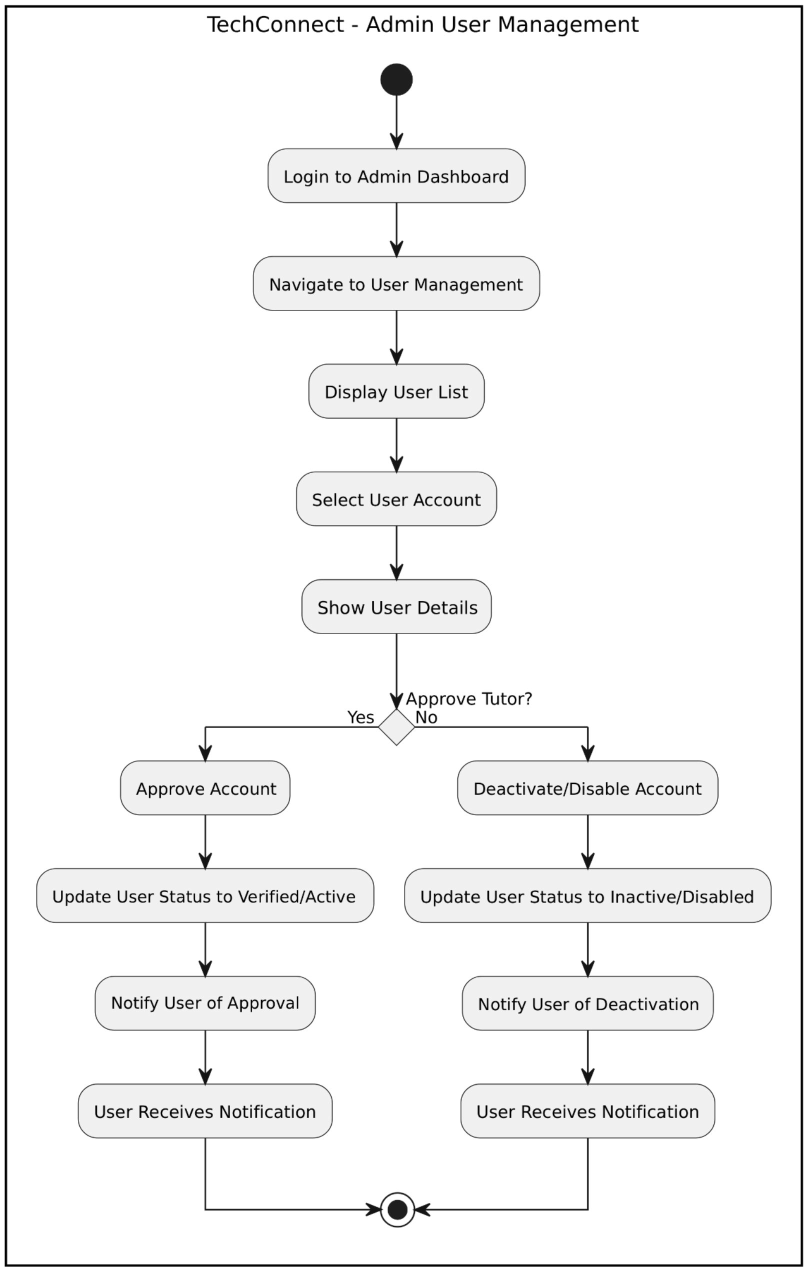

Figure 16. *Admin User Management*

This activity diagram illustrates how an admin manages user accounts on the platform. It includes viewing the user list, selecting a user (tutor or tutee), and performing administrative actions such as verifying a tutor's account or deactivating a user.

Figure 17. *Admin Session Oversight*

This diagram shows how an admin monitors tutoring sessions on the platform. The admin can view all scheduled sessions and, if necessary, intervene – for example, by canceling a session (in case of policy violation or by user request via admin).

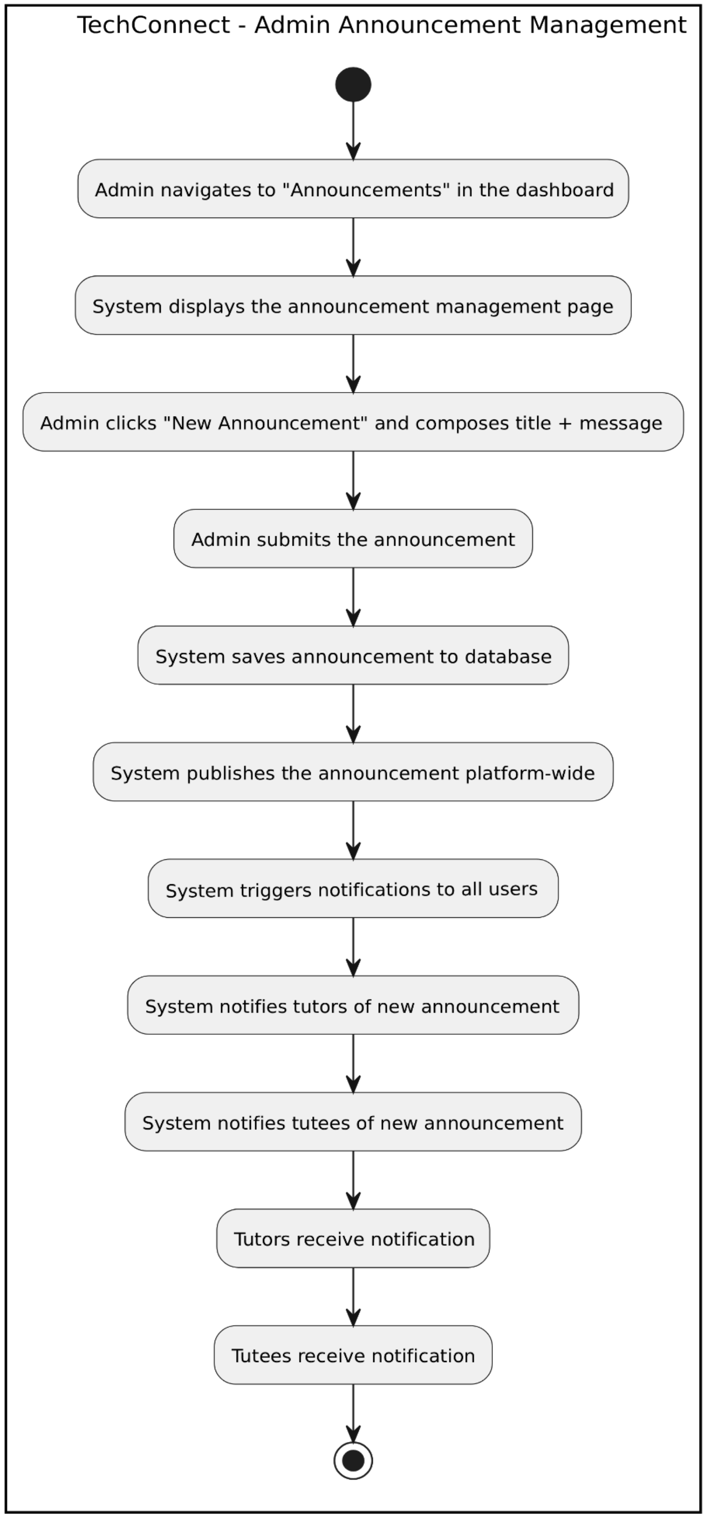

Figure 18. *Admin Announcement Management*

This activity diagram describes how an admin creates and publishes an announcement to all users (tutors and tutees) on the platform. The process includes drafting the announcement, posting it, and the system distributing it so that every user can see it (usually via notifications or a dashboard update).

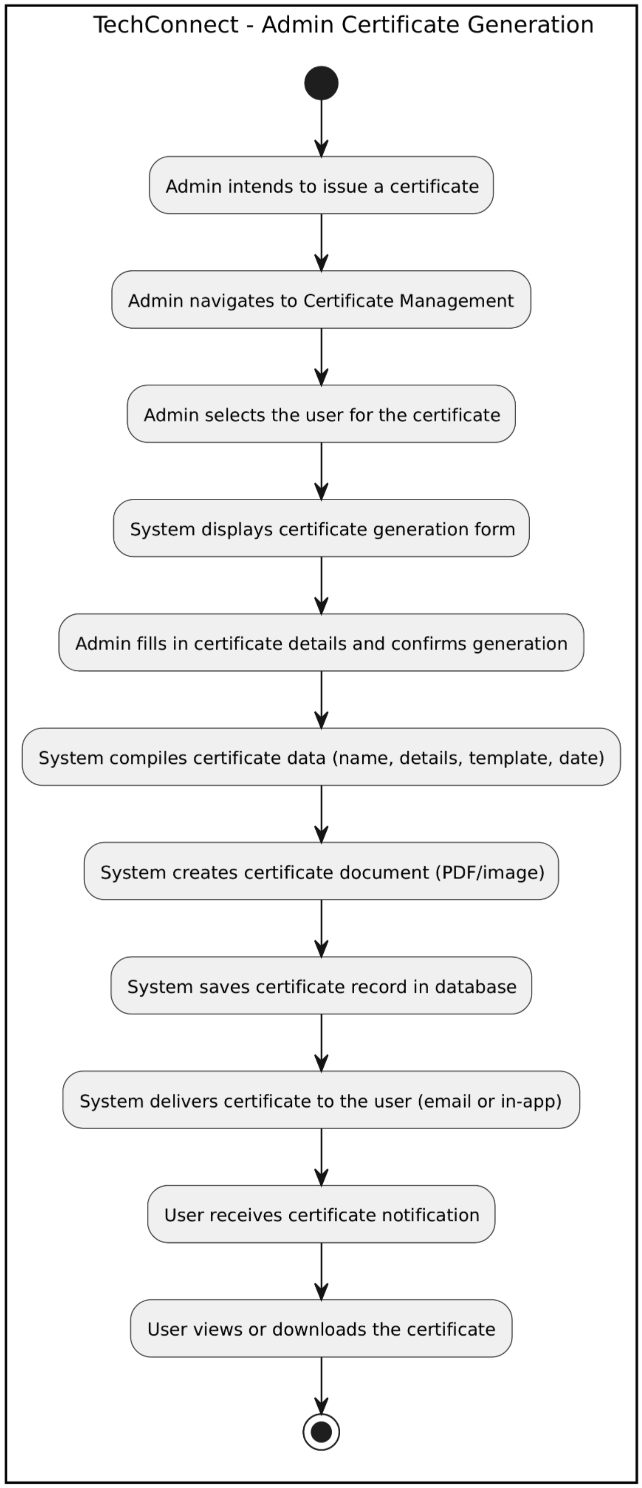

Figure 19. *Admin Certificate Generation*

This diagram outlines how an admin generates a certificate for a user (for example, a certificate of completion or recognition after certain achievements). It covers the admin selecting the user, the system creating the certificate, and delivering it to the user.

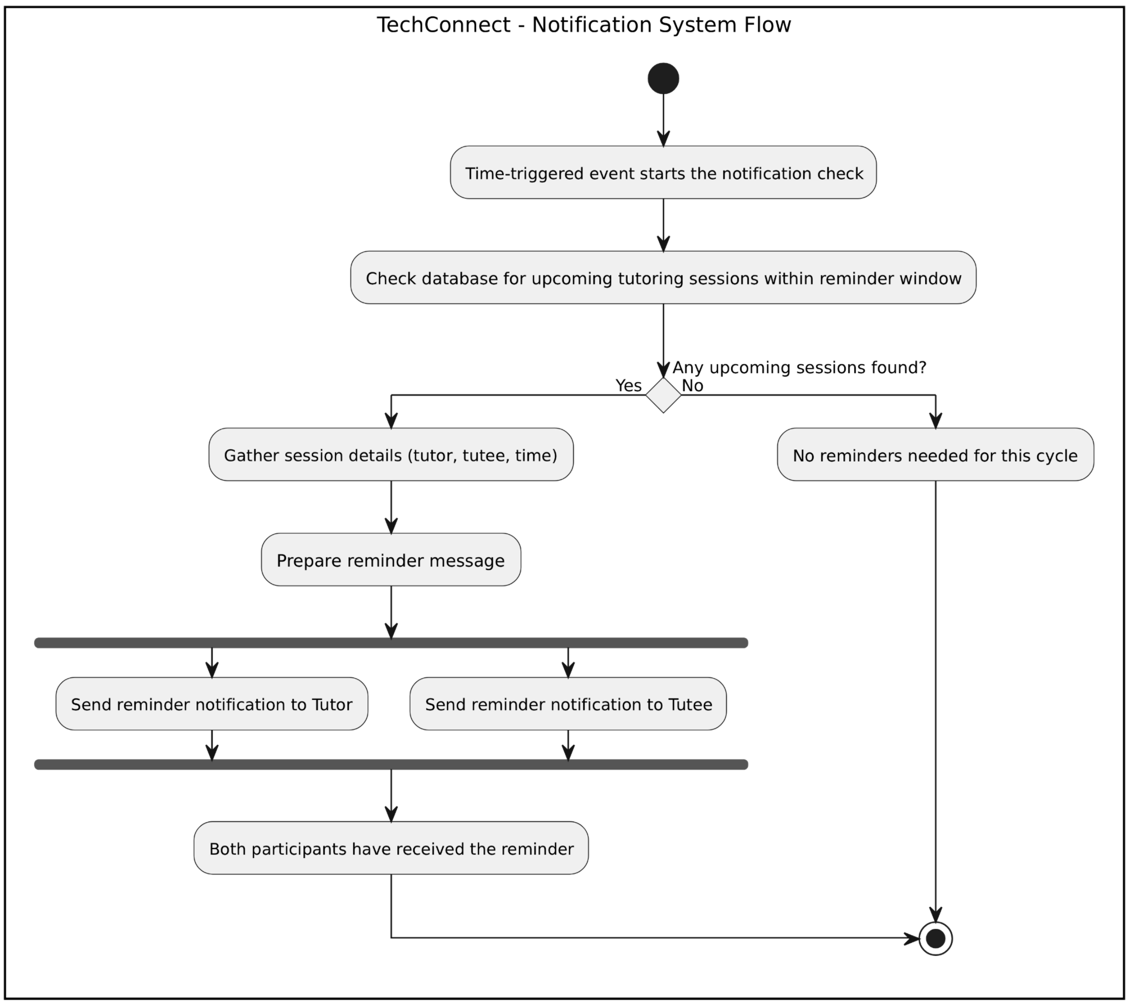

Figure 20. *Notification System*

This activity diagram represents the automated notification system that keeps users informed of important events. As an example scenario, it illustrates a scheduled session reminder that the system sends to both the tutor and tutee before a session begins.

Sequence Diagram

User Registration & Authentication

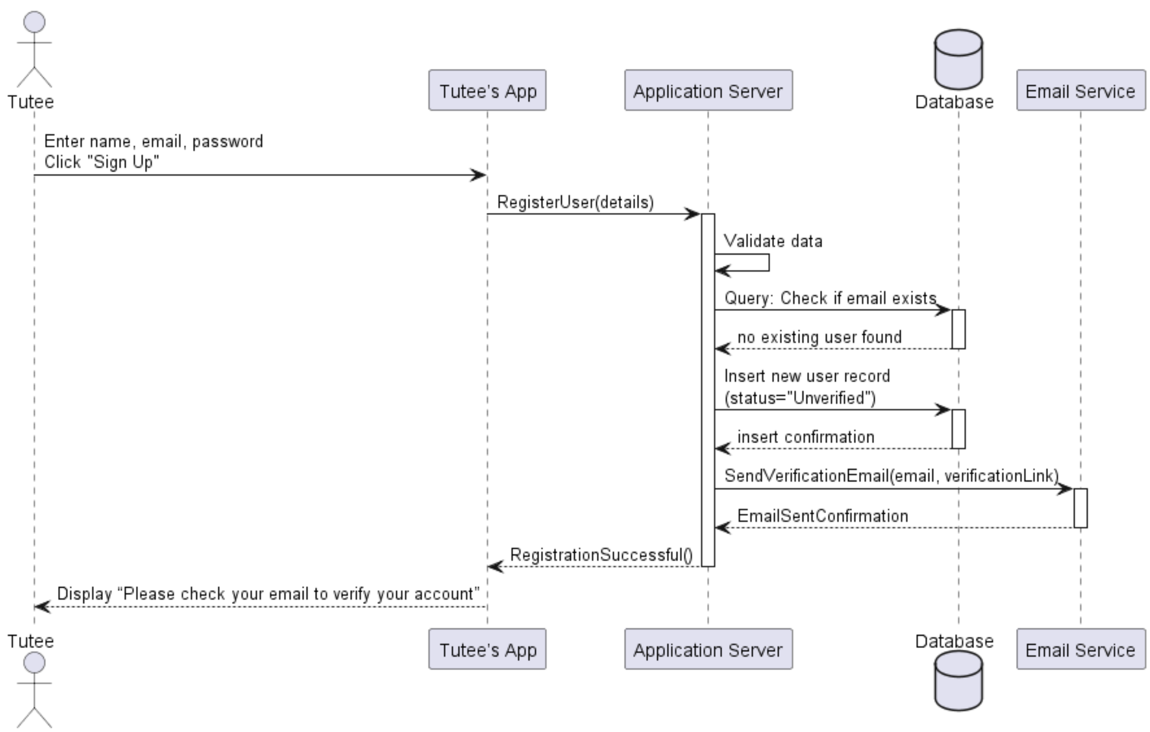

Figure 21. *Tutee Registers an Account*

This diagram shows the registration process for a new Tutee on the TechConnect platform. It outlines how the tutee's app interacts with the server to create an account and confirm the registration via email verification.

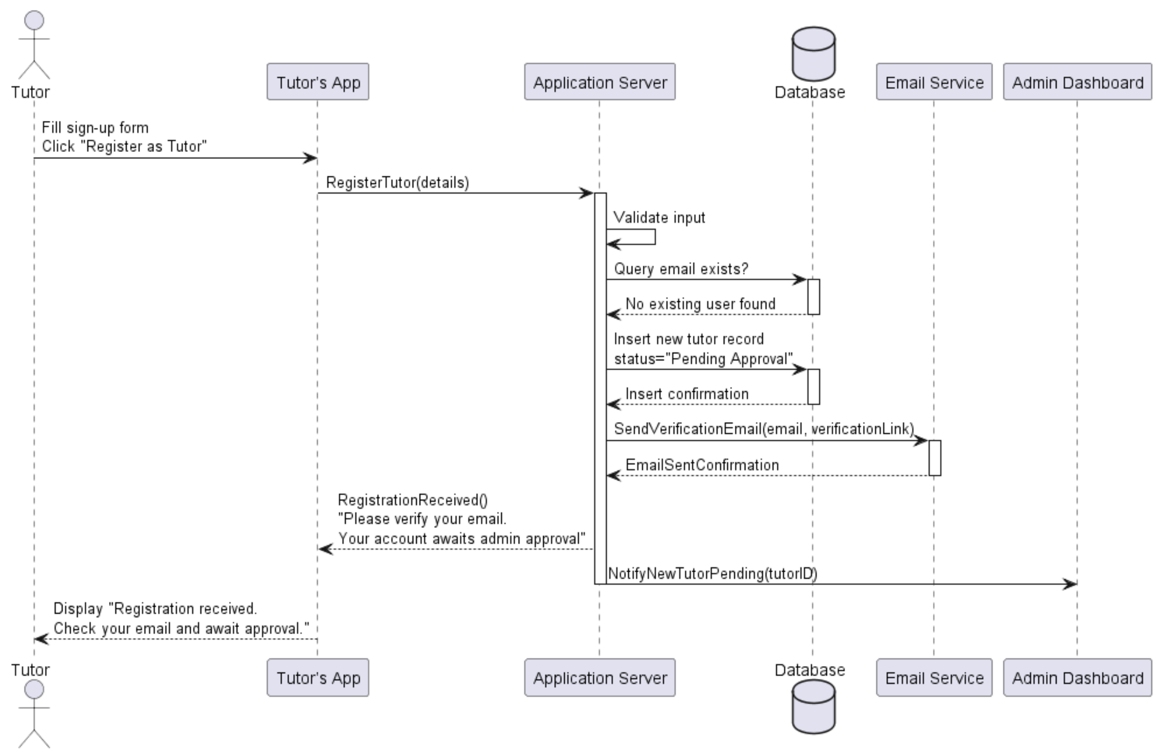

Figure 22. *Tutor Registers an Account*

This diagram details how a new Tutor registers on the platform. The process is similar to tutee registration but includes flagging the account for administrative approval after email verification, meaning the tutor cannot fully use tutor features until an admin activates their account.

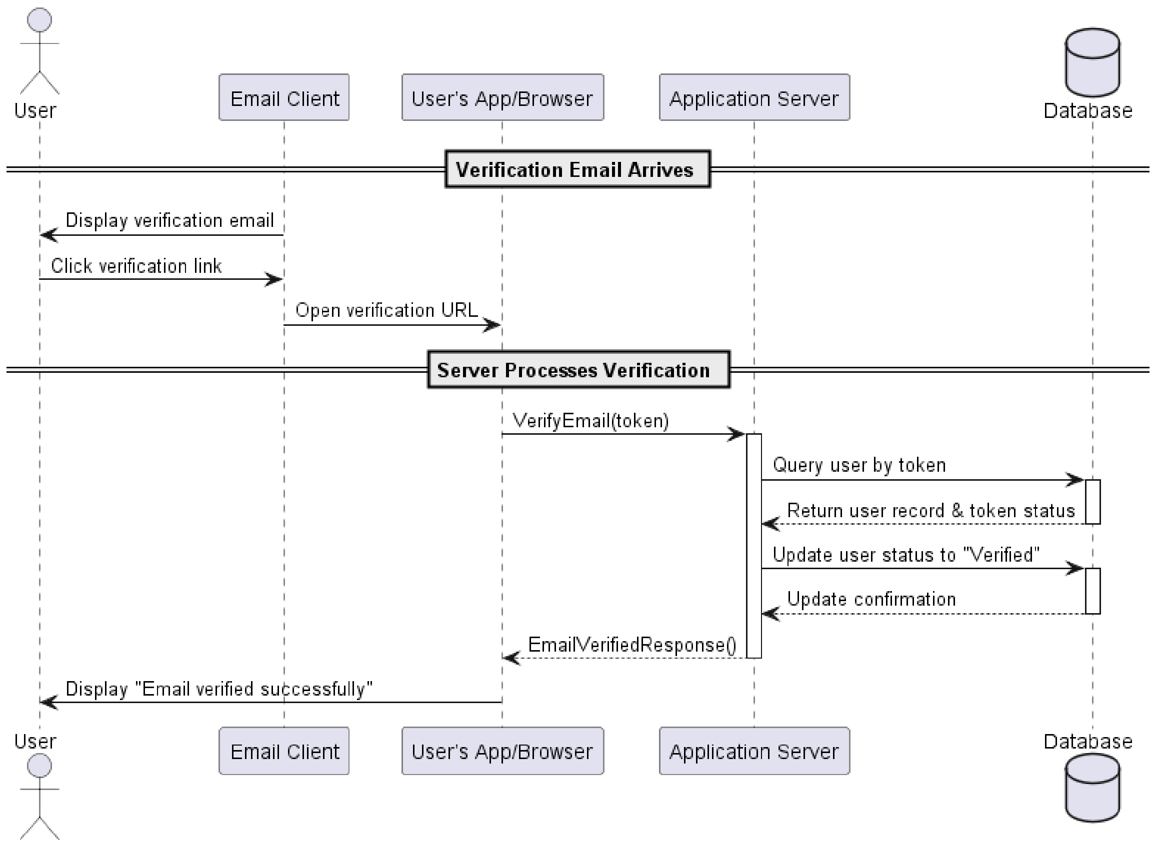

Figure 23. *User Email Verification*

This sequence diagram illustrates how a newly registered user (tutee or tutor) verifies their email to activate their account. It covers the steps from the user receiving the verification email to having their account marked as verified in the system.

