

Executive Summary

The **PhD Student Manager** is a web-based lab management platform developed for the Robotic Vision Laboratory (RVL) at The University of Texas at Arlington. Research labs like RVL currently rely on disconnected tools: spreadsheets, emails, and shared drives - to manage student progress, paper submissions, equipment reservations, and travel reimbursements, creating inefficiencies and visibility gaps for both supervisors and students.

To address this, we designed and built a centralized, role-based web application using React, Next.js, and Supabase. The system supports three user roles - supervisors, PhD students, and administrators - each with tailored dashboards and workflows. Core features include weekly progress report submission, research paper tracking, lab equipment and server reservations, Kanban-based task management, and a digital travel reimbursement workflow.

The system follows a three-tier architecture (Presentation, Application, and Data layers), ensuring security, scalability, and maintainability through role-based access control and row-level security policies.

The result is a unified platform that reduces administrative workload, improves accountability, and enables the lab to focus more on high-quality research.

Background

- PhD research labs already manage a wide range of overlapping responsibilities - including tracking student progress, submitting weekly reports, managing paper submissions, coordinating equipment reservations, and handling travel reimbursements - yet many labs, including RVL at UTA still rely on disconnected tools such as spreadsheets, emails, and shared drives.
- This lack of a centralized system can lead to missed deadlines, scheduling conflicts, unclear accountability, and limited visibility into overall lab activity, increasing the administrative workload for both supervisors and students.
- Existing solutions like Laboratory Information Management Systems (LIMS) are typically designed for clinical or high-throughput environments and do not address the needs of academic research labs, such as student progress tracking, conference management, and structured reporting workflows.
- The Robotic Vision Laboratory (RVL) at UTA, led by Dr. Beksi, served as the primary stakeholder, with Hussain Alkatheri acting as sponsor and product owner to guide system requirements and development priorities.

