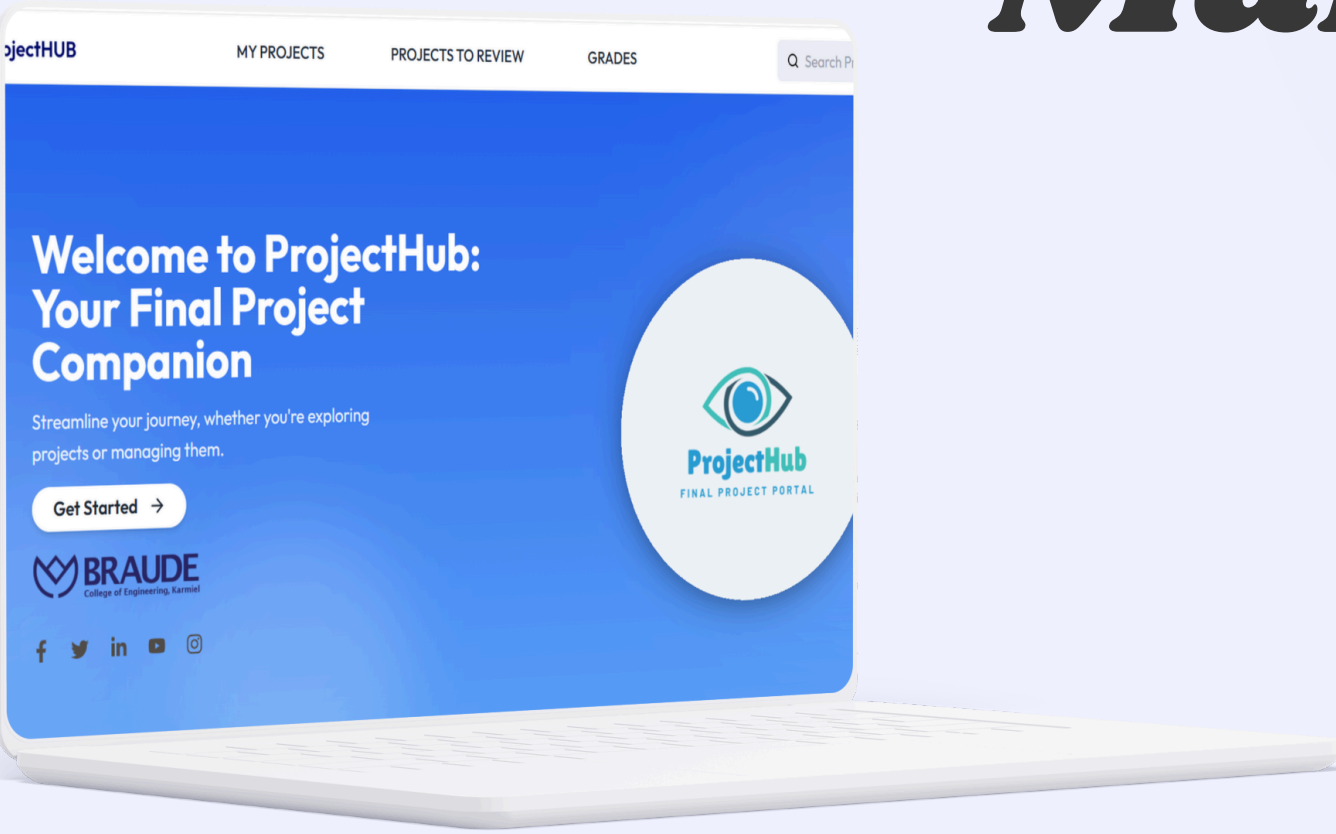# ProjectHub: Final Project Management System
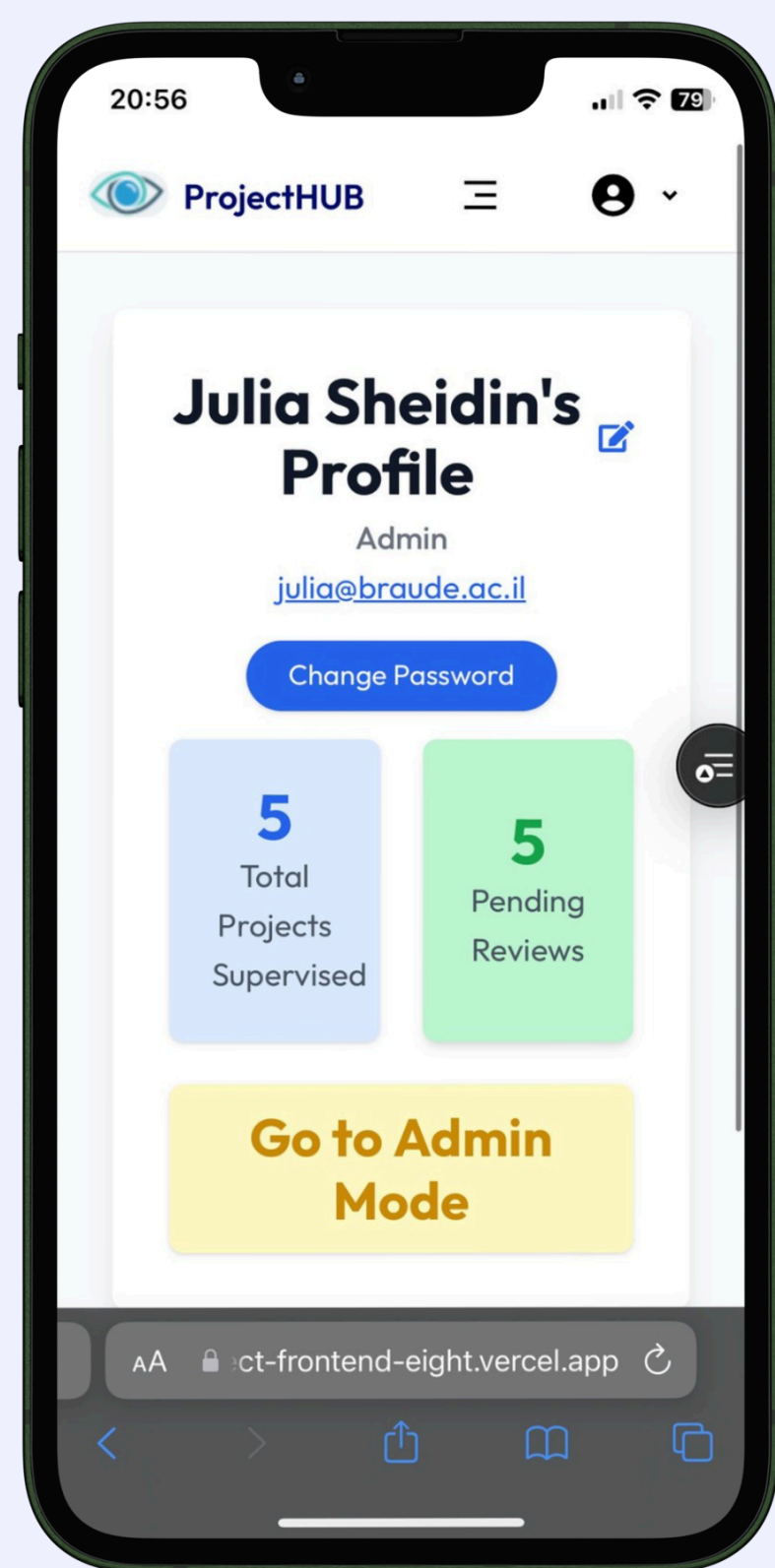