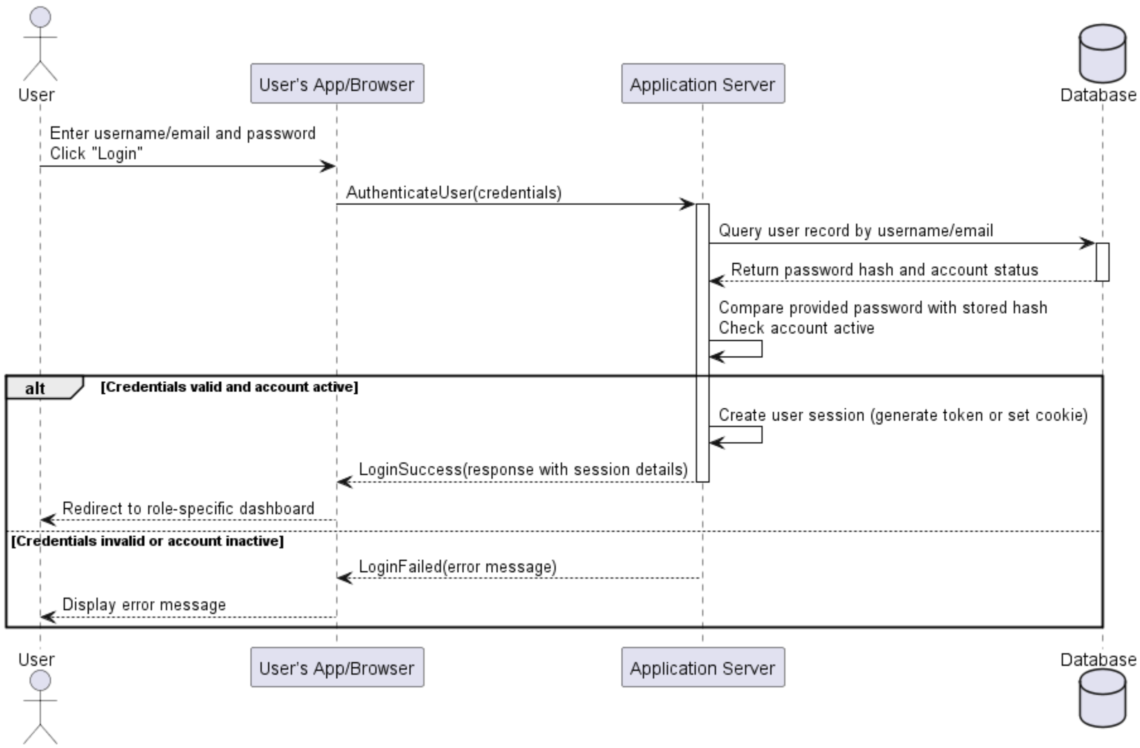

Figure 24. *User Login to Platform*

This diagram depicts the login process for any user (tutee, tutor, or admin) entering their credentials to access the TechConnect platform. It shows the front-end and back-end interaction for authentication.

Tutor Profile Management

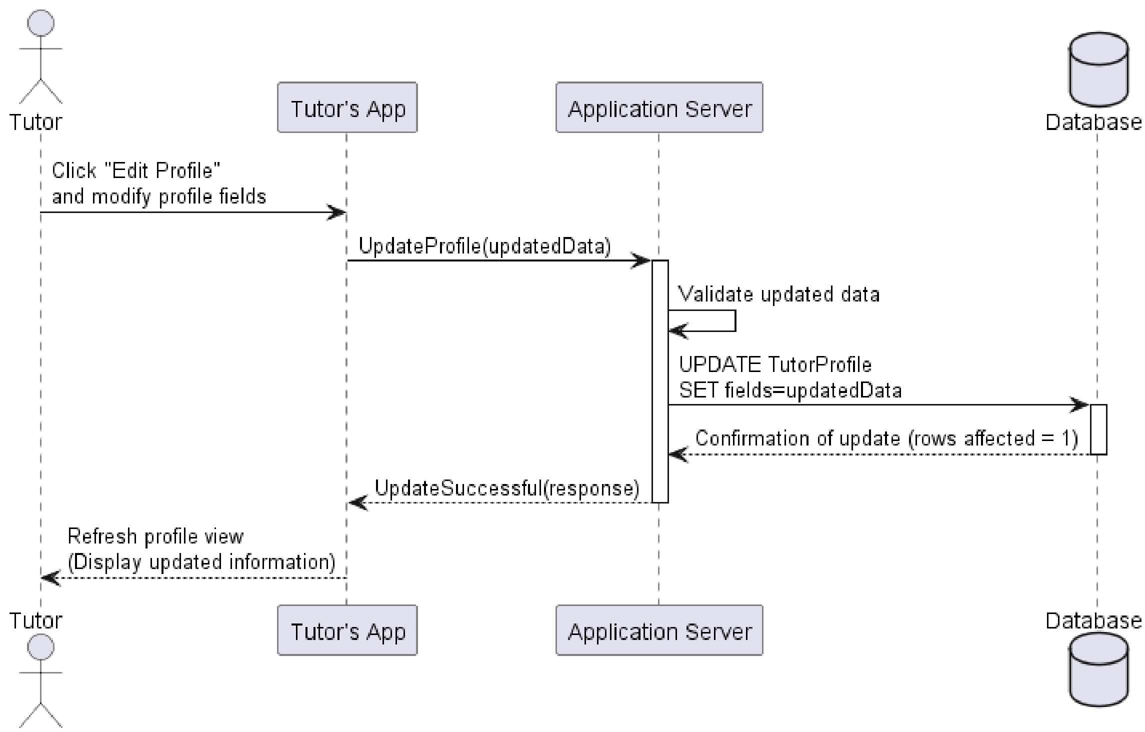

Figure 25. *Tutor Updates Profile Information*

This sequence covers how a Tutor updates their profile details (such as biography, subjects they can tutor, qualifications, etc.). It shows the interaction between the front-end where the tutor enters new info and the back-end that saves these changes.

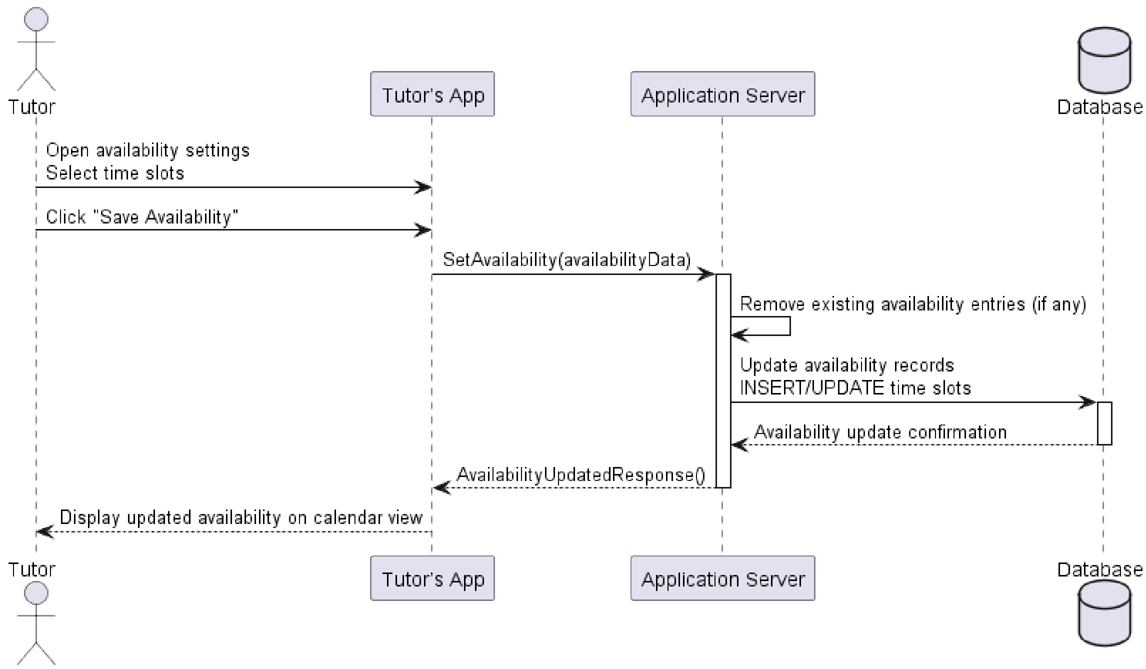

Figure 26. *Tutor Sets Availability Schedule*

This diagram illustrates how a Tutor specifies or updates their availability (the times they are free to conduct tutoring sessions). The tutor interacts with the UI to set time slots, which the server records in the database for use in session matching.

Session Matching & Booking

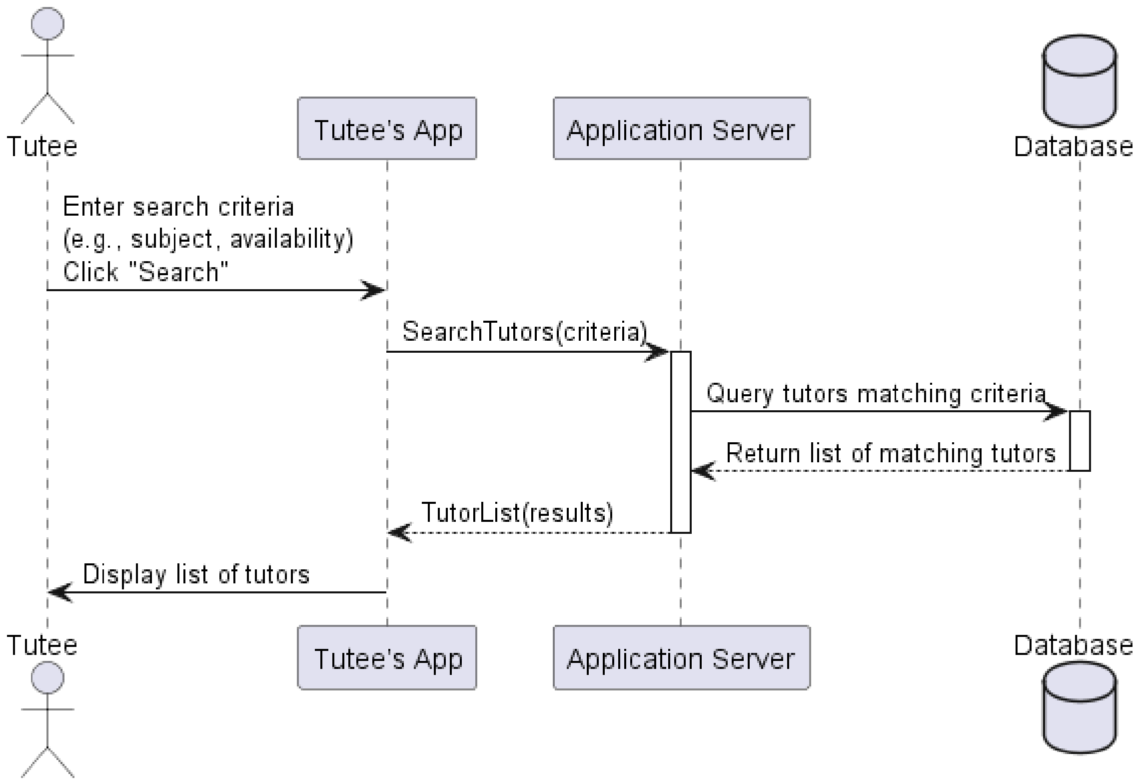

Figure 27. *Tutee Searches for Available Tutors*

This sequence diagram shows how a Tutee searches or browses for tutors on the platform, possibly filtering by subject or other criteria. The front-end requests a list of tutors from the server, which queries the database and returns matching tutors that are available.

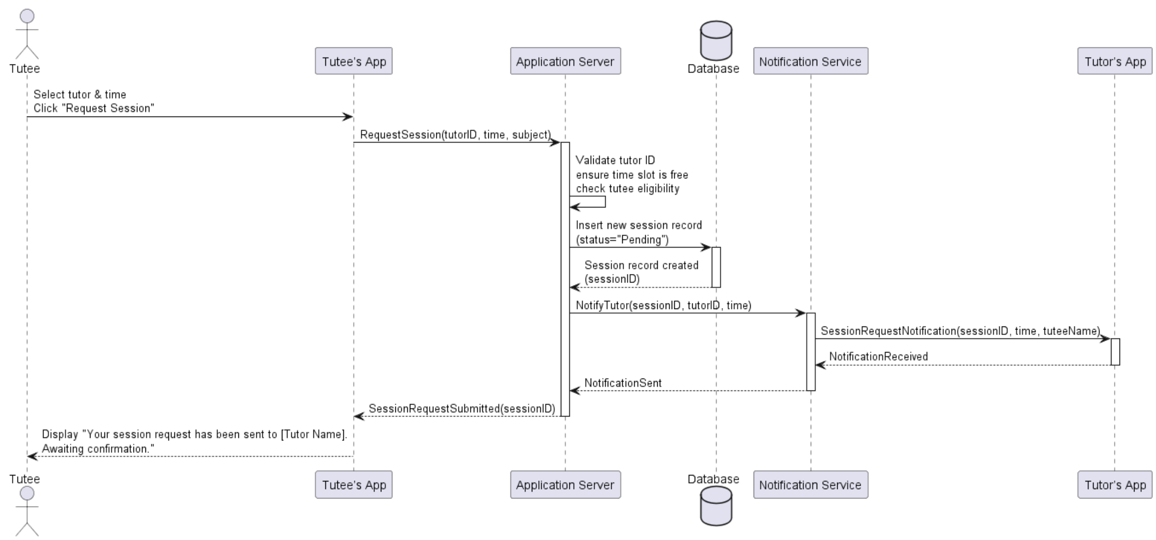

Figure 28. *Tutee Requests a Tutoring Session*

In this diagram, the Tutee selects a tutor and requests a tutoring session at a specific time. The sequence covers sending the session request to the server and how the server records this request and notifies the chosen tutor.

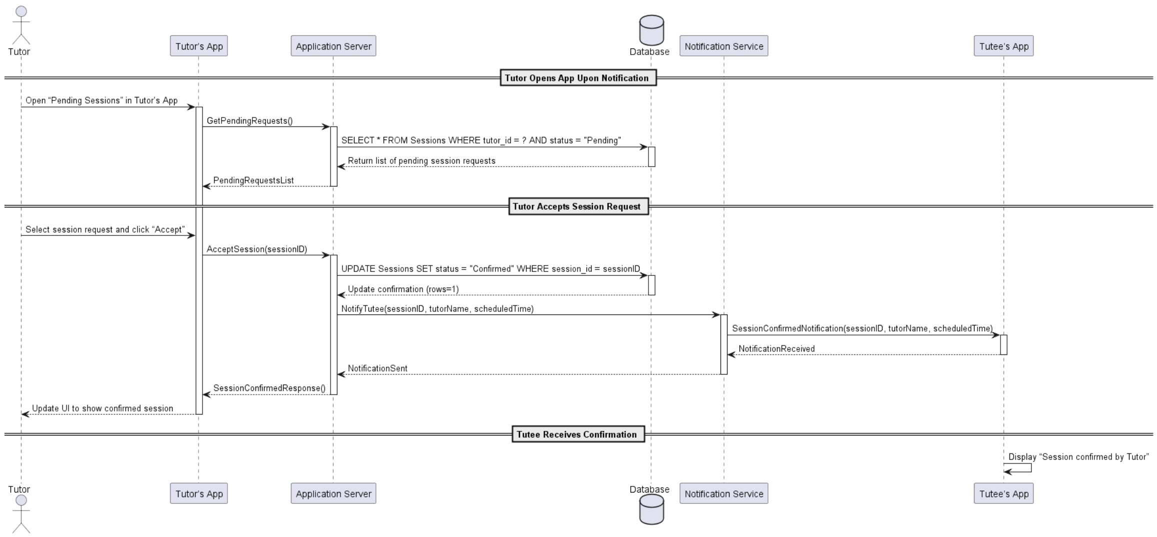

Figure 29. *Tutor Accepts a Session Request*

This sequence diagram describes how a Tutor responds to a pending session request by accepting it. Once the tutor accepts, the system confirms the booking, updates the session status, and notifies the tutee that the session is confirmed.

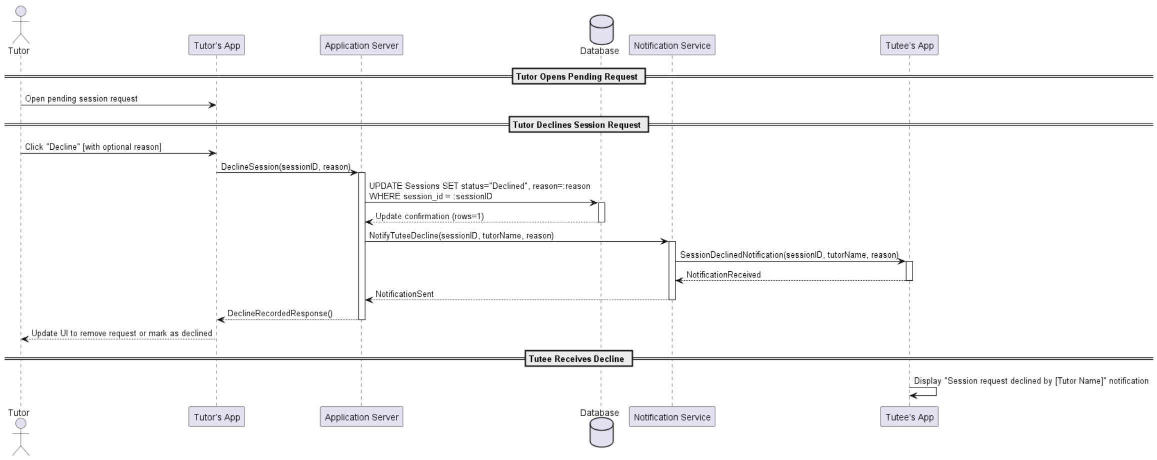

__Figure 30.__ Tutor Declines a Session Request

This sequence shows what happens when a Tutor declines a tutee's session request. The system updates the session status to declined and notifies the tutee, who may choose to request a different tutor or time.

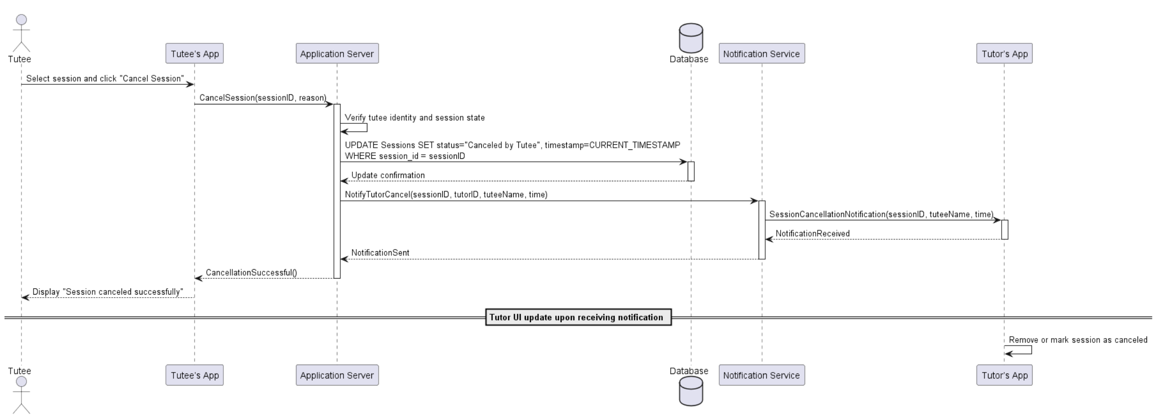

__Figure 31.__ Tutee Cancels a Scheduled Session

This diagram outlines the process when a Tutee decides to cancel a session that had been previously confirmed. The tutee's cancellation triggers updates in the system and notifies the tutor of the cancellation.

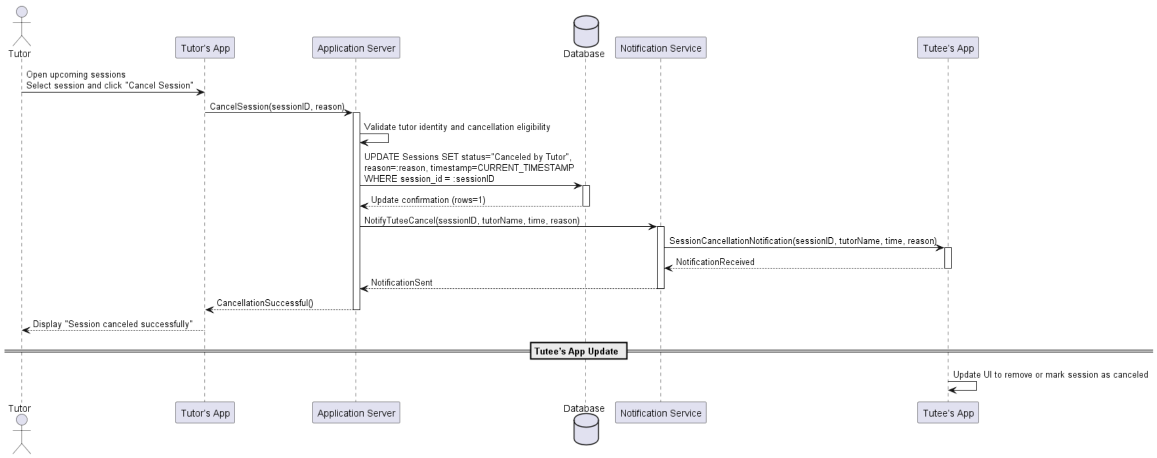

Figure 32. *Tutor Cancels a Scheduled Session*

This sequence diagram shows the scenario where a Tutor needs to cancel an upcoming session that was confirmed. The tutor initiates the cancellation, the system updates the session status, and the tutee is notified of the cancellation.