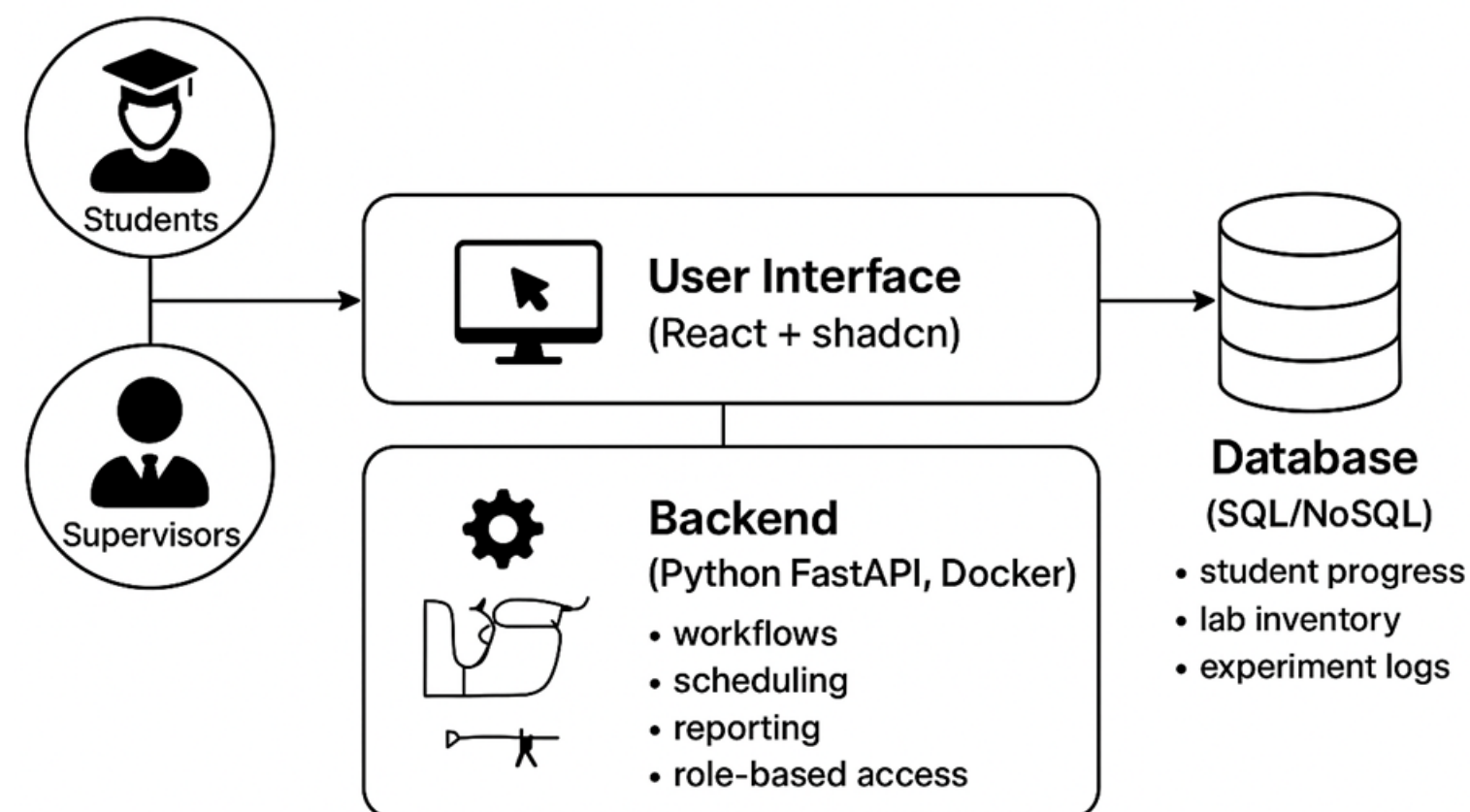


Figure 1: System overview showing the three core layers and primary user roles

Experimental Setup

System Architecture

- Three-tier architecture: Presentation, Application, and Data layers
- Separates UI, business logic, and data storage for scalability and maintainability
- Role-based access control enforced at every layer via Supabase RLS policies

Core Module Design

- Weekly progress report submission and supervisor review workflow
- Research paper tracking and conference submission pipeline
- Lab equipment and server reservation scheduling system
- Kanban-based task management board for project tracking
- Digital travel reimbursement request and approval workflow

Component	Technology	Purpose
Frontend	React / Next.js	Role-based UI & dashboards
Backend	Supabase	Auth, APIs, real-time updates
Database	PostgreSQL	Structured data & security
Security	RLS Policies	Row-level access per user role