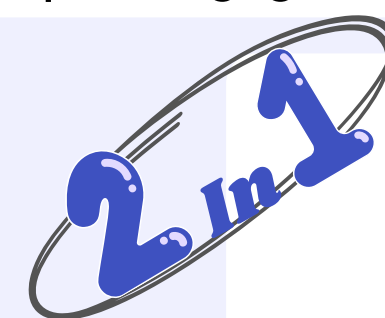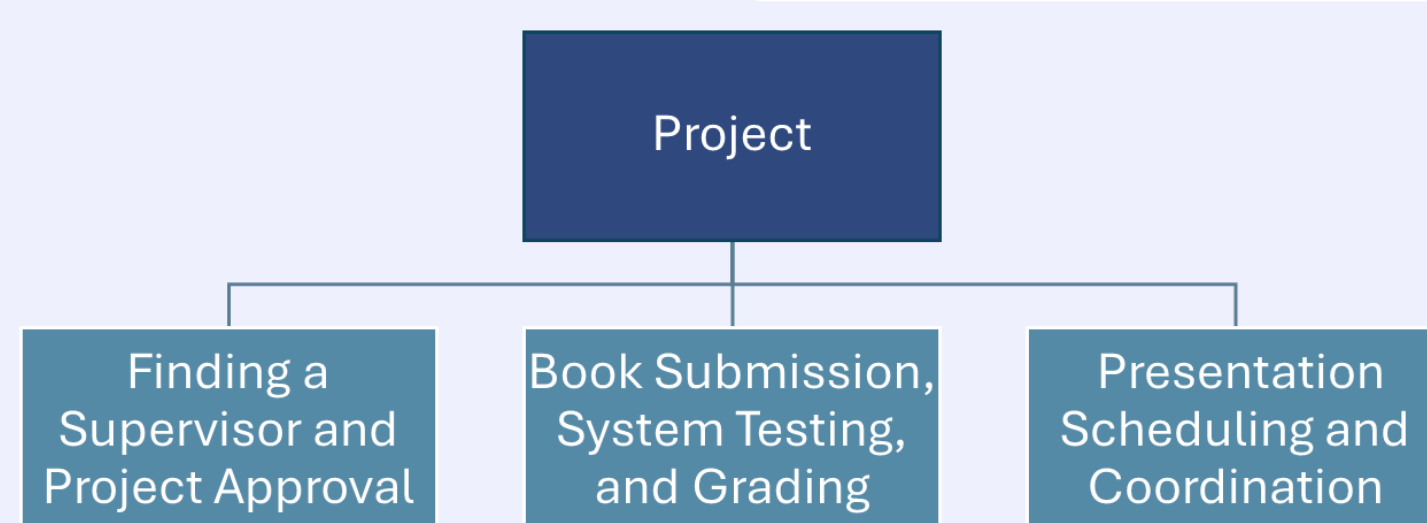Supervisor: Dr. Julia Sheidin
Students: Noy Malka, Noa Krispin