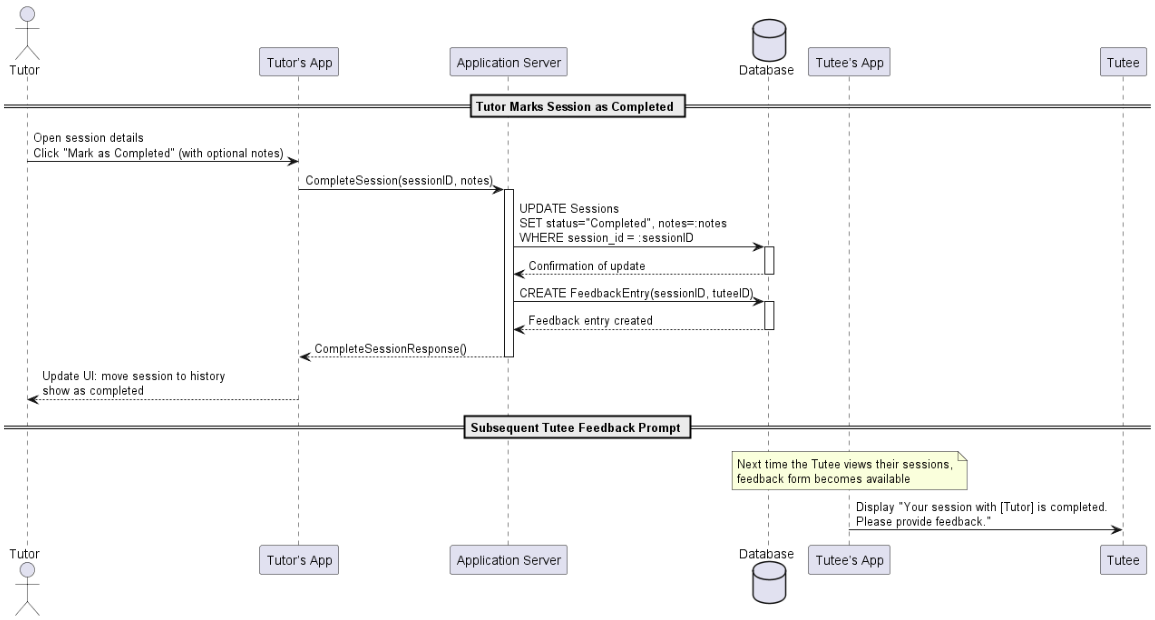

Figure 33. *Tutor Marks Session as Completed*

This diagram describes the process after a tutoring session has taken place. The Tutor marks the session as completed in the system, which updates the session status and triggers the availability of a feedback form for the tutee (and possibly for the tutor to report any notes).

Feedback & Rating Submission

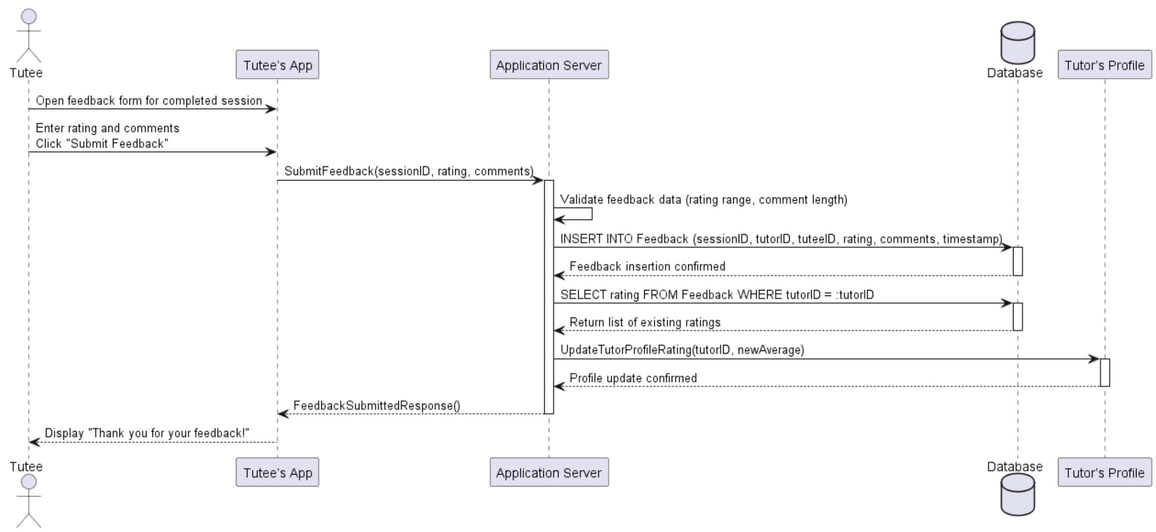

Figure 34. *Tutee Submits Feedback and Rating for Tutor*

This sequence diagram illustrates how a Tutee provides feedback and a star rating for a tutor after a completed session. The process involves the tutee filling out a feedback form, the system saving this feedback, and updating the tutor's overall rating.

Learning Resource Upload & Download

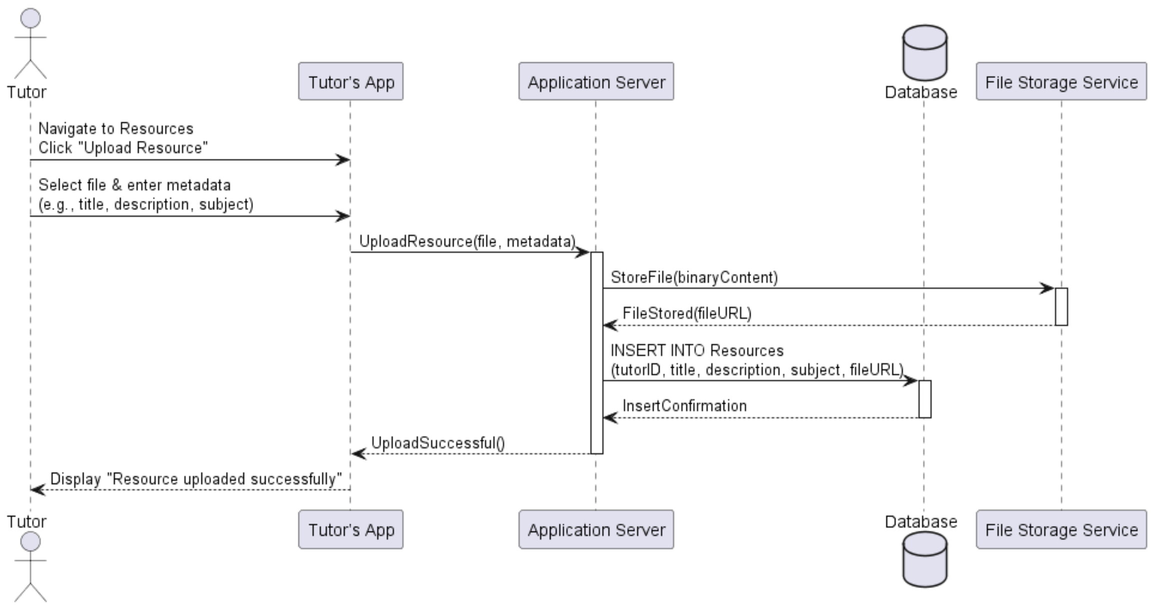

Figure 35. *Tutor Uploads a Learning Resource*

Tutors can upload learning materials or resources (documents, links, etc.) to the platform, either to share with tutees or to include in their profile/course materials. This diagram shows the process of a tutor uploading a file and how the system stores it.

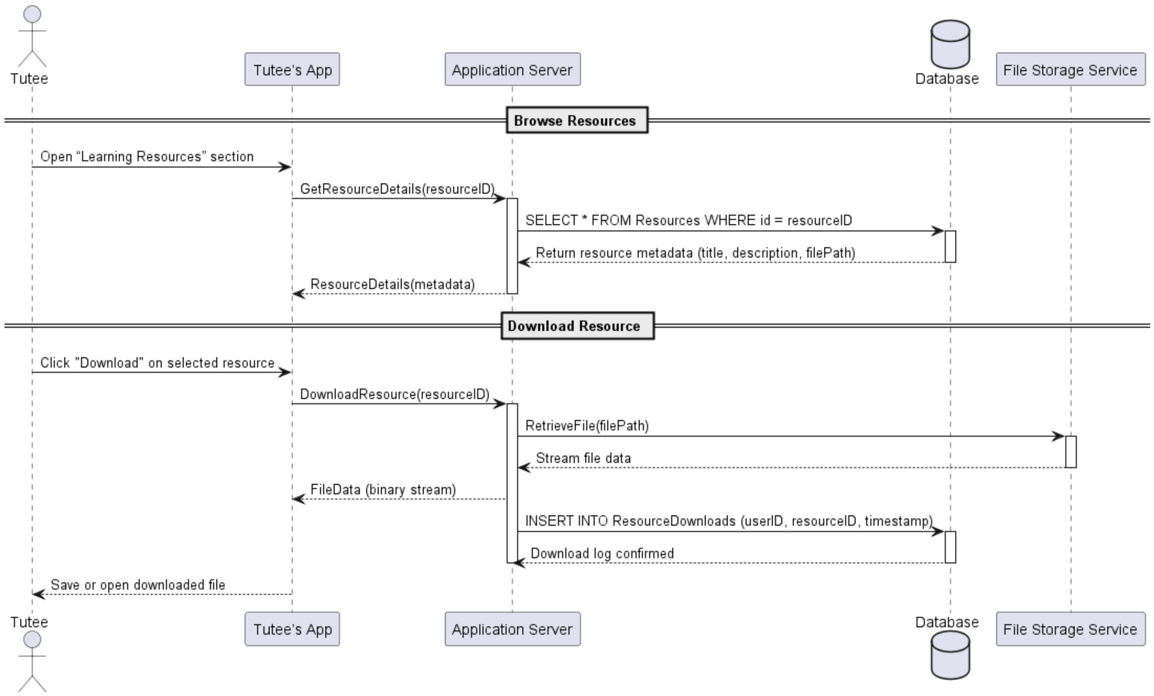

Figure 36. *Tutee Browses and Downloads a Resource*

This sequence diagram illustrates how a Tutee finds and downloads a learning resource from the platform. The tutee searches or selects a resource, and the system retrieves the file for the tutee to access.

Session Management by Tutors

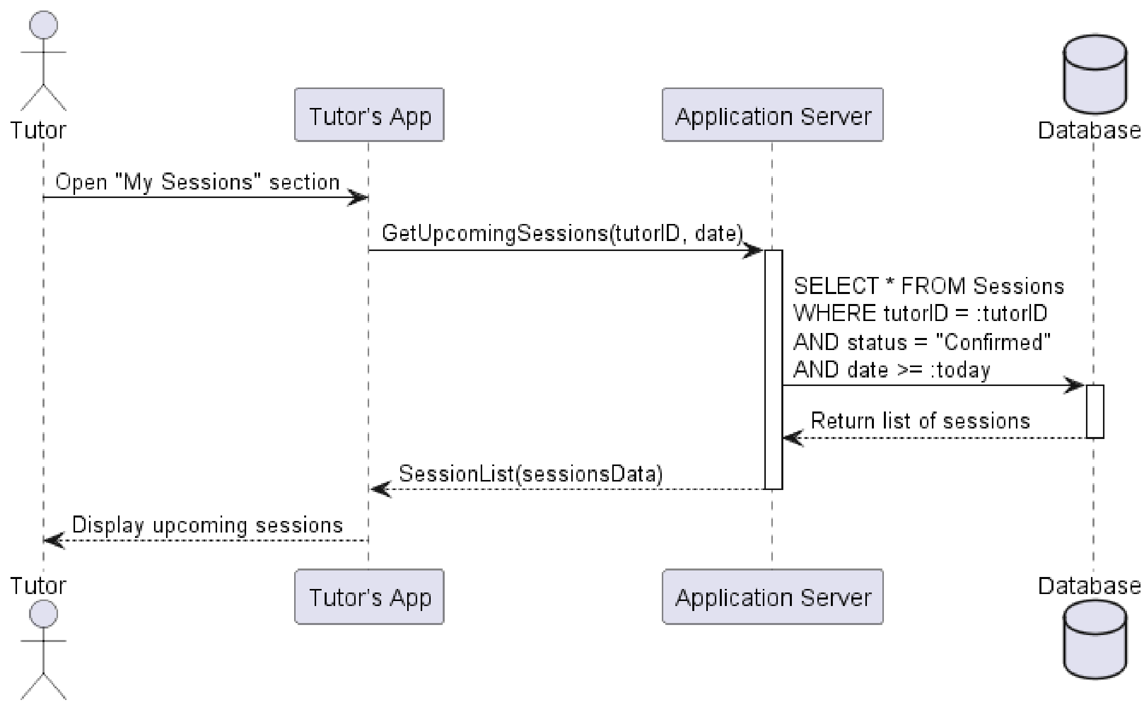

Figure 37. *Tutor Views Scheduled Sessions*

This diagram shows how a Tutor can view their schedule of upcoming tutoring sessions. The tutor's app requests a list of sessions, and the server provides the data from the database.

Administrative Oversight

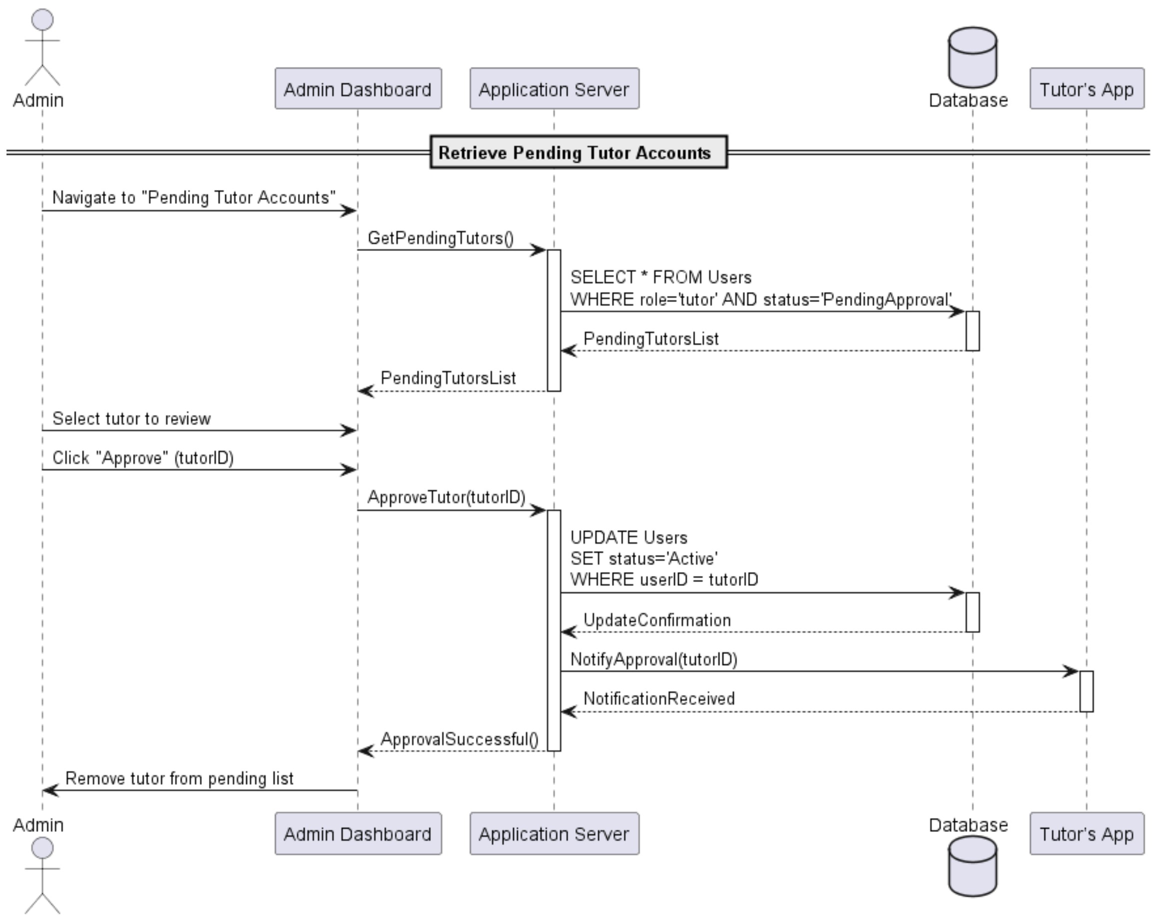

Figure 38. *Admin Approves Tutor Account*

After a tutor registers (from the earlier Tutor Registers sequence) and verifies their email, an administrator needs to review and approve the tutor's account. This diagram shows how an admin user reviews pending tutor accounts and approves a tutor, activating their account for tutoring.

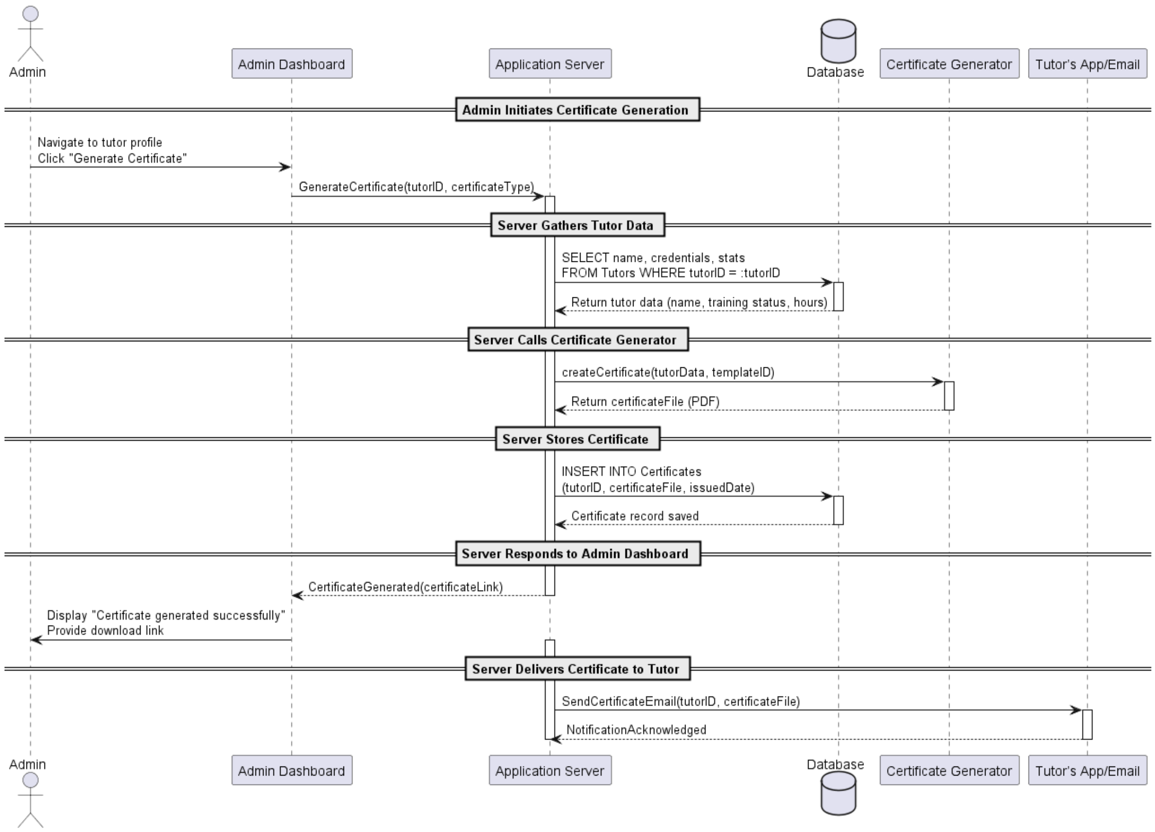

Figure 39. *Admin Generates Tutor Certificate*

This sequence diagram demonstrates how an Admin generates a certificate for a tutor, for example, to certify the tutor's participation or performance in the peer tutoring program. The admin initiates the certificate generation, and the system produces a certificate (likely a PDF) and makes it available for download or emailing.

Notification Triggers & Updates

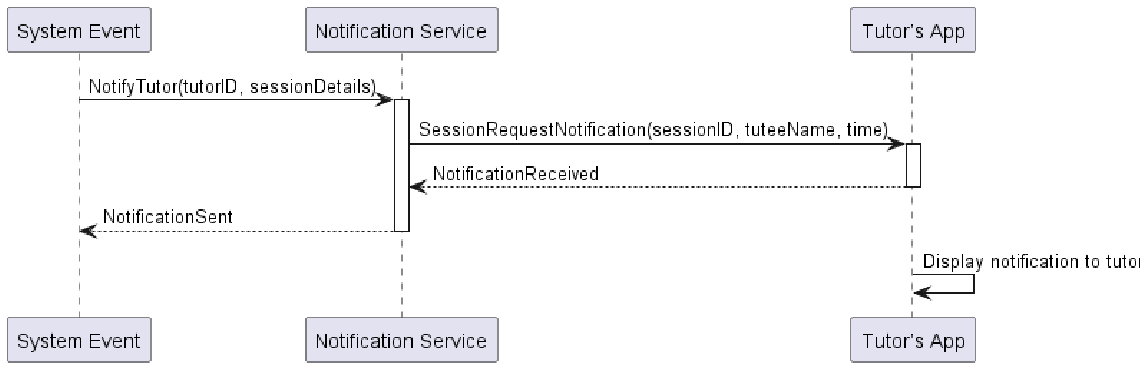

Figure 40. *Session Request Notification to Tutor*

This sequence diagram highlights how the system sends a notification to a tutor when a tutee requests a session. It is an elaboration of the notification step within the session request flow, showing the interaction with the notification subsystem.

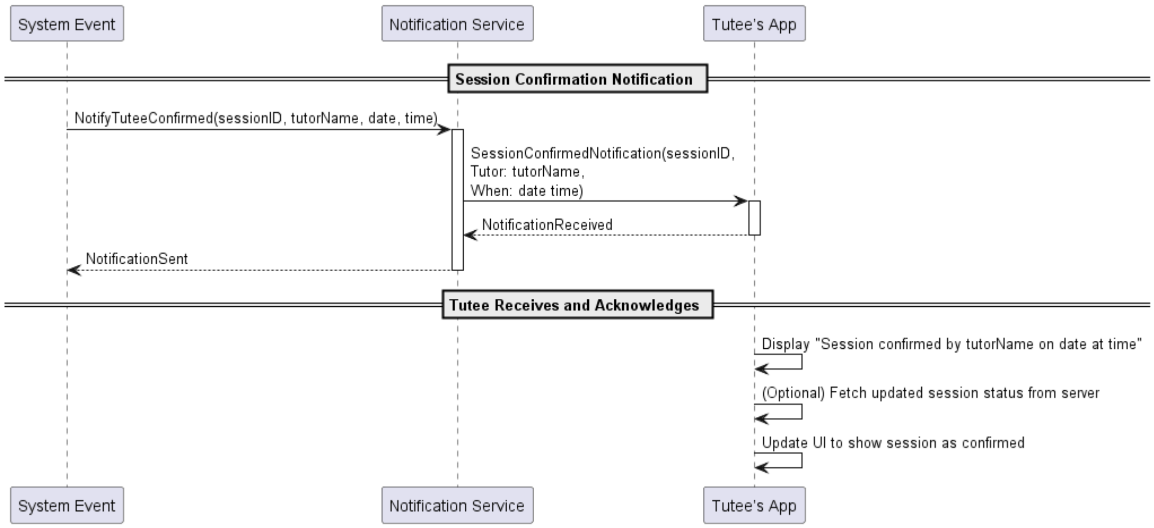

Figure 41. *Session Confirmation Notification to Tutee*

This diagram focuses on the notification sent to a tutee when a tutor has accepted their session request (i.e., when a session is confirmed). It details how the system generates and sends this notification to update the tutee.