Table 1: Technology stack and component roles

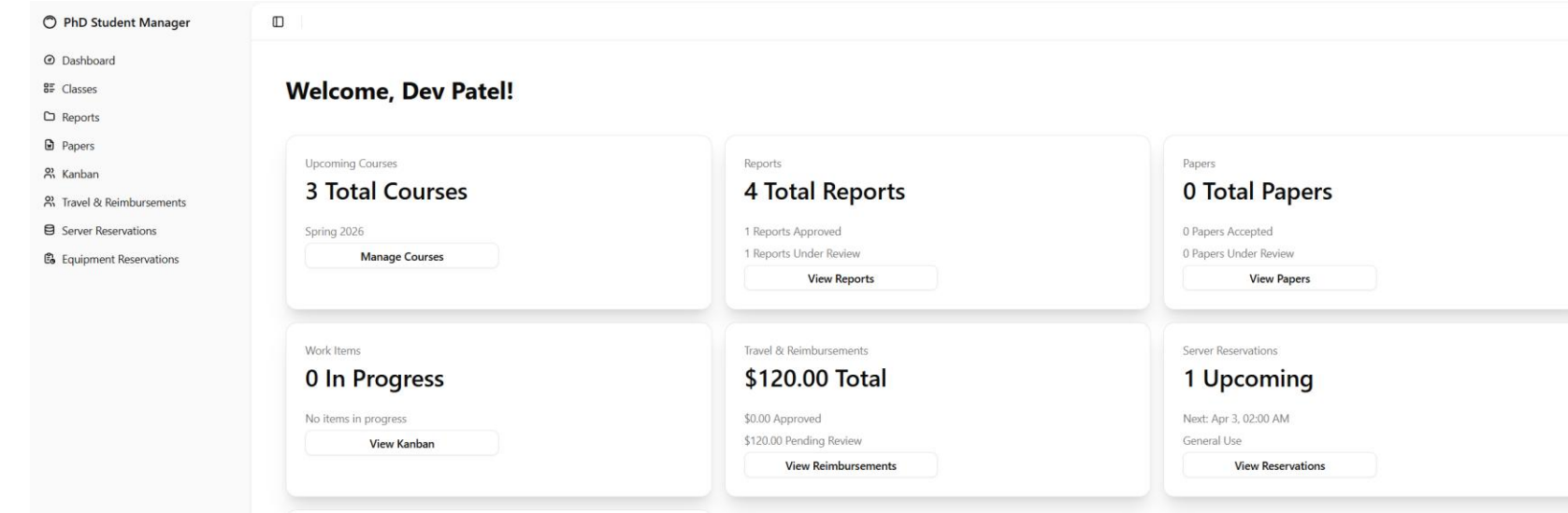


Figure 2: Student dashboard displaying centralized access to all lab management modules including reports, papers, reservations, and task tracking

Experimental Test Plan

Module Functional Testing

Weekly Reports Submit → Review → Approve pipeline validated end-to-end	Paper Tracking Status transitions: submitted → under review → accepted/rejected
Reservations Conflict detection for equipment & server booking tested	Kanban Board Task CRUD, drag-and-drop & status transitions verified
Travel Reimbursement Full request → documentation upload → supervisor approval → confirmation workflow validated	

Usability & Performance Testing

- System walkthrough conducted with sponsor Hussain Alkatheri to validate workflows against original requirements
- All three user roles simulated independently to evaluate dashboard clarity and navigation flow
- Performance tested under concurrent user interactions across multiple sessions
- Cross-browser compatibility verified on Chrome, Firefox, and Edge
- Report submission flow evaluated for intuitive UX and minimal steps to completion

Role-Based Access Testing

Supervisor All students & approvals	PhD Student Own records only	Admin System-wide management
---	--	--

- RLS policies validated: users restricted to authorized data across all modules
- Supervisors confirmed to view all students - students restricted to own records
- Admin panel tested for user creation, role assignment, and system oversight

System & Integration Testing

- End-to-end workflows tested across frontend and Supabase backend
- Real-time updates verified using Supabase live subscriptions
- Authentication and session handling tested for secure login and role assignment
- Data consistency validated across all 5 modules simultaneously

Report Submission Workflow Validation

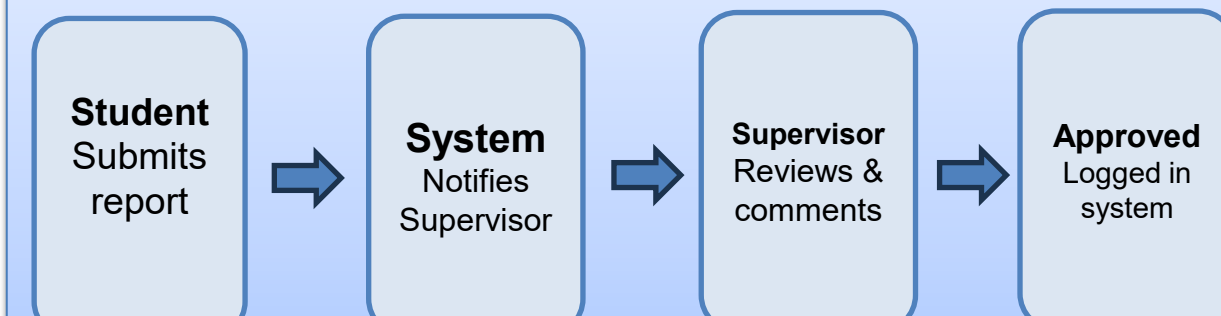


Figure 3: Validated report submission workflow across student and supervisor roles

