LabConnect

Will B, Jaswanth D, Pragathi A, Aniket S, Doan N

Introduction

Research opportunities are an important part of RPI, helping students gain hands-on skills, explore academic interests, and pursue personal and career-oriented goals. LabConnect is a centralized, web-based platform designed to streamline how research positions are posted, discovered, and managed across campus. By providing a uniform process for listing lab openings and a user-friendly interface for browsing them, LabConnect makes it easier for students to find positions that match their interests.

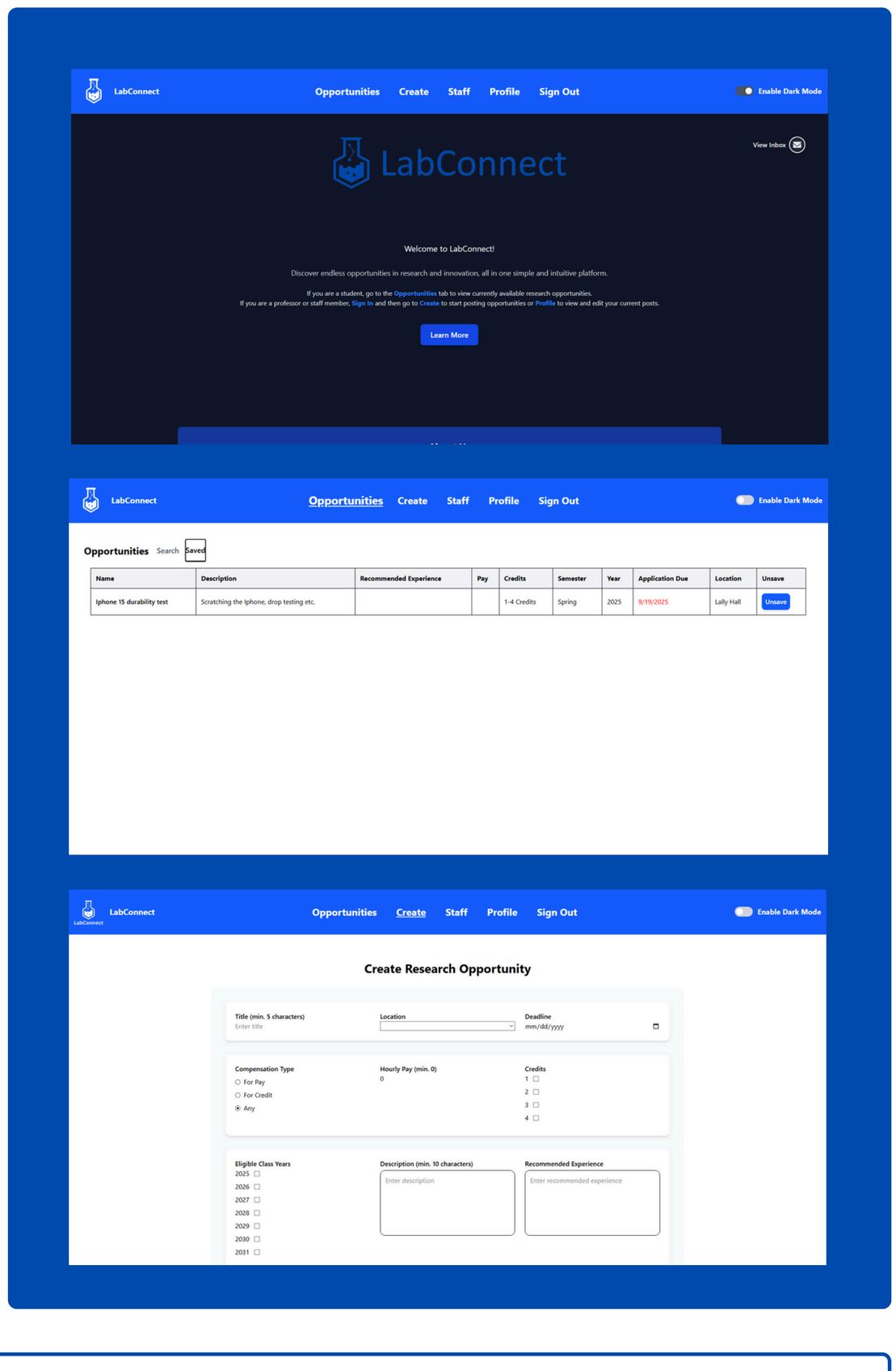
Objectives

- Saved opportunities page, routes and implementation
- Sitewide CSS rework
- Sitewide dark mode compatibility
- Backend test improvements
- User registration page and routes
- Professor registration implementation
- Individual opportunity page and routes
- Administration panel

Technology

Results

- **Style improvements:**
  - Sitewide rework of CSS for more unified styles
  - Sitewide dark mode support
- **Individual Post Display**
  - New page to view an individual opportunity
  - Route to get single opportunity data
  - Implementation into opportunities search and profiles
- **Saved Opportunities Functionality**
  - New subpage with opportunities search to show saved opportunities
  - Routes to save and unsave opportunities and to get all saved opportunities for a user
  - Implementation into opportunity search
- **User registration routes**
  - Built secure GET and PUT endpoints for users to manage their profiles, including personal details, majors and departments
- **Documentation**
  - Documentation for better project overview along with general updates



Conclusions

Our team has concluded that starting initially with good coding standards early on is essential for long-term maintainability and code quality. We learned firsthand that Test-Driven Development is far more effective than writing tests after the code is finished, as adding to existing test cases throughout development is much easier than recreating them later. We also recognized the impact of technical debt, especially in our frontend code, where unused or unnecessary components created avoidable confusion. This experience highlighted the importance of applying solid coding principles from the start, specifically in regards to abstraction and inheritance, in order to produce cleaner, more maintainable, and less bloated code.

Acknowledgements

Thank you to all of the faculty and mentors at RCOS. Additionally thank you to the previous project lead, Rafael Cenzano, for all of the help and effort he has put into the project