Class Diagram

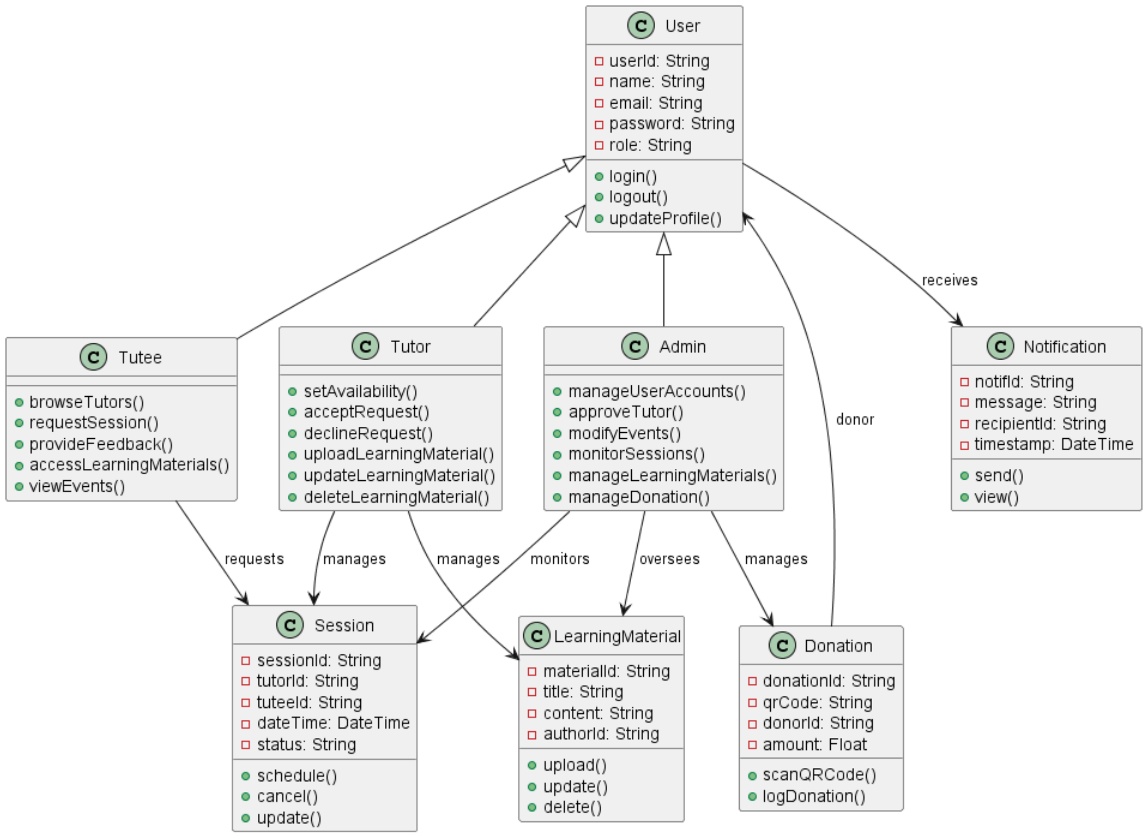

Figure 42. *Class Diagram*

The figure illustrates the class diagram of the TechConnect peer tutoring platform, presenting its classes, attributes, and methods, along with the relationships that define how users and core system components interact. At the core of the system is the User class, which serves as the foundation for all participants. It contains personal and account-related attributes such as userId, name, email, password, and role, and provides essential methods including login(), logout(), and updateProfile(). This class is extended by three specialized subclasses: Tutee, Tutor, and Administrator, each of which inherits basic user functions while introducing role-specific methods.

The Tutee class represents learners who seek academic support. It provides methods for browsing available tutors, requesting tutoring sessions, providing feedback, accessing learning materials, and viewing events. These functions support the tutee's journey from session discovery to active participation in peer-assisted learning.

The Tutor class represents peer mentors who deliver academic assistance. Its methods include setting availability, accepting or declining tutoring requests, and managing supplementary learning materials through upload, update, and deletion operations. Tutors also receive notifications and feedback, which help them refine their mentoring effectiveness over time.

The Administrator class oversees the platform and maintains operational integrity. It includes methods for managing user accounts, approving and verifying tutor profiles, modifying events, monitoring sessions, managing official learning materials, and administering the donation module. These functions ensure that the system remains secure, reliable, and academically credible.

Supporting these user roles are several core system classes. The Session class manages tutoring engagements between tutees and tutors, containing attributes such as sessionId, tutorId, tuteeId, dateTime, and status. Its methods schedule(), cancel(), and update() enable dynamic and flexible session handling. The LearningMaterial class stores educational resources with attributes such as materialId, title, content, and authorId, and methods for upload(), update(), and delete() that allow tutors to contribute resources and administrators to oversee quality.

The Notification class facilitates in-app communication across the platform. It includes attributes such as notifId, message, recipientId, and timestamp, and methods send() and view() to keep users informed of session updates, approvals, and other relevant events. The Donation class enables financial contributions through QR code–based transactions. It contains attributes such as donationId, qrCode, donorId, and amount, and provides methods for scanQRCode() and logDonation() to record and track donations managed by the administrator.

The relationships between these classes highlight the interconnected structure of the platform. The Tutee requests and participates in Sessions, while Tutors manage Sessions and contribute Learning Materials. Administrators monitor Sessions, oversee Learning Materials, and manage Donations. All user types receive Notifications, ensuring consistent communication throughout the system.

Entity Relationship Diagram

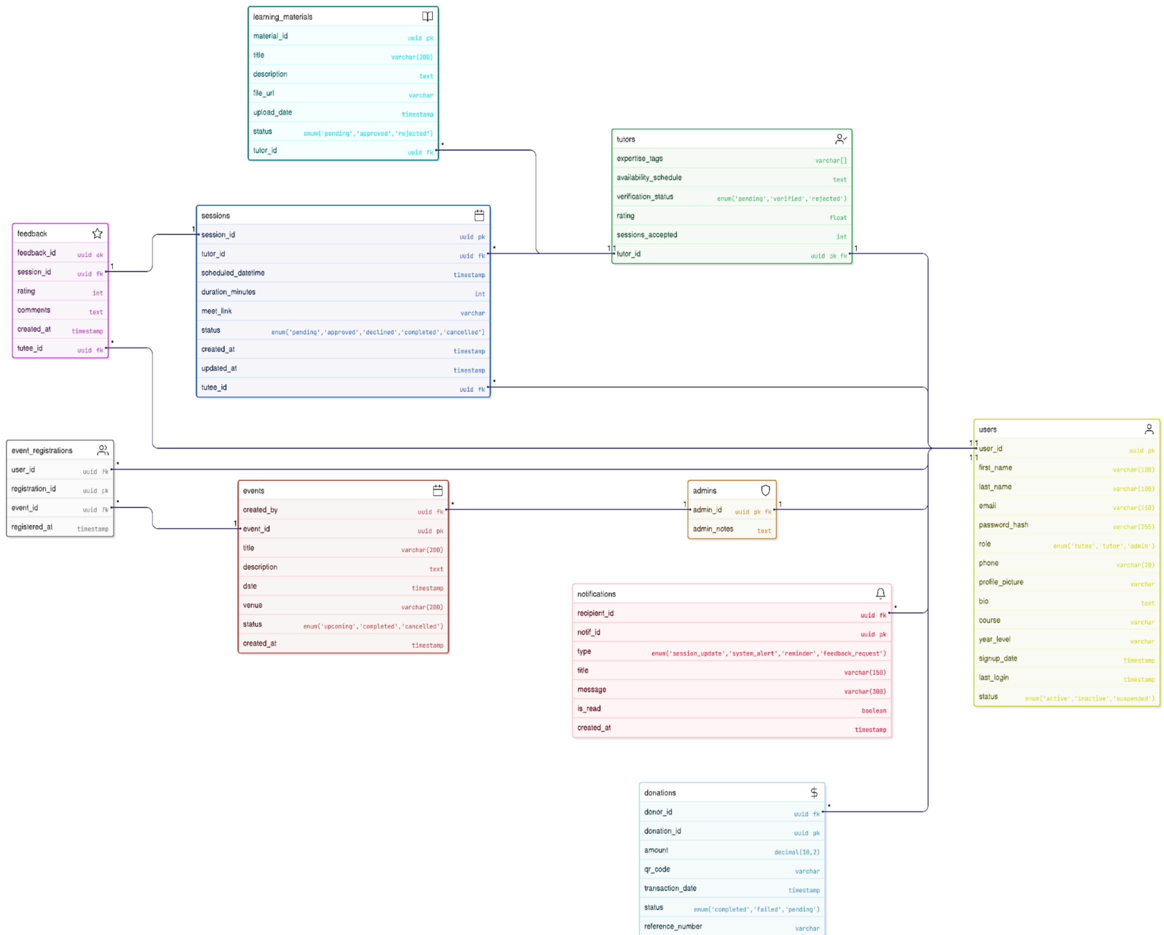

Figure 43. *Entity Relationship Diagram*

The *Users* table serves as the foundation of the TechConnect system, supporting user authentication, profile management, and role-based access. It securely manages user credentials through email–password combinations with encrypted password storage, while the role attribute distinguishes between Tutee, Tutor, and Administrator. The table includes personal information such as full name, contact details, and academic profile data (course, year level, and optional biography), with timestamps for account creation and recent logins. User lifecycle management is facilitated through a status attribute (active, suspended, inactive), ensuring administrators retain control over system participation.

Building on this foundation, the *Tutors* table extends the user entity by adding attributes specific to peer mentors. These include areas of expertise expressed through subject tags, structured availability schedules, verification status to confirm eligibility, and tutor performance indicators such as accumulated ratings and accepted sessions. By contrast, the *Administrators* table contains additional metadata supporting oversight functions, including notes for account handling and internal documentation.

The *Sessions* table represents the central activity of the platform, mapping the interaction between Tutors and Tutees. Each session record contains information such as scheduled date and time, duration, meeting link, and session status (pending, approved, declined, completed, or cancelled). Session updates are timestamped for auditability, and relational integrity is maintained through foreign keys linking both tutor and tutee participants.

The *Feedback* table is directly associated with completed sessions, capturing post-interaction evaluations submitted by tutees. Attributes include rating scores, qualitative comments, and creation timestamps, which support performance monitoring and continuous improvement of tutoring services.

The *LearningMaterials* table supports knowledge sharing within the platform. Each record includes metadata such as title, description, file location, and upload date, along with an approval status managed by administrators to ensure the relevance and quality of supplementary learning resources. Materials are linked to tutors as creators, while administrative oversight provides structured content governance.

Complementing this, the *Notifications* table provides an in-app alerting mechanism that keeps users informed of important system activities. It records notification types (e.g., session updates, reminders, system alerts, feedback requests), titles, messages, read status, and timestamps. Each notification is tied to a specific user and, where appropriate, linked to relevant entities such as sessions or feedback, ensuring contextual communication across the platform.

The *Donations* table introduces a voluntary support module, allowing users to contribute financially through QR code–based transactions. Each donation record includes donor identity, amount, reference number, transaction status, and timestamps, enabling transparent and accountable handling of user contributions without requiring integration of complex external payment gateways.

The *Events* table strengthens community engagement by managing academic events, workshops, and announcements. Event records include details such as title, description, scheduled date, venue, administrative creator, and event status (upcoming, completed, or cancelled). This module enables administrators to create structured learning opportunities beyond one-to-one tutoring sessions, while also supporting broader peer interaction.

The relationships among these entities demonstrate the integrated nature of TechConnect. Users (as tutees or tutors) participate in sessions, tutors contribute learning materials, administrators verify tutors and manage uploads, sessions generate feedback, users receive notifications linked to system events, and donations are attributed to donors. Meanwhile, administrators create events, and users may register as participants. Together, these relationships ensure that TechConnect functions as a cohesive ecosystem for tutoring, resource sharing, event management, and user engagement.

Deployment Diagram

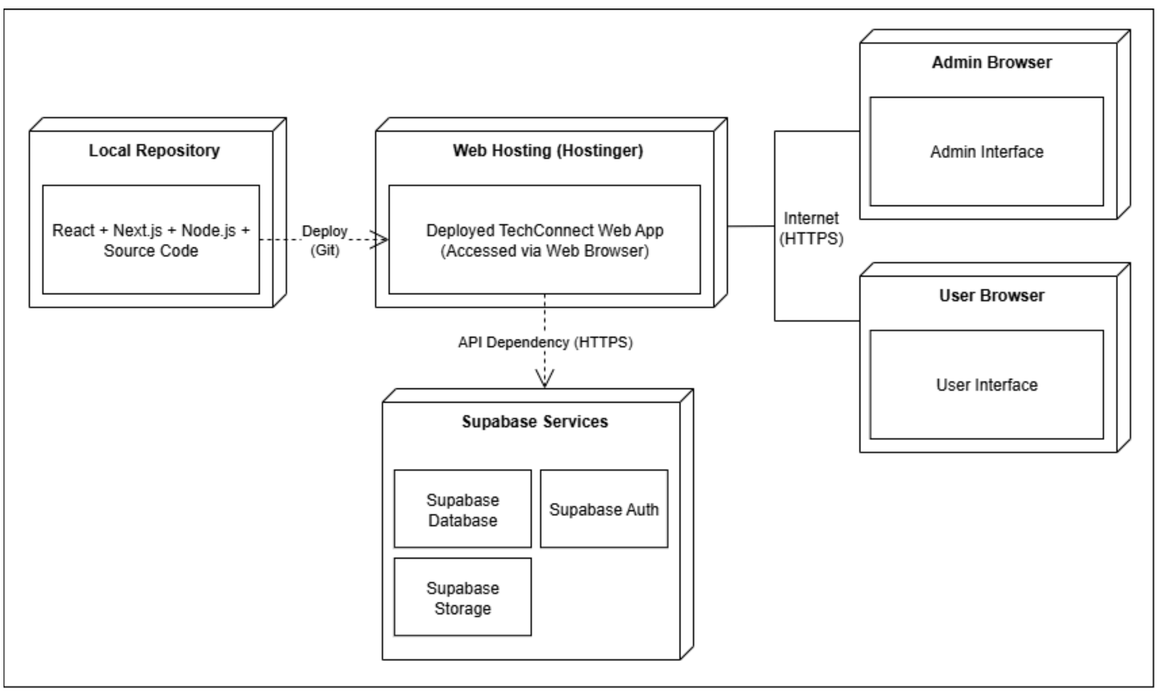

Figure 44. *Deployment Diagram of TechConnect*

This diagram illustrates the high-level deployment architecture of the TechConnect peer tutoring system, showing how each component interacts to deliver a responsive, multi-platform experience. Users access the platform through their devices (web browser), where the React UI (Next.js frontend) runs.

Authentication (sign-in, sign-up) is securely handled by Supabase Auth, ensuring proper credential management before further interactions occur. Once authenticated, users communicate with the Next.js backend, hosted on Hostinger, which manages business logic and API routes. The backend is responsible for server-side rendering (SSR) and exposing API endpoints that allow clients to manage tutoring sessions, update profiles, and submit feedback.

For persistent data storage, the system uses the Supabase Database (Postgres) to store user records, session logs, feedback, and events. Real-time functionality, such as in-app notifications, is powered by Supabase Realtime (WebSocket connections), ensuring instant updates to connected clients.

File uploads and downloads (learning materials, QR codes, certificates) are handled by Supabase Storage, allowing the platform to efficiently manage large binary content. The Donation Module uses this storage service to fetch and display QR images for seamless contributions.

Whenever relevant events occur—such as new tutoring sessions or feedback submissions—the platform supports real-time alerts through in-app notifications. Additionally, the backend integrates with external APIs such as the Google Calendar API and Google Meet API to create and attach meeting links, providing students and tutors with smooth scheduling and collaboration tools.

Together, these components form a scalable, secure, and real-time tutoring platform. The architecture leverages Next.js for frontend and backend integration, Supabase for authentication, database, storage, and real-time updates, and Google APIs for scheduling and meetings. This ensures a reliable and engaging user experience for students and tutors alike.

Development and Testing Procedure

The researchers adopted an Agile–Kanban software methodology to guide the development and testing of *TechConnect: A Peer Tutoring Platform for the College of Industrial Technology*. This iterative approach allowed the team to deliver features in cycles while continuously refining the system based on feedback from actual users. Development began by identifying the core features needed for peer tutoring, such as user authentication, tutor–student matching, session scheduling, file sharing, and in-app notifications. A Kanban board with columns "To Do," "In Progress," and "Completed" was maintained to track progress, prioritize tasks, and ensure that the most critical features were implemented first. The platform was built with a Next.js frontend and backend hosted on Hostinger, while Supabase provided essential backend services including authentication, database management, real-time notifications, and file storage. Integration with Google Calendar and Google Meet APIs was implemented to support automated scheduling and seamless online meeting creation. Throughout the development process, the researchers collaborated with students and faculty from the

College of Industrial Technology, conducting regular testing sessions after each feature milestone. These sessions allowed users to try new functions and provide immediate feedback, which helped the team quickly address usability issues, refine workflows, and align the platform with real student needs. Testing focused on evaluating the platform's functional suitability, performance efficiency, compatibility with external APIs, reliability, security, maintainability, flexibility, and safety. Special emphasis was placed on safeguarding student data using Supabase Auth and encrypted HTTPS communication, ensuring real-time responsiveness through Supabase Realtime, and validating stable API endpoints via Next.js. By following this Agile–Kanban process, the platform was continuously improved and adapted based on real user experiences, resulting in a secure, scalable, and user-friendly tutoring system that supports the scheduling, communication, and collaborative needs of students in the College of Industrial Technology.

Description of Prototype

This section presents a detailed description of the TechConnect prototype, organized according to each page of the platform.

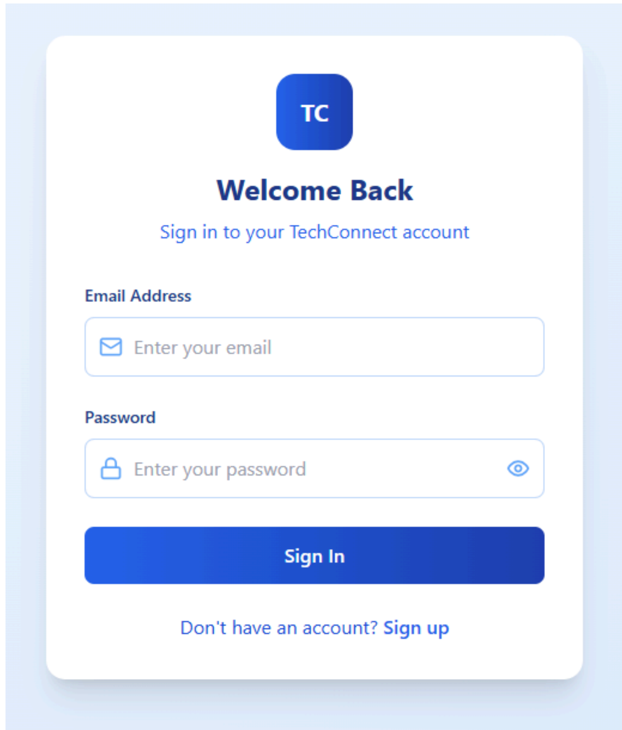

Figure 45. *Sign-In Screen*

This figure demonstrates the sign-in interface where users enter their university email and password to access TechConnect.

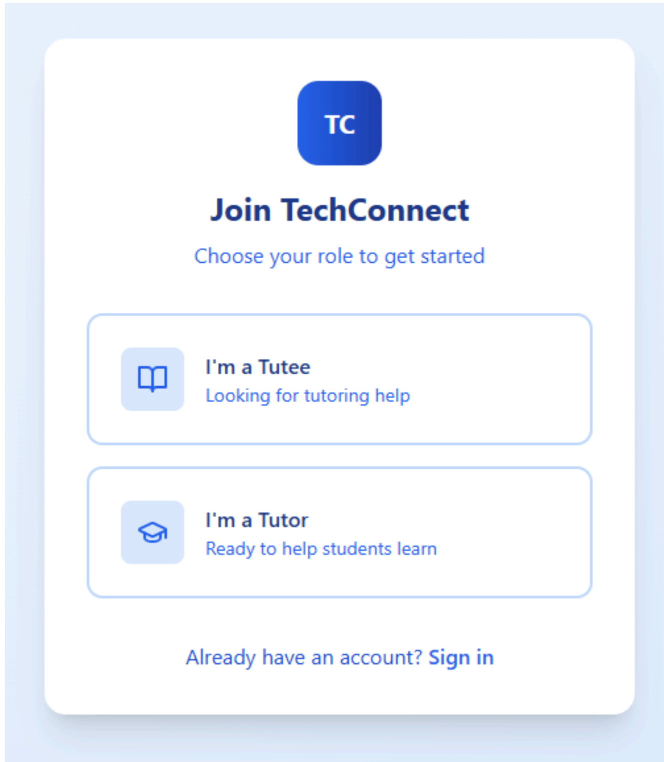

Figure 46. *Role-Selection Screen*

This figure illustrates the role-selection interface where new users choose Tutor or Tutee to tailor their experience.

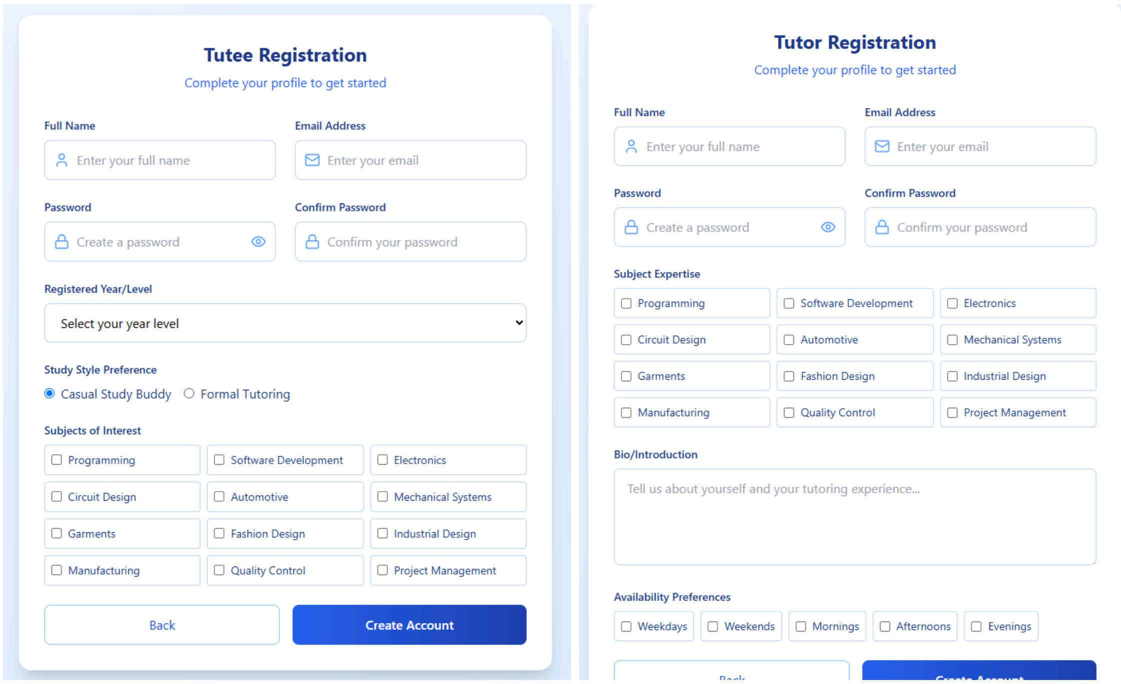

Figure 47. *Tutor & Tutee Registration Form*

This figure shows the registration form where users input personal details; tutors also specify expertise tags and preferences.