Table 2: Test coverage by module

Module	Test Type	Roles	Status
Weekly Reports	Functional, workflow	Student, supervisor	✓
Paper Tracking	Functional, Status	Student, supervisor	✓
Reservations	Conflict Detection	Student, admin	✓
Kanban Board	Functional, UI	All roles	✓
Travel Reimb.	End-to-end	Student, supervisor	✓

Experimental Results

High-Fidelity Prototype — Live Deployed Web Application

The PhD Student Manager was developed and deployed as a fully functional, role-based web application at phd.e16a.com. All 5 core modules were completed and validated, serving as a high-fidelity simulation of a real lab management system for the RVL at UTA.

Module Prototype Results - All 5 Modules Functional

Complete Weekly Reports Students submit structured weekly progress reports; supervisors review, comment, and approve - full pipeline functional	Complete Paper Tracking Research paper submissions tracked across full status lifecycle: submitted → under review → accepted/rejected	Complete Reservations Equipment and server reservation scheduling with conflict detection fully operational across student and admin roles
Complete Kanban Board Task management board with drag-and-drop, status columns, and CRUD operations functional across all roles		Complete Travel Reimbursement Digital reimbursement workflow fully implemented - students submit requests with documentation, supervisors review and approve, system logs confirmed reimbursements

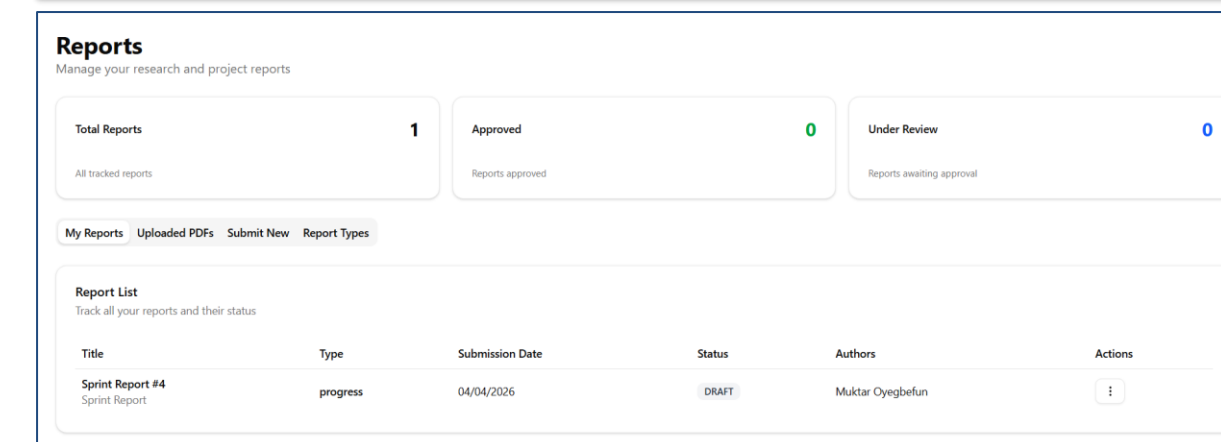


Figure 3: Weekly report submission interface displaying the student input form and supervisor review workflow