Introduction

ProjectHub is a web-based platform that simplifies academic final project management. It provides an automated and structured system for project tracking, grading, and evaluation, reducing administrative workload for faculty and supervisors.

System Requirements

- **Project Management** – Ability to upload, organize, and oversee student projects efficiently.
- **Grading & Evaluation** – A structured approach to submitting and tracking project grades.
- **Real-Time Updates** – Firebase-powered instant data synchronization.
- **Reminders & Notifications System** – Coordinators can send evaluation reminders to supervisors.
- **Role-Based Access** – Coordinators assign roles and manage access permissions.
- **Excel Import & Export** – Supports bulk uploads of project details and exporting grades for institutional use.

2 in 1

ProjectHub serves both supervisors and coordinators:
Supervisors manage projects, evaluate student work, and submit grades.
Project Coordinators (**Admins**) oversee project workflows, assign roles, and track grading progress.

Project
- Finding a Supervisor and Project Approval
- Book Submission, System Testing, and Grading
- Presentation Scheduling and Coordination

Challenges & logic

Evaluator Assignment Limitations – The system is designed exclusively for supervisors and project coordinators. To address this limitation in Part 3 , we allowed project coordinators to upload an Excel file listing evaluators and their assigned projects.

Starting with Part 2 Instead of Part 1 – Since Part 1 (project approval process) was not implemented, project data had to be imported through Excel uploads to fill the missing gaps.

Project Metrics & Performance Evaluation

User Experience & Accessibility – Ensured a simple and intuitive interface for supervisors.

Performance Optimization – Fast data retrieval using Firebase's real-time synchronization.

Scalability – The cloud-based infrastructure supports future enhancements.

Security Measures – Implemented strict authentication rules to protect sensitive project data.

Client Layer
Built with React and Tailwind CSS for styling, handling the user interface and API communication.

Server Layer
Developed using Node.js and Express.js, managing business logic, authentication, and database queries.

Database Layer
Uses Firebase Firestore, a NoSQL database for storing users, projects, and evaluation data.