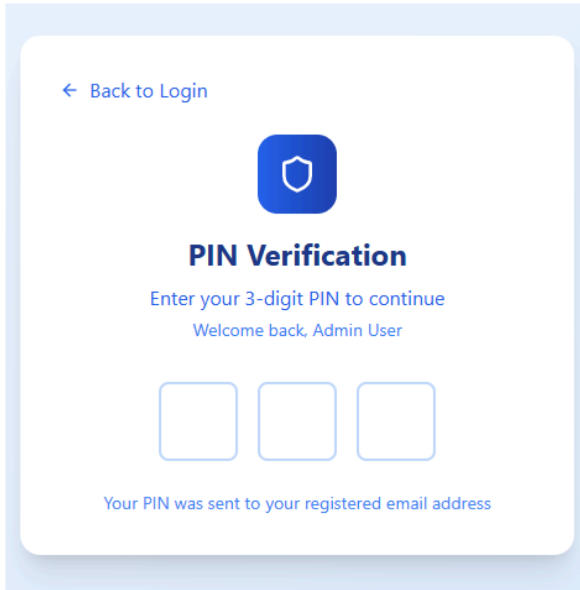

Figure 48. *PIN Verification Screen*

This figure depicts the PIN verification step in which users enter the one-time code sent via email to activate their account.

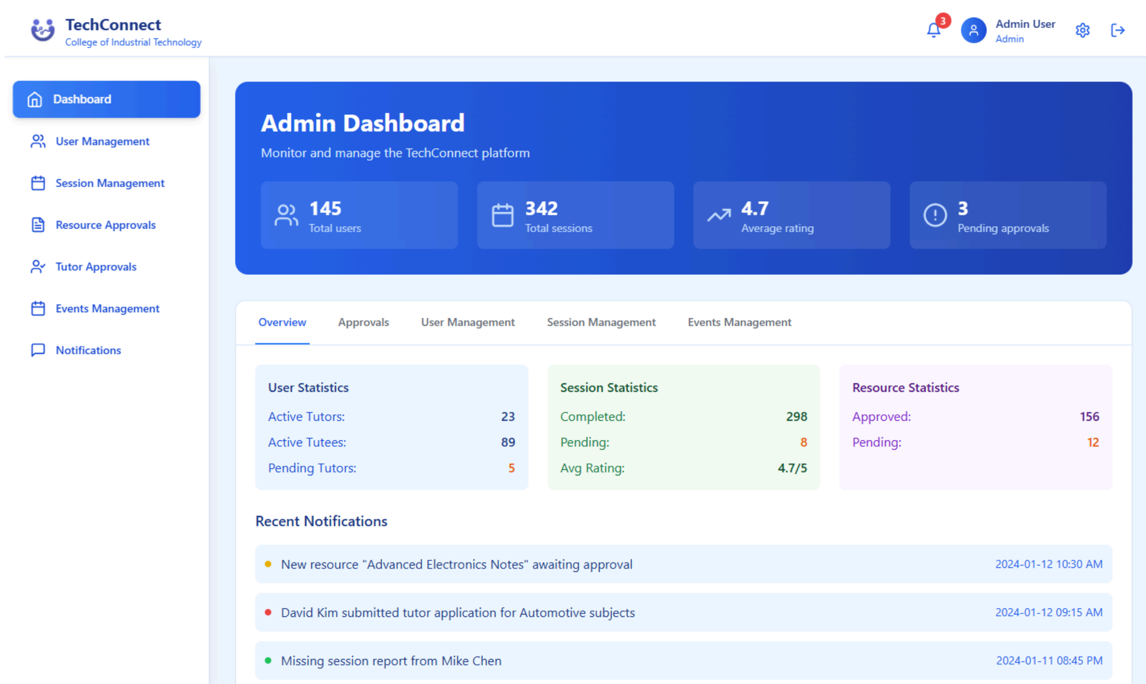

Figure 49. *Admin Dashboard*

This figure demonstrates the administrator’s dashboard, summarizing key metrics (user counts, active sessions, announcements) and providing links to management modules.

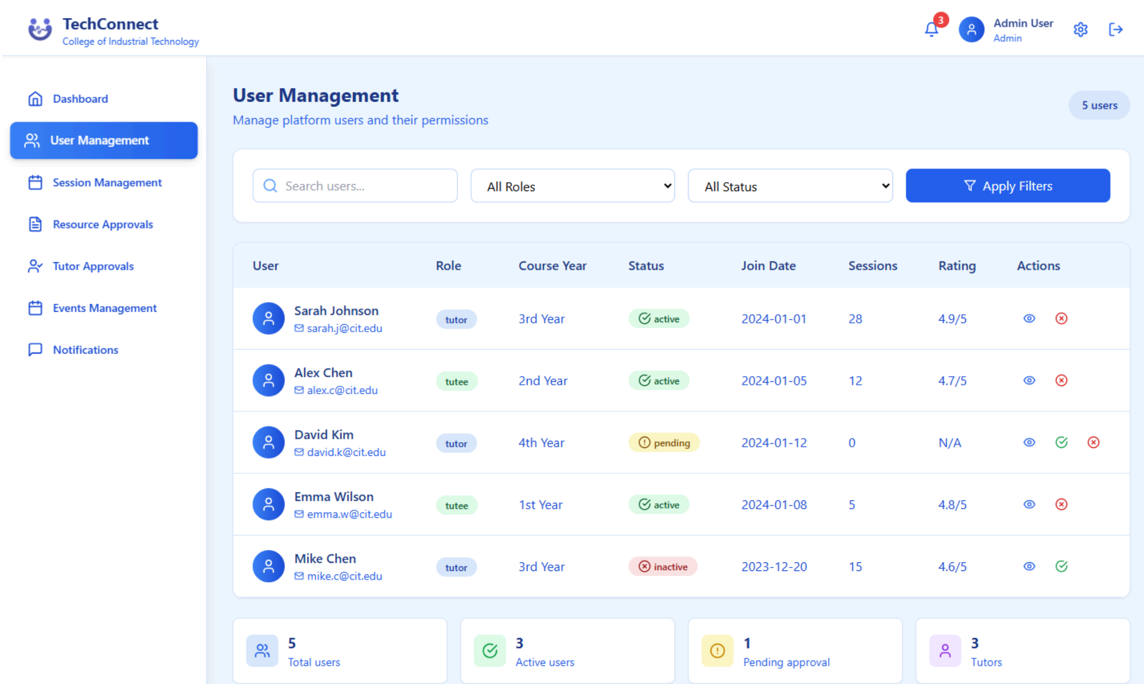

Figure 50. *User Management Module*

This figure illustrates the admin interface for searching, viewing, and activating/deactivating user accounts.

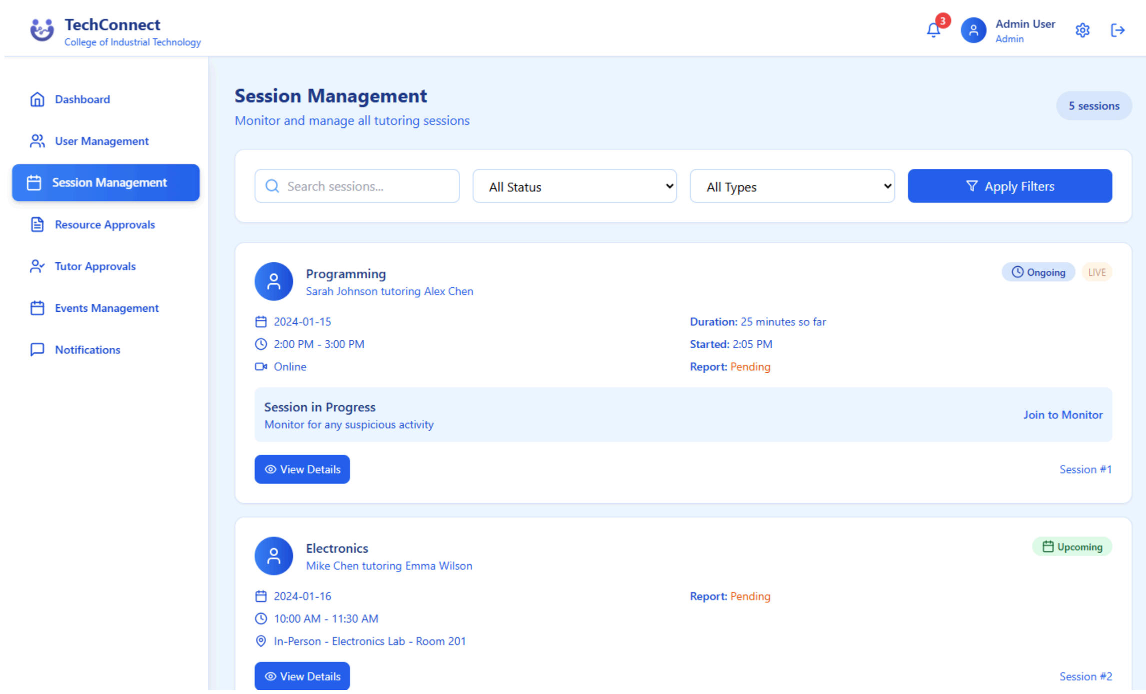

Figure 51. *Session Management*

This figure shows the admin view listing all tutoring sessions (requested, confirmed, canceled), with filters to review status and details.

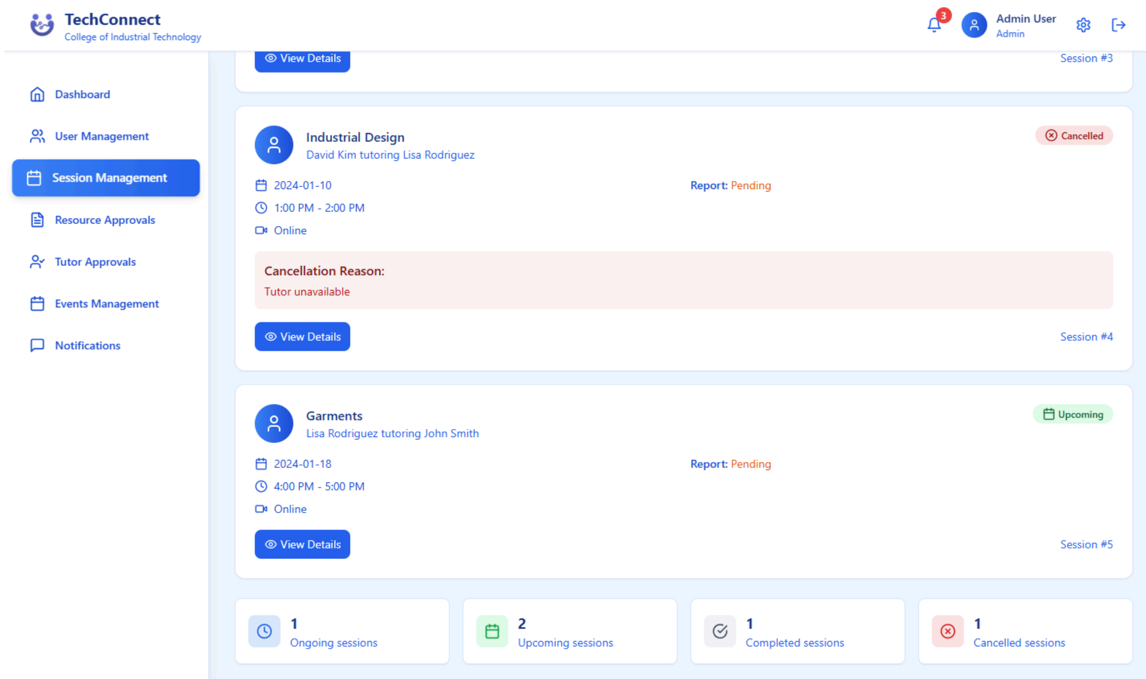

Figure 52. *Session Management*

This figure demonstrates the detailed session management list where administrators can inspect and manage individual session records.

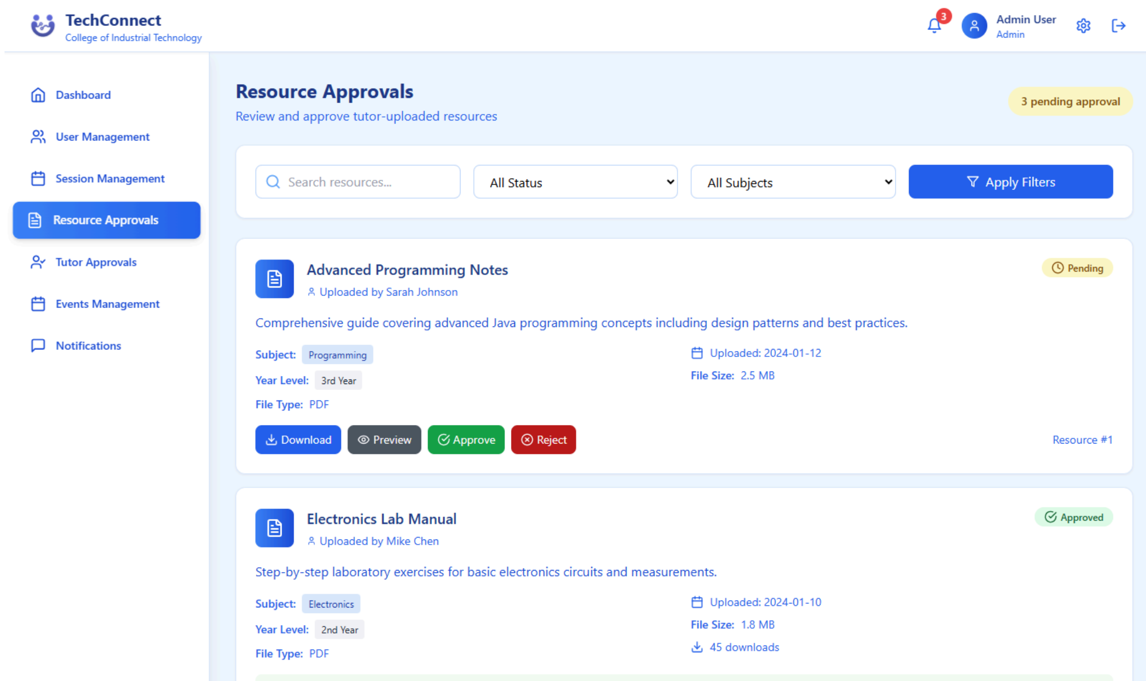

Figure 53. *Resource Approvals*

This figure illustrates the admin panel displaying tutor-submitted learning materials, allowing approval or rejection.

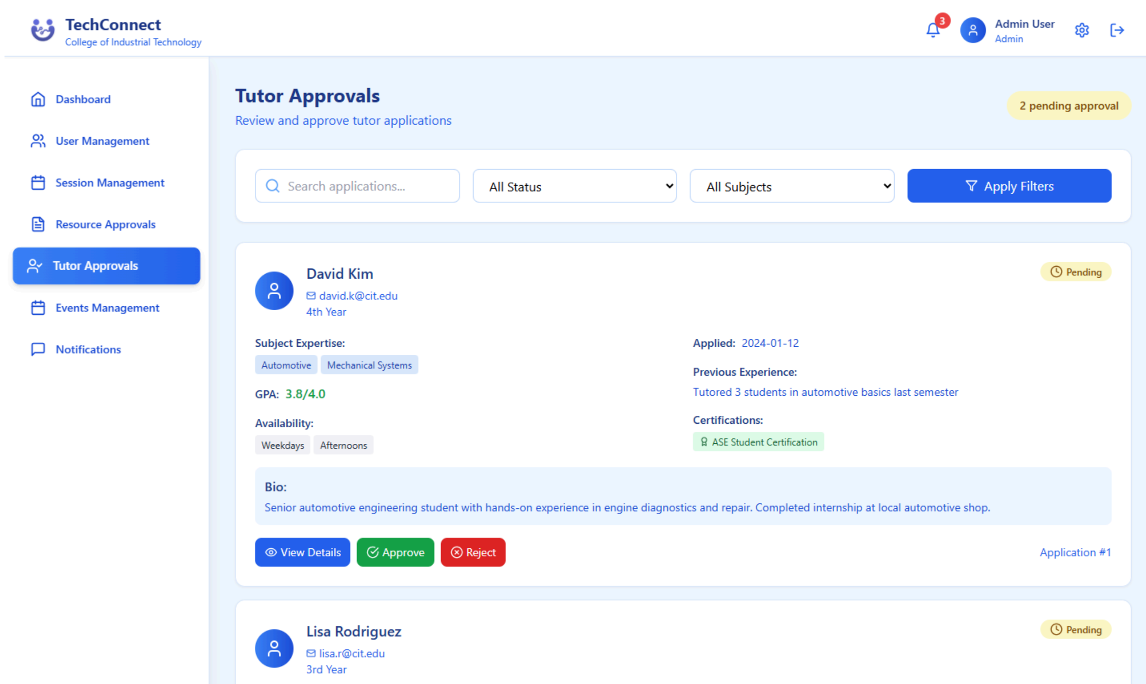

Figure 54. *Tutor Approvals*

This figure shows the admin module for reviewing and approving or declining pending tutor registration requests.

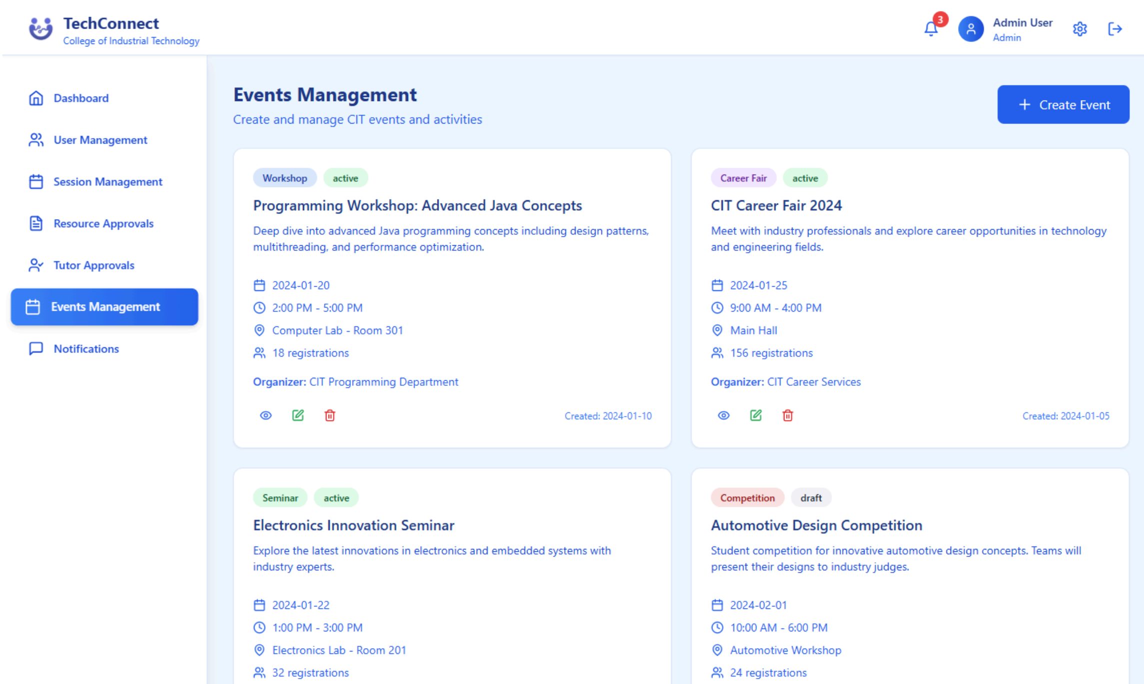

Figure 55. *Events Management*

This figure demonstrates the interface where administrators create or edit academic events (title, date/time, location, description) for the campus calendar.

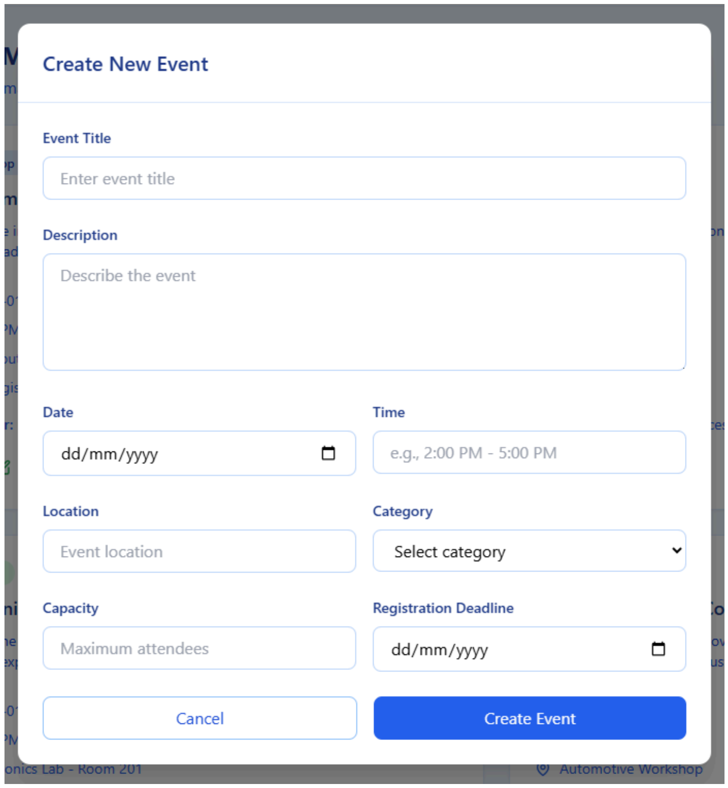

Figure 56. *Create New Event Form/Modal*

This figure illustrates the admin create new event form/modal, where event details can be entered and saved to the system.