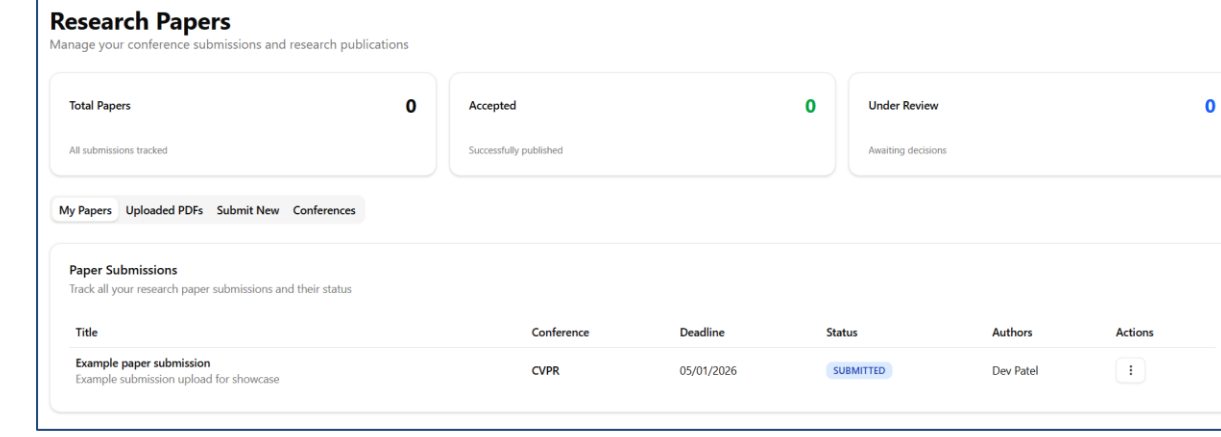


Figure 4: Paper tracking module showing research submission status pipeline from submitted through accepted or rejected

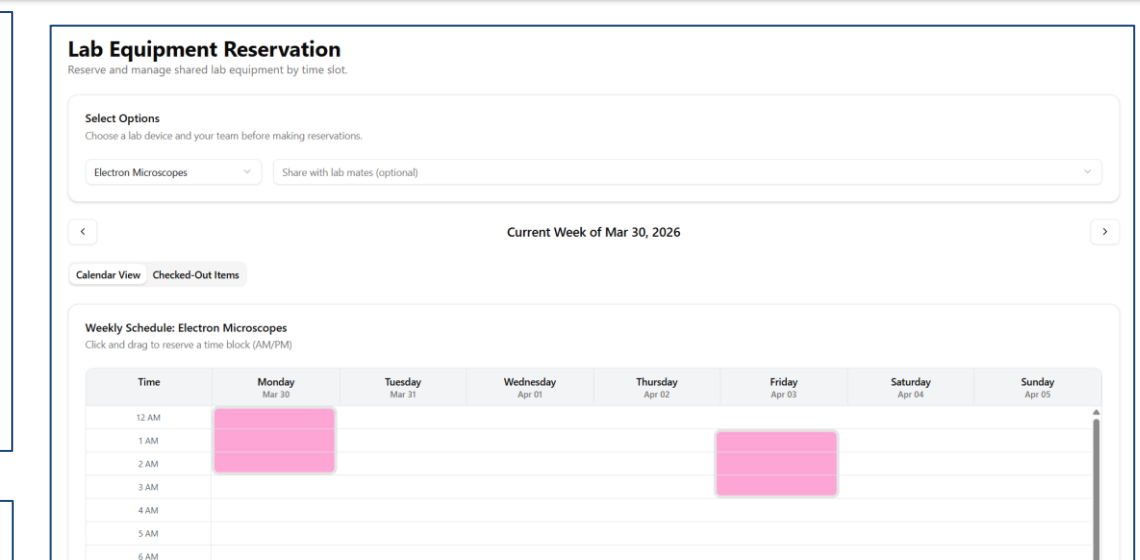
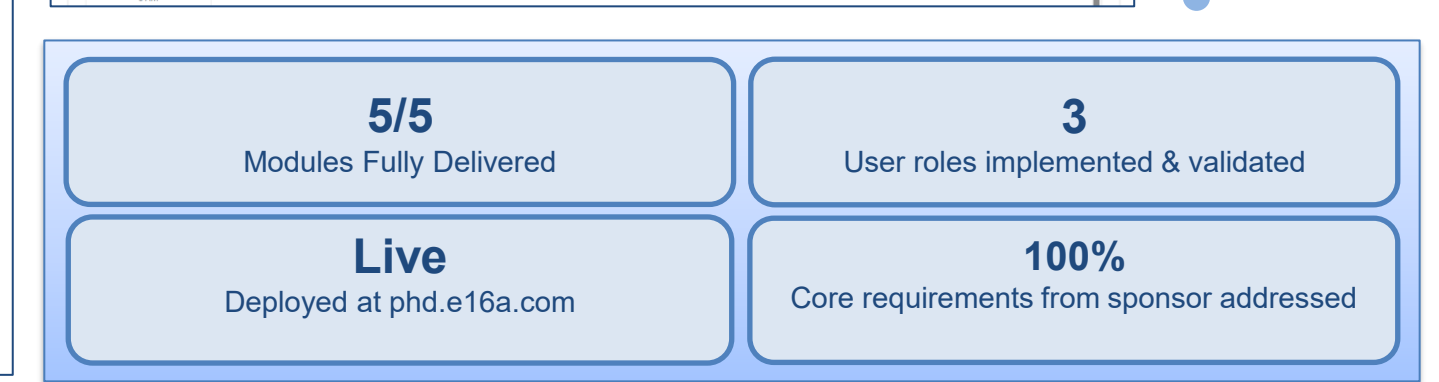


Figure 5: Equipment and server reservation scheduling interface with conflict detection across student and admin roles



Conclusions

Design Goals vs. Client Requirements

Centralized Platform All 5 lab workflows unified into one web-based system - directly replacing disconnected spreadsheets and emails	Role-Based Access Supervisor, PhD Student, and Admin roles fully implemented with secure RLS-enforced data boundaries	All Modules Delivered Reports, paper tracking, reservations, Kanban, and travel reimbursement - all fully functional and deployed
--	---	---

Key Insights & Lessons Learned

- Time Management**
Balancing this project alongside other coursework required stricter sprint planning - earlier deadline-setting per module would have reduced late-stage pressure
- Address Bugs Early**
Deferring bug fixes to later sprints increased integration complexity - resolving issues as they appeared would have kept the codebase cleaner
- Scope & Task Distribution**
Ensuring all team members shared an equal understanding of the project scope and evenly split workload improved consistency across modules

Constraints

- Team carried concurrent academic course load throughout the project timeline
- No dedicated QA environment - testing conducted within the same deployment
- Sponsor direction provided early in the semester; later iterations were self-directed
- Scope limited to RVL at UTA: system not yet generalized for multi-institution use

Acknowledgements

- Hussain Alkatheri: sponsor and product owner, provided initial requirements and project direction
- Robotic Vision Laboratory (RVL) at UTA - inspiration and reference use case for system requirements
- Dr. Shawn Gieser: course instructor for CSE 4316 and CSE 4317, provided guidance throughout the design and development process

Future Plans & Next Steps

Multi-Institution Expansion Generalize platform for PhD labs at universities nationwide beyond UTA	Mobile App Native iOS/Android app for on-the-go report submissions and notifications	Analytics Dashboard Lab-wide progress insights and reporting trends for supervisors	Email Notifications Automated alerts for deadlines, approvals, and reservation reminders
--	--	---	--

References

- GoalGetters Team, "Architectural Design Specification: PhD Student Manager," 2025, CSE 4316 Senior Design I, Fall 2025.
- GoalGetters Team, "System Requirements Specification: PhD Student Manager," 2025, CSE 4316 Senior Design I, Fall 2025.
- GoalGetters Team, "Detailed Design Specification: PhD Student Manager," 2026, CSE 4317 Senior Design II, Spring 2026.
- Meta Platforms Inc., "React Documentation," 2024. [Online]. Available: <https://react.dev>.
- Vercel Inc., "Next.js Documentation," 2024. [Online]. Available: <https://nextjs.org/docs>.
- Supabase Inc., "Supabase Documentation," 2024. [Online]. Available: <https://supabase.com/docs>.