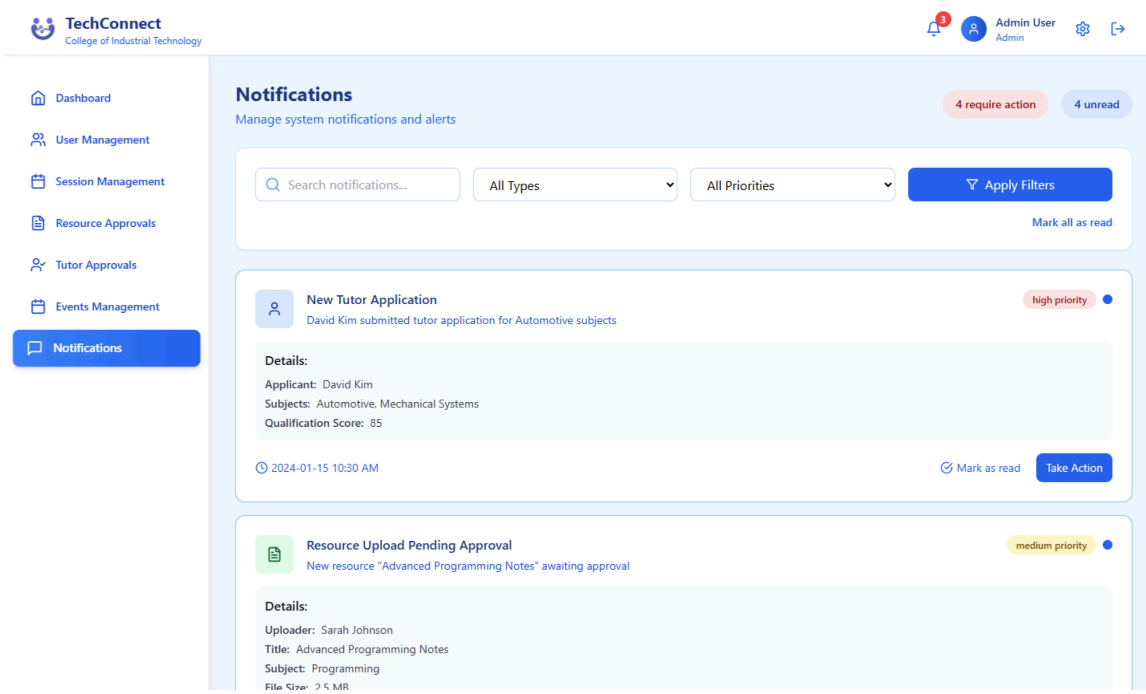

Figure 57. *Notifications Center*

This figure shows the full-page notification center listing all in-app alerts, grouped by type (session requests, reminders, announcements).

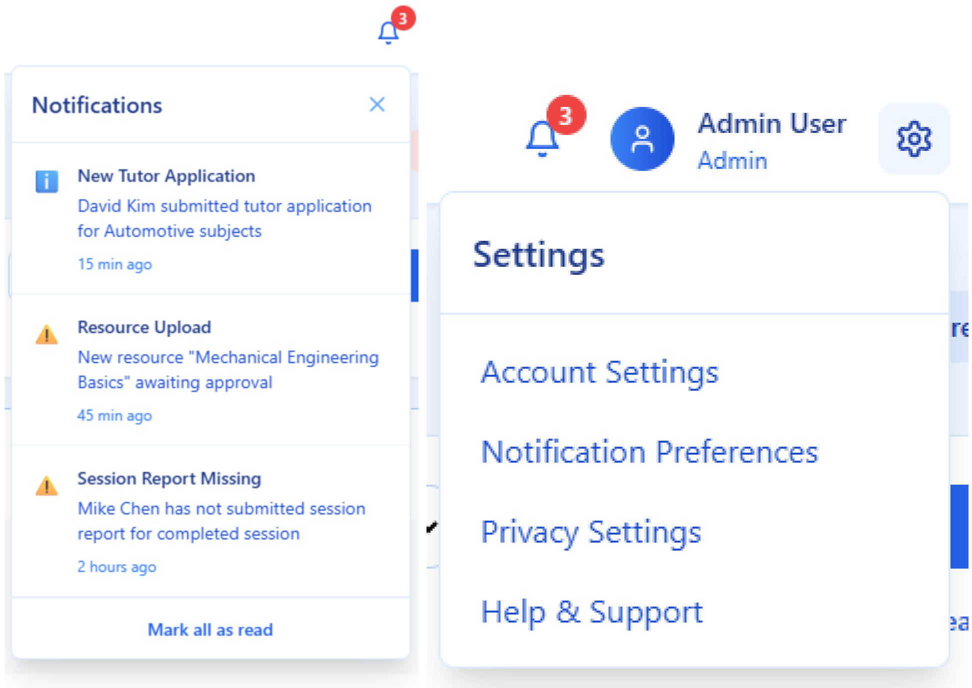

Figure 58. *Quick-Access Pop-overs (Notifications & Settings)*

This figure demonstrates the profile menu drop-down, giving access to user profile and application settings.

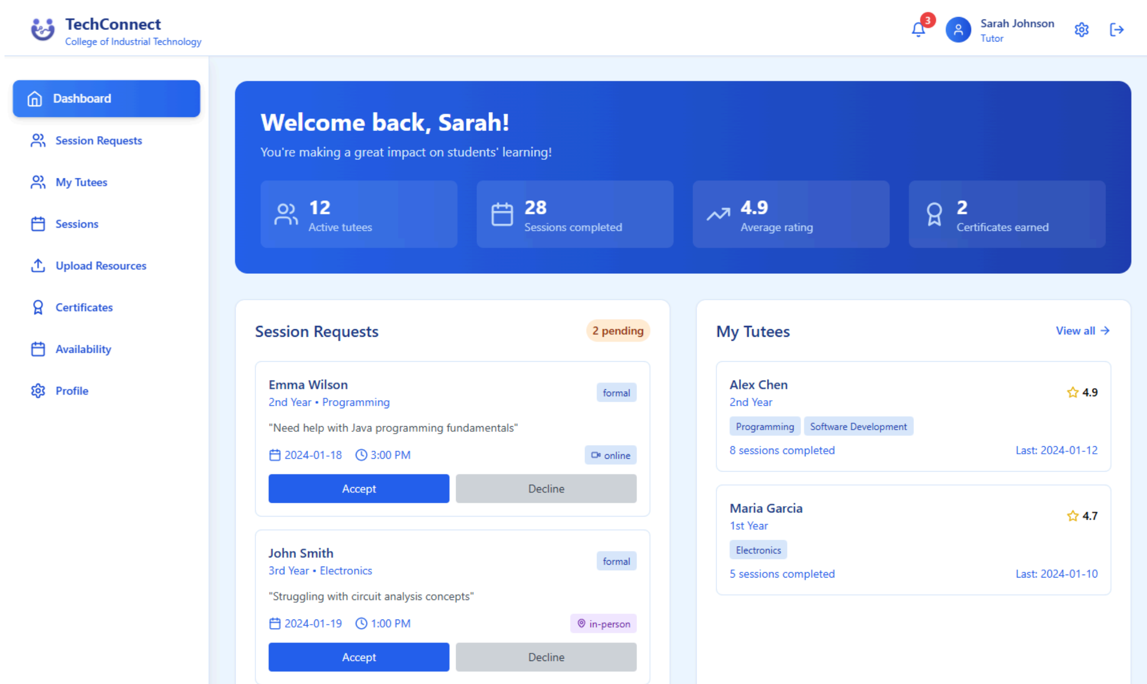

Figure 59. *Tutor Dashboard*

This figure illustrates the tutor's main dashboard, listing upcoming sessions and displaying performance metrics (average rating, session count, progress toward certificates).

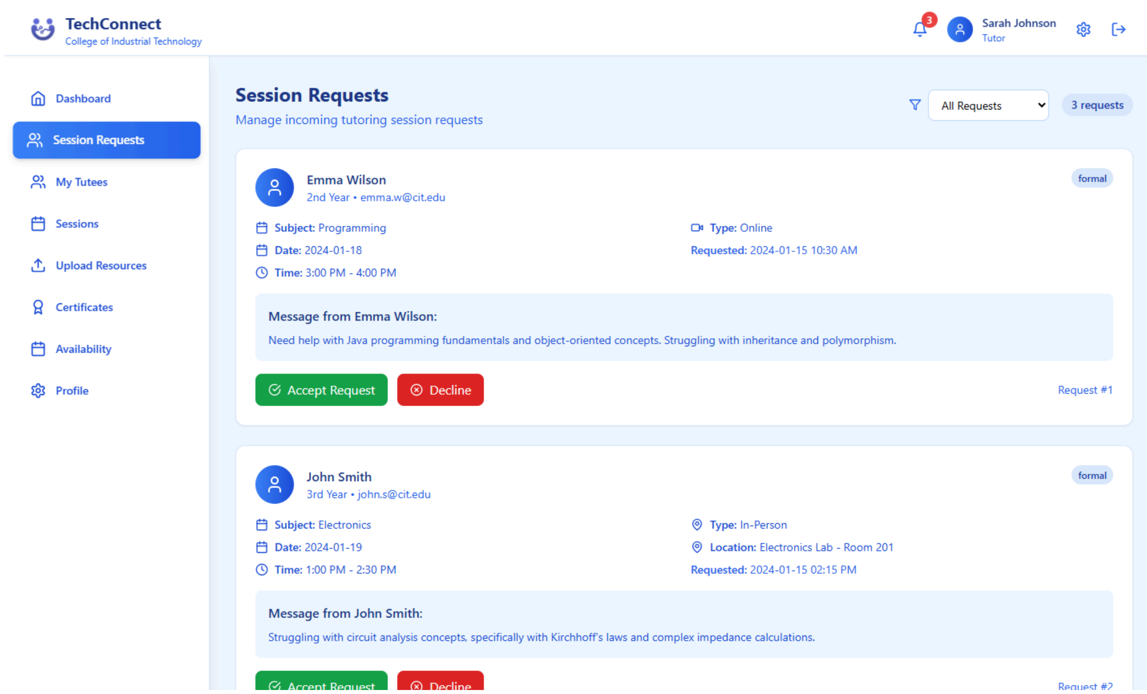

Figure 60. *Session Requests*

This figure shows the tutor's view of incoming session requests, with details (proposed time, subject, tutee name) and accept/decline controls.

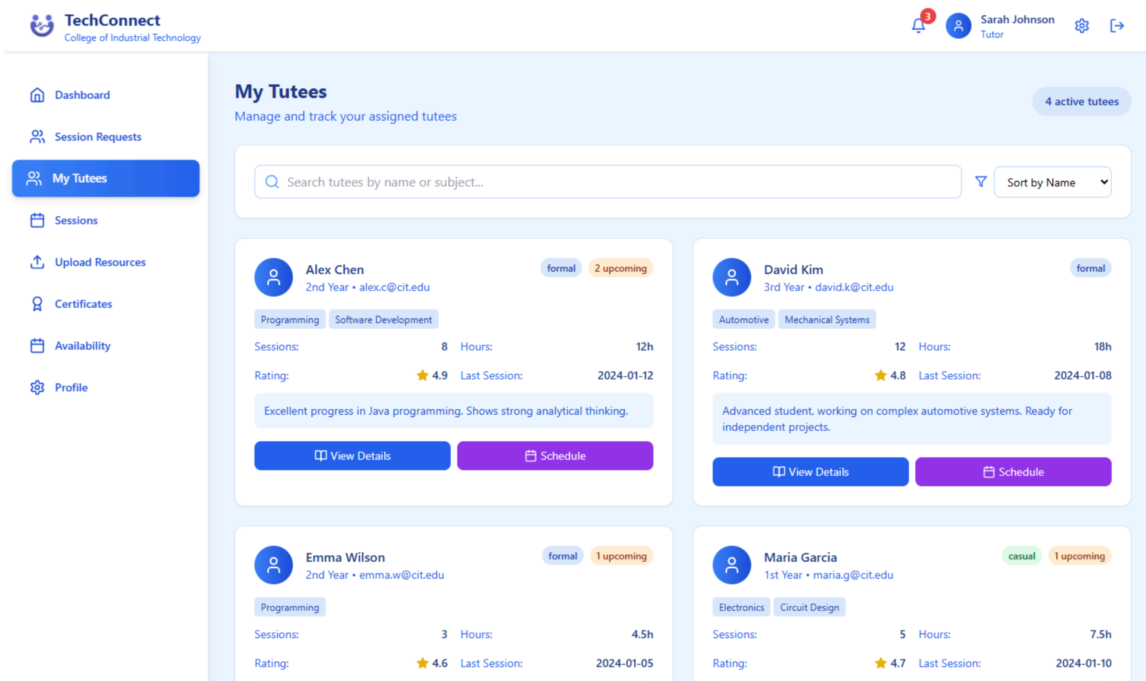

Figure 61. *My Tutees*

This figure demonstrates the tutor’s list of assigned tutees, showing each student’s name and contact info for ongoing sessions.

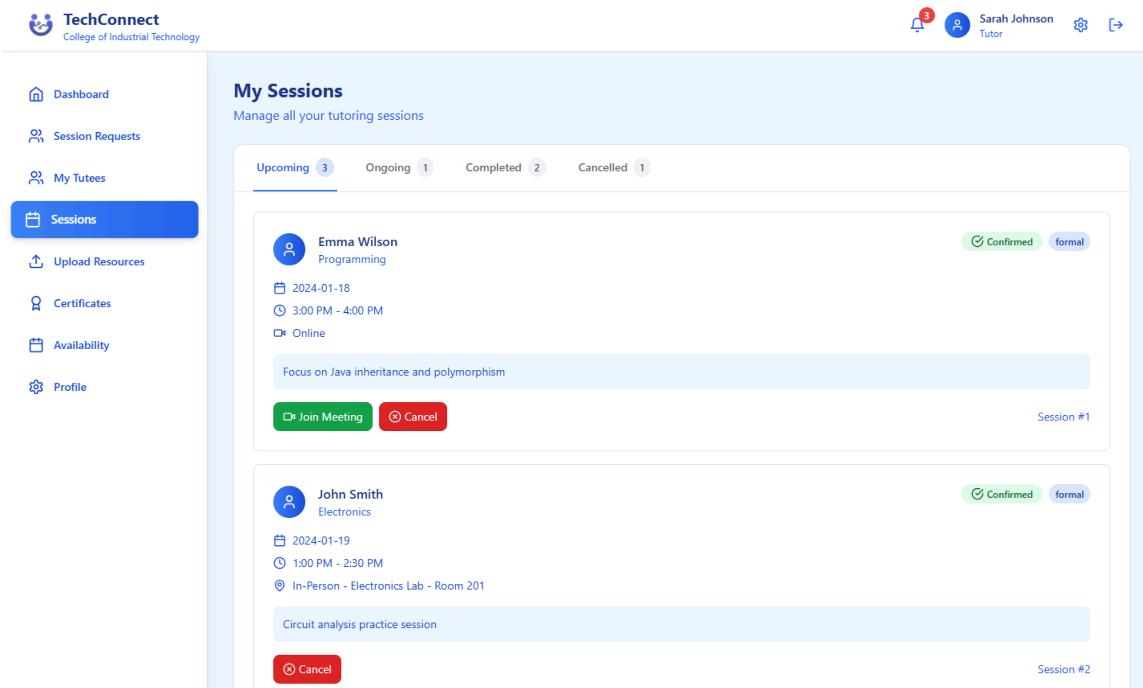

Figure 62. *My Sessions (Tutor View)*

This figure illustrates the tutor's session calendar or list, showing confirmed sessions (upcoming and past) with date, subject, and tutee details.

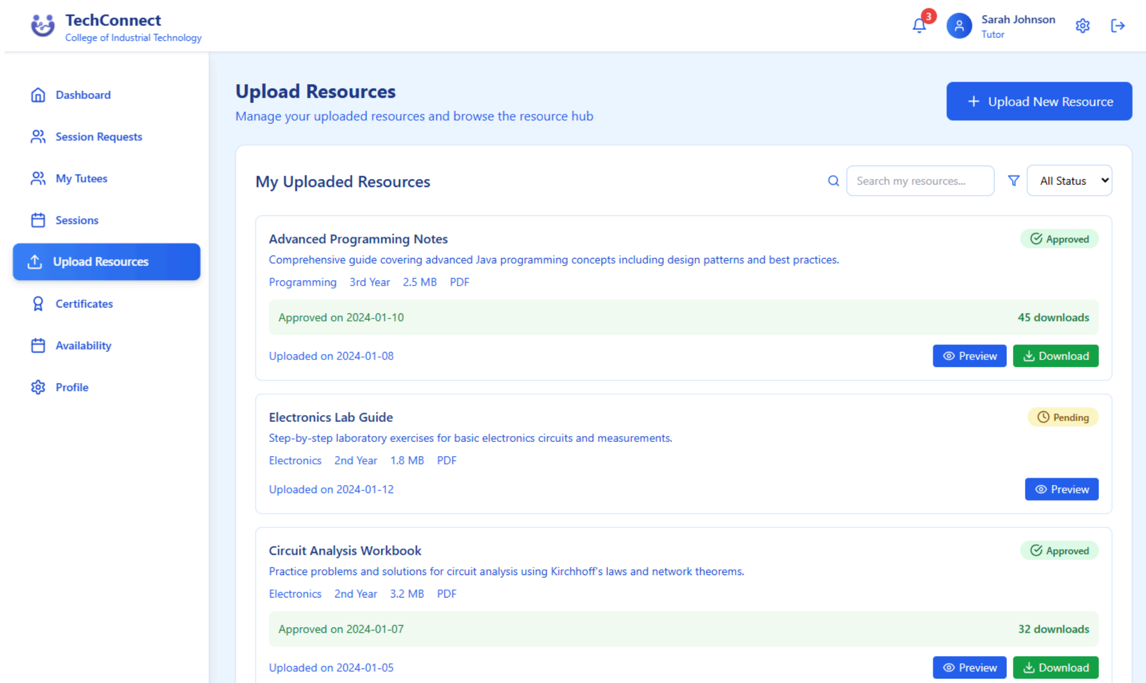

Figure 63. *Upload Resources (Tutor Hub)*

This figure shows the resource upload interface where tutors select a file, enter title, description, and subject tags to add materials.

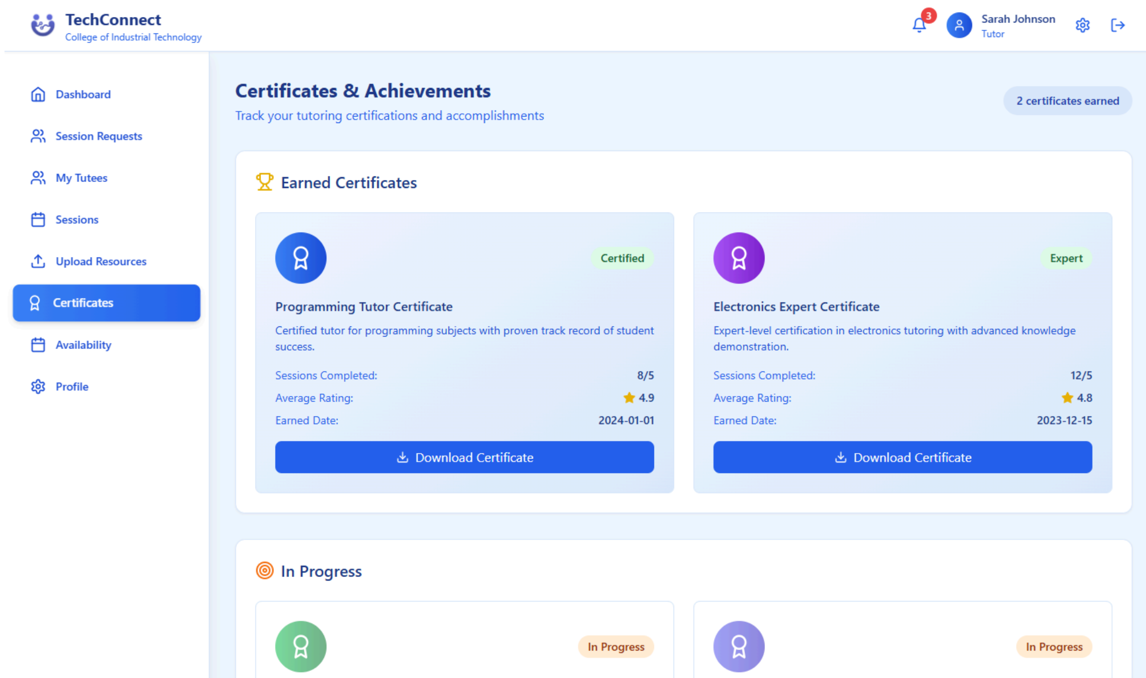

Figure 64. *Certificates & Achievements*

This figure illustrates the achievements page where users view and download earned certificates and badges (e.g., tutoring excellence).

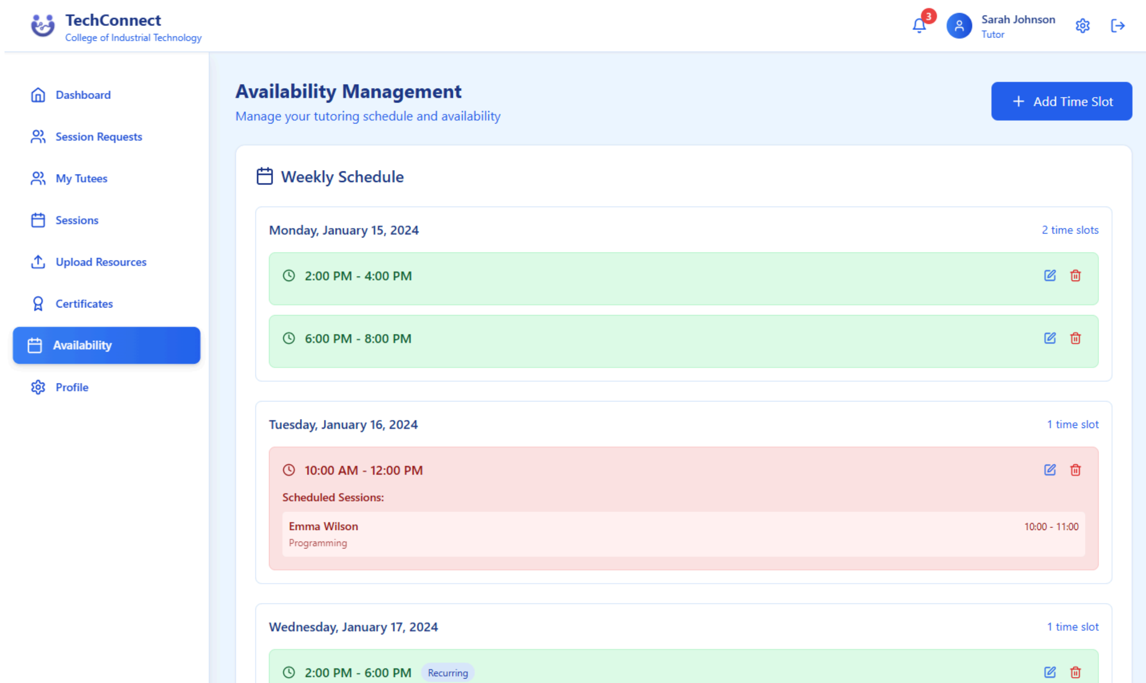

Figure 65. *Availability Management*

This figure demonstrates the interactive calendar where tutors select available time slots for booking by tutees.

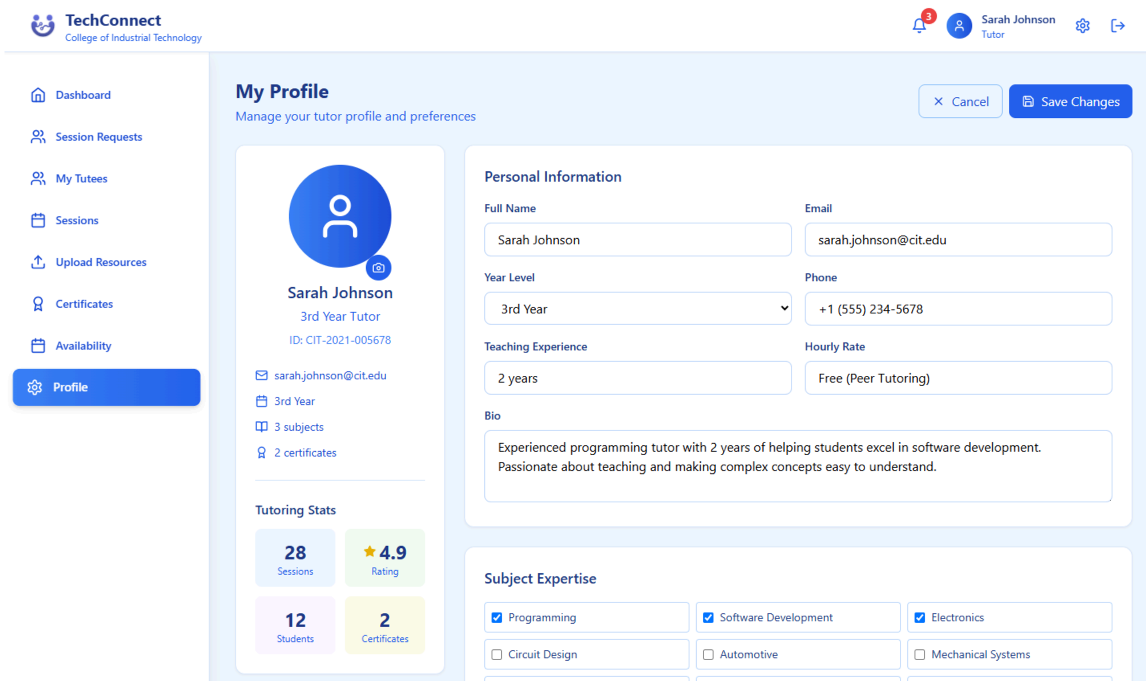

Figure 66. *My Profile (Tutor)*

This figure shows the tutor’s profile page displaying personal information and editable fields (subject expertise, photo).

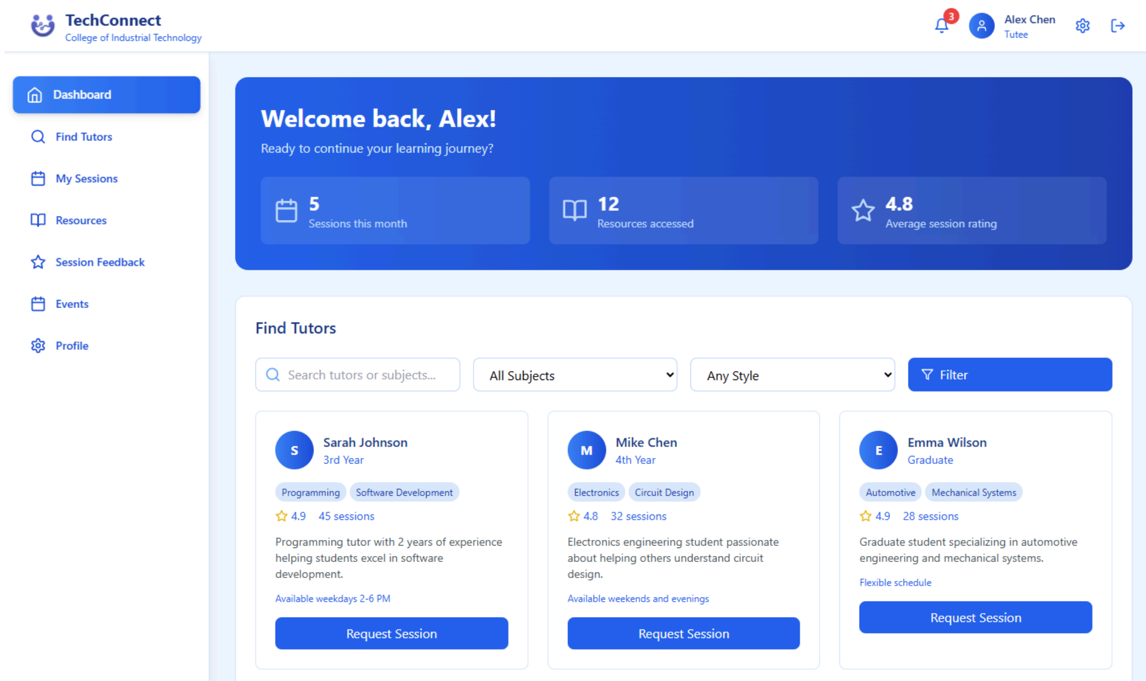

Figure 67. *Tutee Dashboard*

This figure illustrates the tutee's dashboard, showing pending requests, confirmed sessions, and a learning progress chart.

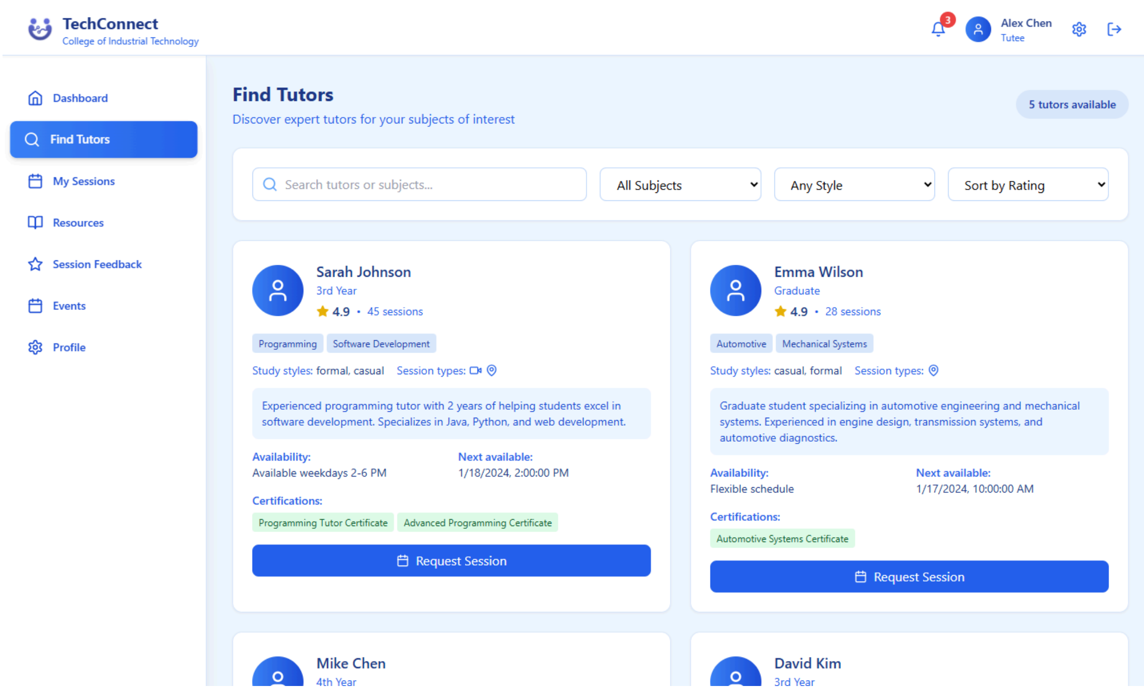

Figure 68. *Find Tutors (Detailed List)*

This figure demonstrates the search results listing matching tutors for a subject, showing name, expertise, rating, and next available slot.

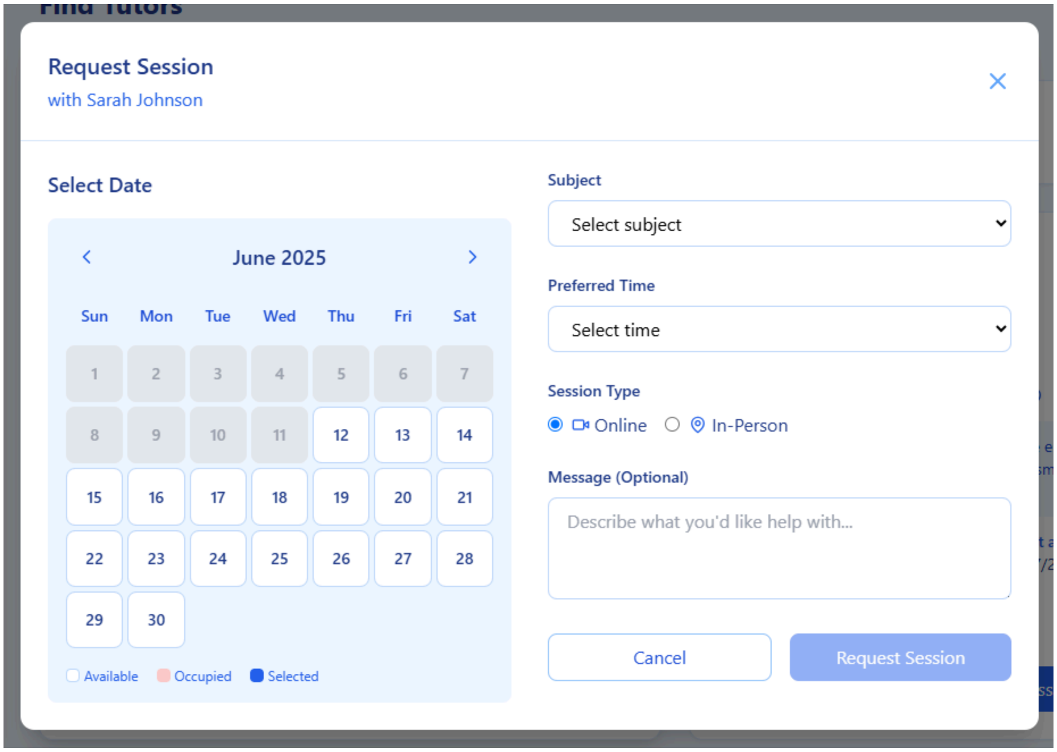

Figure 69. *Request Session Modal*

This figure shows the pop-up form where a tutee specifies date/time and session objective to request a tutoring session.

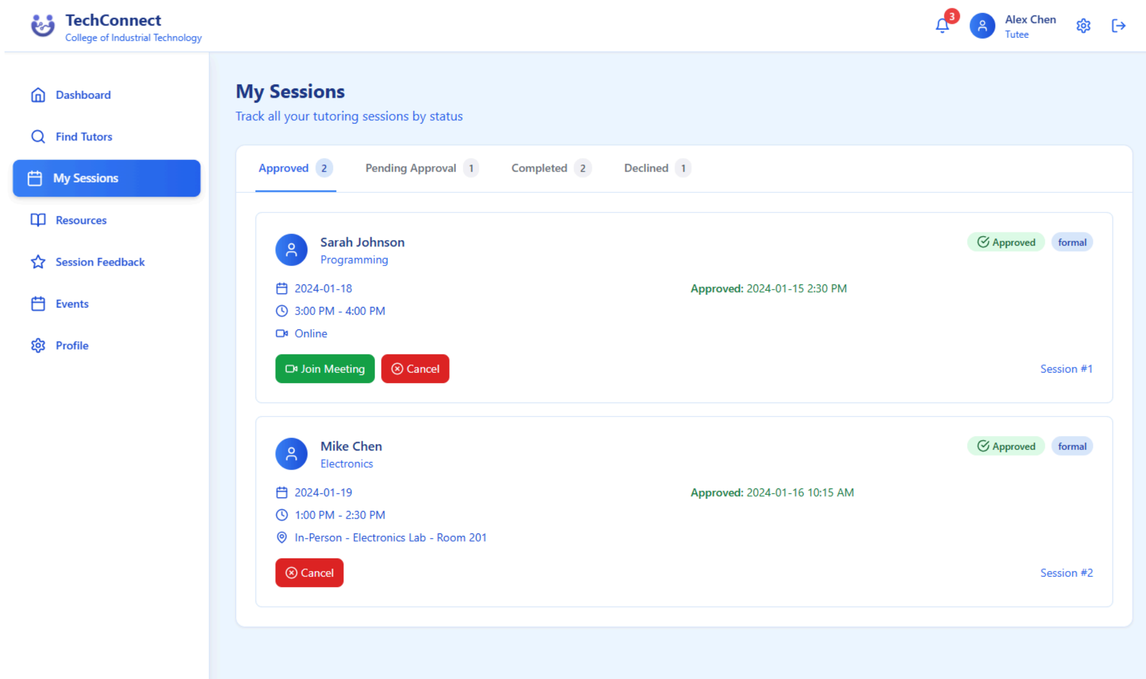

Figure 70. *My Sessions (Tutee View)*

This figure illustrates the tutee's session calendar or list, showing booked sessions (upcoming and past) with details.

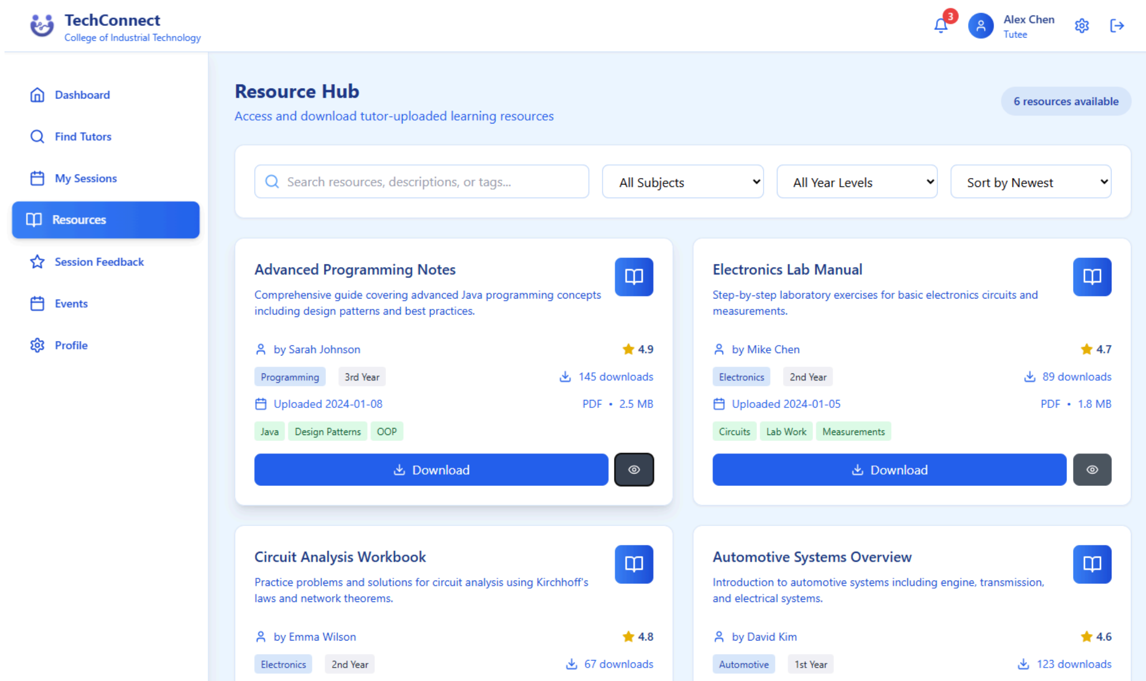

Figure 71. Resource Hub (Student)

This figure shows the materials library where tutees search and filter educational resources by subject, type, or keywords.

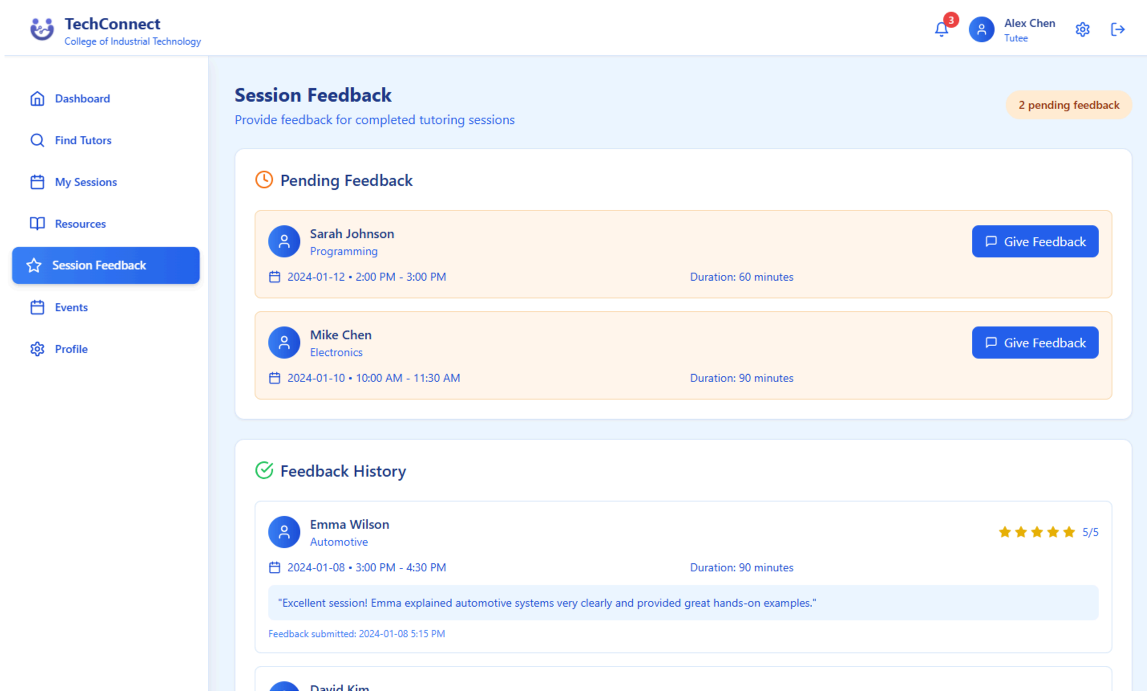

***Figure** 72. Events (Tutee View)*

This figure demonstrates the tutee's list view of upcoming academic events, showing titles, dates, and locations.

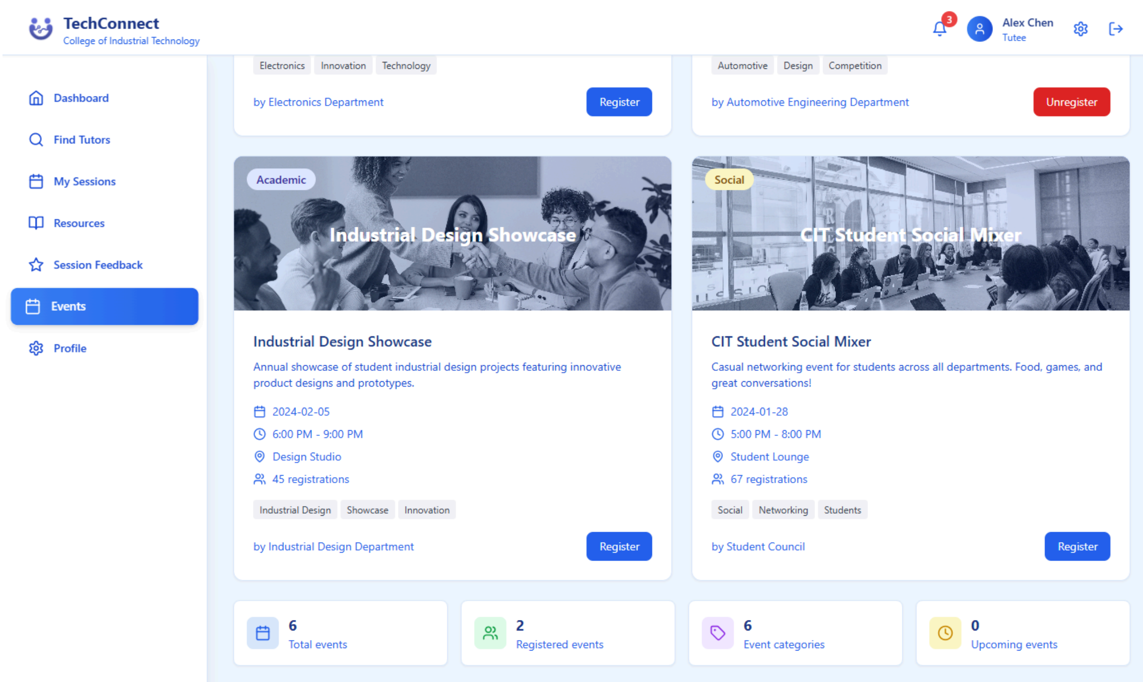

***Figure* 73.** *Events (Tutee View)*

This figure illustrates the tutee's calendar view of academic events; selecting an event reveals details (date, time, description, location).

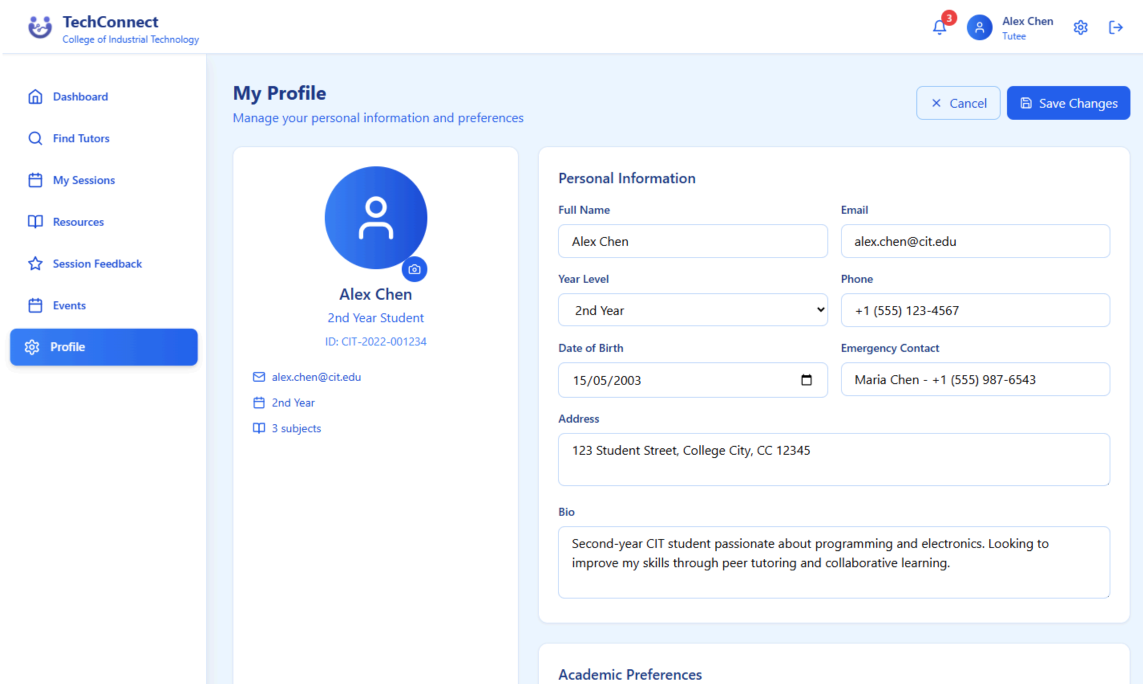

Figure 74. *My Profile (Tutee)*

This figure shows the tutee's profile page with their personal information (name, program, contact info, student ID).

Implementation Plan

TechConnect deployment will begin after the capstone defense and approval by the College Dean and University IT Services. At that point, researchers will transfer the full system, source code, hosting configuration, and user manual, to CIT IT personnel, who will take over maintenance and updates. A formal handover agreement will mark this transition, after which researchers will no longer provide routine support. The table below details the strategies, activities, responsible parties, and timelines for moving from development to operation.

Strategy	Activities	Persons Involved	Duration
Securing Institutiona l Approval	● Prepare and submit a formal request letter to the CIT Dean and University IT Services detailing project objectives, system capabilities, and required resources. ● Follow up on any inquiries or requested clarifications from the approvers.	Researchers, CIT Department Head, University IT Staff	3 Days
System Deploymen t	● Configure the hosting environment on Hostinger (Node.js plan). ● Deploy the Next.js-based React application build together with Node.js serverless functions. ● Set up Supabase services (Authentication, Database Schema, Storage, and Real-Time Ruless). ● Verify domain configuration and SSL certificate issuance.	Researchers, CIT IT Personnel	1 Day

Distribution of User Manuals	• Finalize and print the user manual, covering account management, session booking, resource uploads, dashboards, and troubleshooting guidance. • Distribute printed and digital copies of the user manual to CIT IT Personnel, peer tutors, and faculty advisors.	Researchers, CIT IT Personnel, Peer Tutors, Faculty Advisor	1 Day
Training Sessions	• Conduct hands‑on workshops for: ○ CIT IT Personnel (system configuration, user account administration, analytics reporting). ○ Peer Tutors (creating/maintaining profile, managing availability, conducting sessions, uploading resources). ○ Tutees (account registration, searching for tutors, booking sessions, accessing resources). ○ Administrators (reviewing dashboards, generating reports, moderating content). • Collect feedback and answer questions in real time to ensure comfort with all system functions.	Researchers, CIT IT Personnel, Peer Tutors, Tutees, Administrators	3 Days

Table 1. *Implementation Plan of TechConnect System*

The approval request letter will summarize TechConnect's features, requirements, and training plan. The researchers will coordinate with the CIT Department Head to arrange demonstrations for University IT Services and respond to any inquiries. Prior to handover, the team will complete end-to-end testing in a staging environment that mirrors production. During deployment, the researchers and CIT IT staff will configure the Hostinger environment, deploy the Next.js-based React front end, publish the Node.js serverless functions, and integrate Supabase services, including authentication, database schema, file storage rules, and real-time policies. Domain setup and SSL certificates will also be validated to ensure secure system access.

A comprehensive user manual—covering account management, session booking, resource uploads, dashboard navigation, and troubleshooting—will be finalized and distributed in both printed binders and PDF copies to CIT IT personnel, peer tutors, and faculty advisers. Over three days, role-based workshops will be conducted in a production-mirrored lab: CIT IT staff will be trained on system administration and analytics; peer tutors on profile management, availability, tutoring sessions, and resource uploads; tutees on registration, tutor search, session booking, and resource access; and administrators on dashboard review, report generation, and content moderation. Feedback will be collected and questions addressed in real time to ensure all user groups become proficient in using TechConnect.

Research Instrument

The researchers utilized a checklist type questionnaire employing a Likert scale to gather evaluative data from respondents regarding the TechConnect peer tutoring platform. This survey instrument contained statements that allowed participants to rate various aspects of the system, with response options ranging from “acceptable” to “unacceptable.” Utilizing this structured assessment approach enabled the systematic collection and statistical analysis of data related to the system’s usability, functionality, and overall software quality, facilitating the identification of both strengths and potential areas for enhancement.

The questionnaire was thoughtfully developed with close collaboration among the researchers, ensuring clarity and comprehensibility of each statement to accurately capture respondents’ perceptions. The content of the questionnaire was carefully adapted from the specific objectives of the study, guided by the ISO 25010 software quality standards. Specifically, respondents evaluated the system based on essential software quality characteristics, including functional suitability, performance efficiency, compatibility, interaction capability, reliability, security, maintainability, flexibility, and safety.

These quality attributes were assessed in relation to key functionalities of TechConnect, such as user registration and authentication, tutor tutee matching accuracy, session scheduling, resource management capabilities, responsiveness of user interfaces, notification effectiveness, and system security measures. This comprehensive approach ensured that feedback collected from respondents comprising CIT students, tutors, and administrators was directly relevant to the practical and academic context in which TechConnect operates at Southern Luzon State University, College of Industrial Technology.

Respondents of the Study

Students from SLSU's CIT will participate in the study as the main respondents who will assess the TechConnect platform. The researchers selected CIT students because they are the intended users of this peer tutoring system. CIT was specifically chosen due to its technical programs that require specialized knowledge and hands-on skills, making peer tutoring particularly valuable for students who need help with complex technical concepts and practical applications that are unique to industrial technology fields.

The respondents will include students from different year levels within CIT, representing various programs such as Information Technology, Automotive Technology, Electronics Technology, Electrical Technology, Food Technology, Mechanical Technology, Print and Media Technology, and Garment Technology. This diverse group ensures that the system evaluation covers the needs of students across all technical disciplines offered by the department.

To make the evaluation feasible within the given two-week timeframe, the researchers will invite approximately 20 to 30 students to serve as end-user respondents, with a purposive balance of tutees, tutors, and at least a few administrators to represent all roles within the system. The selection of respondents will be carried out through non-probability convenience sampling combined with purposive stratification, ensuring that participants come from different programs and year levels for comprehensive feedback.

In addition to student participants, the study will also involve five to seven IT experts who will conduct a heuristic and technical evaluation of the system. These experts will be selected through purposive sampling based on their knowledge and experience in web applications, user interface design, and system quality assessment. Their evaluation will complement the feedback from students by identifying usability and technical issues aligned with software quality standards.

The participation of both end-users and IT experts will provide a balanced evaluation of TechConnect, ensuring that the system is assessed not only for usability and relevance to students but also for its technical robustness and quality.

Data Gathering Procedure

The researchers will use a survey questionnaire to collect feedback and evaluate the TechConnect peer tutoring platform. The respondents will be students from Southern Luzon State University specifically from the College of Industrial technology, including both potential tutors and tutees who represent the platform's target users.

The questionnaire contains questions to gather valuable insights about the platform's effectiveness, usability, and overall performance. This feedback will help the researchers evaluate the system based on research objectives and determine if it meets the ISO 25010 standards.

The researchers will explain the study's purpose to all the respondents before distributing the survey. Respondents will be given time to explore the platform and test its key features such as session scheduling, tutor-tutee matching system, and the resource hub where study materials are shared. After the respondents have the chance to use the TechConnect and explore its features, they will be asked to complete the survey questionnaires. The researchers will assure all respondents that their responses will remain completely confidential.

Once all surveys are collected, the researchers will organize and analyze the responses. The researchers will use statistical methods to interpret the results and assess how well TechConnect performs as a peer tutoring solution for CIT students. This analysis will help determine whether the platform successfully addresses the academic support needs of students in the College of Industrial Technology and meets the quality standards expected from educational technology systems.

Statistical Treatment

For the statistical evaluation of survey responses, the researchers use a 4-point Likert Scale to assess how respondents perceived the effectiveness of the system. The scale was structured with values ranging 1 (Unacceptable), 2 (Slightly Unacceptable), 3 (Slightly Acceptable), and 4 (Acceptable). In processing the collected data, the researchers computed weighted mean values in each dimension. With this method, the researchers were able to assess the responses and conduct a thorough assessment of the system's overall performance.

Formula:

$$\mathbf{WM} = \frac{\underline{4\,(\mathbf{f}) + 3\,(\mathbf{f}) + 2\,(\mathbf{f}) + 1\,(\mathbf{f})}}{\mathbf{N}}$$

Where:

WM – Weighted Mean | **f** - Frequency | **N** - Total number of respondents

Scale	Range Interval	Qualitative Description
4	3.25 - 4.00	Acceptable (A)
3	2.50 - 3.24	Slightly Acceptable (SA)
2	1.75 - 2.49	Slightly Unacceptable(SU)
1	1.00 - 1.74	Unacceptable (U)

Table 2. *Likert Scale*

The system's efficiency was evaluated using a four-point Likert scale instrument, with each scale value corresponding to a defined range of weighted means and an associated qualitative rating. In this scheme, a weighted mean of 3.25 to 4.00 is interpreted as Acceptable (A), 2.50 to 3.24 as Slightly Acceptable (SA), 1.75 to 2.49 as

Slightly Unacceptable (SU), and 1.00 to 1.74 as Unacceptable (U). This even-point scale (lacking a neutral midpoint) was chosen to compel respondents to form a clear positive or negative inclination in their ratings of the peer tutoring platform's efficiency, thereby avoiding ambiguous "neutral" responses.

Using a Likert type scale allows respondents to independently express the intensity of their opinions on the system's performance (Rokeman, 2024). These scales are valued in educational technology for converting subjective perceptions into quantifiable data, providing richer feedback (Koo & Yang, 2025). Recent studies highlight the importance of tailoring the scale's design, choosing the right number of response options and clear labels to fit the research context, as too few or too many options can limit clarity and nuance (Lai et al., 2022; Rokeman, 2024). Additionally, precise wording of items is essential to ensure reliability and validity, as ambiguity can cause bias and inconsistent responses (Zeng et al., 2024). Well designed Likert items yield more accurate and trustworthy data, improving the evaluation's ability to measure system efficiency effectively (Rokeman, 2024).

References:

Accreditly. (2024). Make the web accessible with Tailwind CSS. https://accreditly.io/articles/make-the-web-accessible-with-tailwind-css

Anifa, M., Ramakrishnan, S., Kabiraj, S., & Joghee, S. (2024). Systematic review of literature on agile approach. SAGE Open, 14(3). https://doi.org/10.1177/09711023241272294

Appic Softwares. (2024). React JS education app development 2024. https://appicsoftwares.com/blog/react-js-education-app-development/

Arc42. (2023). Update on ISO 25010, version 2023. Arc42 Quality Model. https://quality.arc42.org/articles/iso-25010-update-2023

Arco-Tirado, J. L., Fernández-Martín, F. D., & Hervás-Torres, M. (2019). Evidence-based peer-tutoring program to improve students' performance at the university. Studies in Higher Education, 45(11), 2190–2202.

Asana. (2025). What is Kanban? A beginner's guide for agile teams. Asana Resources. https://asana.com/resources/what-is-kanban

Atlassian. (2025). Kanban - A brief introduction. Atlassian Agile Coach. https://www.atlassian.com/agile/kanban

Barz, N., Benick, M., Dörrenbächer-Ulrich, L., Perels, F., & Merceron, A. (2024). Students' acceptance of e-learning: Extending the technology acceptance model with self-regulated learning and affinity for technology. Discover Education, 3(1), Article 114. https://doi.org/10.1007/s44217-024-00195-7

Baturay, M. H., & Birtane, M. (2013). Responsive web design: A new type of design for web-based instructional content. Procedia-Social and Behavioral Sciences, 106, 2275–2279.

Binariks. (2024). Education data interoperability and why APIs are a solution. https://binariks.com/blog/data-interoperability-in-educational-technology-and-why-you-need-apis-for-it/

BusinessWorld. (2024, September 9). The digital shift: Transforming PHL education through cloud-based edtech. https://www.bworldonline.com/special-reports/2024/09/09/619444/the-digital-shift-transforming-phl-education-through-cloud-based-edtech/

Cal.com. (2024). Open scheduling infrastructure. https://cal.com/

Chen, H., Shi, N., Chen, L., & Lee, R. (2024). Enhancing educational Q&A systems using a Chaotic Fuzzy Logic-Augmented large language model. Frontiers in Artificial Intelligence, 7, 1404940. https://doi.org/10.3389/frai.2024.1404940

Codacy. (2023). An exploration of the ISO/IEC 25010 software quality model. Codacy Blog. https://blog.codacy.com/iso-25010-software-quality-model

Cronofy. (2024). Scheduling automation software. https://www.cronofy.com/

DataCamp. (2024). Fuzzy string matching in Python tutorial. https://www.datacamp.com/tutorial/fuzzy-string-python

Dong, H., Dacre, N., Baxter, D., & Ceylan, S. (2024). What is agile project management? Developing a new definition following a systematic literature review. Project

Management Journal, 55(4), 87569728241254095. https://doi.org/10.1177/87569728241254095

Flatirons. (2025). What is user acceptance testing? A guide in 2025. Flatirons Development Blog. https://flatirons.com/blog/what-is-acceptance-testing/

GeeksforGeeks. (2024). FuzzyWuzzy Python library. https://www.geeksforgeeks.org/python/fuzzywuzzy-python-library/

Green, B. (2023). Adaptive fuzzy string matching: How to merge datasets with only one (messy) identifying field. Political Analysis, 30(4), 275–290. https://doi.org/10.1017/pan.2022.15

Haider, M., & Yasmin, A. (2015). Significance of scaffolding and peer tutoring in the light of Vygotsky's theory of zone of proximal development. International Journal of Languages, Literature and Linguistics, 1(3), 170–173.

Hyperlink InfoSystem. (2025). The 20 best education APIs in 2025. https://www.hyperlinkinfosystem.com/research/the-20-best-education-apis-in-2020

IEEE Xplore. (2021). Comprehensive integration of API usage patterns. https://ieeexplore.ieee.org/document/9463023/

Indonesian Cloud. (2024). Learning management system. https://indonesiancloud.com/learning-management-system/

Inflectra. (2025). Kanban methodology: Agile software development. Inflectra Solutions. https://www.inflectra.com/Solutions/Methodologies/Kanban.aspx

Inquirer Business. (2024). Education technology comes of age. https://business.inquirer.net/435063/education-technology-comes-of-age

International Organization for Standardization. (2023). ISO/IEC 25010:2023 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model. https://www.iso.org/standard/78176.html

ISO25000. (2024). ISO/IEC 25010. ISO 25000 Software Product Quality. https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

Jaiswal, I. A., & Jain, E. A. (2025). API design and integration in a microservices environment. International Journal for Research Publication and Seminar, 16(1), 409–419. https://doi.org/10.36676/jrps.v16.i1.205

Keller, P. (2024, December 12). Scaling peer tutoring at the University of New Hampshire: A partnership with Knack [Blog post]. Knack.

Kemp, A. (2024). Testing a novel extended educational technology acceptance model using student attitudes towards virtual classrooms. British Journal of Educational Technology, 55(4), 1543–1567. https://doi.org/10.1111/bjet.13440

Koo, M., & Yang, S.-W. (2025). Likert-type scale. Encyclopedia, 5(1), 18. https://doi.org/10.3390/encyclopedia5010018

Kumar, S., et al. (2020). The role of scaffolding in the instructional design of online, self-directed, inquiry-based learning environments: Student engagement and learning approaches. Computers & Education, 156, 103–117.

Kuo, Y.-C., Yao, C.-B., & Wu, Z.-Y. (2022). Online peer-tutoring for programming languages based on programming ability and teaching skill. Applied Sciences, 12(17), 8513. https://doi.org/10.3390/app12178513

Lai, J. W. M., De Nobile, J., Bower, M., & Breyer, Y. (2022). Comprehensive evaluation of the use of technology in education: Validation with a cohort of global open online learners. Education and Information Technologies, 27(7), 9877–9911. https://doi.org/10.1007/s10639-022-10986-w

Ley, T., Kump, B., & Gerdenitsch, C. (2010). Scaffolding self-directed learning with personalized learning goal recommendations. In User Modeling, Adaptation, and Personalization (pp. 75–86). Springer.

Lin, C. C., Huang, A. Y. Q., & Lu, O. H. T. (2023). Artificial intelligence in intelligent tutoring systems toward sustainable education: A systematic review. Smart Learning Environments, 10, 41. https://doi.org/10.1186/s40561-023-00260-y

Lin, P., Zhou, Q., Ma, J., Sun, M., Wang, Y., Chen, L., & Bian, C. (2025). Peer tutoring in higher education: Power from pedagogical training. Humanities and Social Sciences Communications, 12, 723. https://doi.org/10.1057/s41599-025-04860-6

Lin, Y., & Yu, Z. (2023). Extending Technology Acceptance Model to higher-education students' use of digital academic reading tools on computers. International Journal of Educational Technology in Higher Education, 20(1), Article 34. https://doi.org/10.1186/s41239-023-00403-8

Marker.io. (2024). User acceptance testing (UAT): Meaning, definition, process. Marker.io Blog. https://marker.io/blog/user-acceptance-testing

Medium. (2024). Solving common accessibility issues in Tailwind CSS for better UX. https://medium.com/@mohantaankit2002/solving-common-accessibility-issues-in-tailwind-css-for-better-ux-577bc84b9649

MENTOR. (2024). Virtual mentoring portals. https://www.mentoring.org/virtual-mentoring-portals/

Mentorloop. (2024). Self directed learning supercharged with mentoring. https://mentorloop.com/blog/self-directed-learning-supercharged/

Monterail. (2023). Software quality standards—How and why we applied ISO 25010. Monterail Blog. https://www.monterail.com/software-qa-standards-iso-25010

Murrell, A. J., & Blake-Beard, S. (2021). The importance of peer mentoring, identity work and holding environments: A study of African American leadership development. International Journal of Environmental Research and Public Health, 18(9), 4920. https://www.mdpi.com/1660-4601/18/9/4920

Murtonen, M., Anto, E., Laakkonen, E., & Vilppu, H. (2023). University teachers' focus on students: Examining the relationships between visual attention, conceptions of teaching and pedagogical training. Frontline Learning Research, 10(2), 64–85.

MyPrivateTutor.ph. (2024). The impact of digital learning on the Philippine education system: A comprehensive analysis. https://www.myprivatetutor.com.ph/blog/digital-learning-on-the-philippine-education-system

Nath, M. (2022). Fuzzy matching algorithms. Medium. https://medium.com/@m.nath/fuzzy-matching-algorithms-81914b1bc498

Nwaesei, A. S., & Liao, T. V. (2023). A programmatic approach to peer-led tutoring to assist students in academic difficulty. American Journal of Pharmaceutical Education, 87(3). https://www.sciencedirect.com/science/article/pii/S0002945923008148

Pacific Certifications. (2024). ISO/IEC 25010:2023 - Systems and software engineering: SQuaRE - Product quality model. https://pacificcert.com/iso-iec-25010-systems-and-software-engineering/

PagePro. (2024). Next js for e-learning platforms. https://pagepro.co/blog/next-js-for-e-learning-platforms/

Paolillo, A. (2024). Can peer tutoring improve academic performance in postgraduate education? Research Features, (153). https://researchfeatures.com/can-peer-tutoring-improve-academic-performance-postgraduate-education/

Parlakkiliç, A. (2022). Evaluating the effects of responsive design on the usability of academic websites in the pandemic. Education and Information Technologies, 27(1), 1307–1322. https://doi.org/10.1007/s10639-021-10650-9

PeopleFluent. (2024). Are you shifting to self-directed learning? Here's how to do it right. https://www.peoplefluent.com/blog/learning/shifting-to-self-directed-learning-supported-by-technology/

QMII. (2024). Understanding ISO/IEC 25010:2023 and SQuaRE for software quality assurance.

https://www.qmii.com/understanding-iso-iec-250102023-and-square-for-software-quality-assurance-2/

Rahman, L. (2024). Vygotsky's Zone of Proximal Development of Teaching and Learning in STEM Education. International Journal of Engineering Research & Technology, 13(8), 1847–1854.

Redvike. (2023). The best educational apps built with React.js. https://redvike.com/article/best-educational-apps/

ResearchGate. (2024). Significance of scaffolding and peer tutoring in the light of Vygotsky's theory of zone of proximal development. https://www.researchgate.net/publication/283624130_Significance_of_Scaffolding_and_Peer_Tutoring_in_the_Light_of_Vygotsky's_Theory_of_Zone_of_Proximal_Development

Reshetov, D. (2025). Whole-class, high-quality peer tutoring is achievable with minimal effort or expense for teachers [Preprint]. arXiv.

Rokeman, N. R. M. (2024). Likert measurement scale in education and social sciences: Explored and explained. EDUCATUM Journal of Social Sciences, 10(1), 77–88. https://doi.org/10.37134/ejoss.vol10.1.7.2024

Ruangkerja. (2024). Learning management system #1 Indonesia untuk perusahaan anda. https://www.ruangkerja.id/

ScienceDirect. (2025). A meta-analysis of the effect of peer tutoring in Science, Technology, Engineering and Mathematics (STEM) subjects. Heliyon, 11(2), e25123.

SEAMEO. (2023, November 30). Fast growth of digital technology is challenging education priorities and practices in Southeast Asia. World Education Blog. https://world-education-blog.org/2023/11/30/fast-growth-of-digital-technology-is-challenging-education-priorities-and-practices-in-southeast-asia/

SEAtS ONE. (2024). Data exchange | Edtech integrations and API. https://seatssoftware.com/education-technology/edtech-integrations-and-api/

Shih, S. C., Chang, C. C., Kuo, B. C., et al. (2023). Mathematics intelligent tutoring system for learning multiplication and division of fractions based on diagnostic teaching. Education and Information Technologies, 28, 9189–9210. https://doi.org/10.1007/s10639-022-11553-z

Simply Psychology. (2024). Zone of proximal development. https://www.simplypsychology.org/zone-of-proximal-development.html

SlashDev. (2024). How to build a custom education management system in NodeJS in 2024. https://slashdev.io/-how-to-build-a-custom-education-management-system-in-nodejs-in-2024

Stand Together Trust. (2024, January 25). Khan Academy and Stand Together Trust announce partnership to leverage AI to transform education [Press release].

Strapi. (2024). Scalable e-learning platforms with Strapi's API integrations. https://strapi.io/blog/scalable-e-learning-platforms-with-strapi-api-integrations

Supabase. (2024). The Postgres development platform. https://supabase.com/

Topping, K. J. (2005). Trends in peer learning. Educational Psychology, 25(6), 631–645.

TutorCruncher. (2024). Tutoring matching service: Find perfect matches today. https://tutorcruncher.com/blog/starting-tutor-business-tutor-matching

UNIKOM. (2024). UNIKOM learning management system. https://lms.unikom.ac.id/

US Patent Office. (2002). Interactive online learning with student-to-tutor matching. US Patent 20020013836A1. https://patents.google.com/patent/US20020013836A1/en

Userback. (2024). User acceptance testing explained: A guide to UAT best practices. Userback Blog. https://userback.io/blog/user-acceptance-testing-explained/

Ventionai. (2024). AI intelligent tutoring systems. https://ventionai.com/education/intelligent-tutoring-systems

Zeng, B., Jeon, M., & Wen, H. (2024). How does item wording affect participants' responses in Likert scale? Evidence from IRT analysis. Frontiers in Psychology, 15, 1304870. https://doi.org/10.3389/fpsyg.2024.1304870

Zou, Y., Kuek, F., Feng, W., & Cheng, X. (2025). Digital learning in the 21st century: Trends, challenges, and innovations in technology integration. Frontiers in Education, 10, 1562391. https://doi.org/10.3389/feduc.2025.1562391