

**Capstone Project Phase B**

**Project Hub**

**Project code 24-2-D-10**

**Final Projects Portal: Topics, Partners, Supervisors and more**



Supervisor:

Dr. Julia Sheidin

Students:

Noy Malka- Noy.malka@e.braude.ac.il

Noa Krispin- Noa.Krispin@e.braude.ac.il

[Git Repository](#)

<https://final-project-frontend-eight.vercel.app/>

# Contents

Abstract.....	4
<b>1</b> <b>Introduction .....</b>	<b>5</b>
<b>2</b> <b>General Description .....</b>	<b>6</b>
<b>2.1</b> <b>Capstone Project Process.....</b>	<b>6</b>
<b>2.2</b> <b>Key Stakeholders .....</b>	<b>7</b>
<b>3</b> <b>Materials and Methods .....</b>	<b>7</b>
<b>3.1</b> <b>ProjectHub Application .....</b>	<b>7</b>
<b>3.2</b> <b>Development Process and Tools .....</b>	<b>9</b>
<b>3.2.1</b> <b>Stakeholder Interviews and Insights .....</b>	<b>10</b>
<b>3.2.2</b> <b>Incremental Planning and Feature Development.....</b>	<b>11</b>
<b>3.2.3</b> <b>Tools and Technologies .....</b>	<b>12</b>
<b>3.2.4</b> <b>Timeline .....</b>	<b>15</b>
<b>4</b> <b>System Diagrams.....</b>	<b>16</b>
<b>4.1</b> <b>Use Case .....</b>	<b>16</b>
<b>4.2</b> <b>Package Diagram.....</b>	<b>17</b>
<b>5</b> <b>Prototype .....</b>	<b>17</b>
<b>5.1</b> <b>Initial Design Concepts .....</b>	<b>17</b>
<b>5.1.1</b> <b>Design Process .....</b>	<b>17</b>
<b>5.1.2</b> <b>Development with Simulated Data .....</b>	<b>21</b>
<b>5.2</b> <b>Presenting the Prototype.....</b>	<b>21</b>
<b>5.3</b> <b>Stakeholder Feedback and Iterations .....</b>	<b>23</b>
<b>5.4</b> <b>Final Prototype and Features .....</b>	<b>23</b>
<b>5.5</b> <b>Testing Plan.....</b>	<b>24</b>
<b>6</b> <b>Challenges - Description and Solutions.....</b>	<b>26</b>
<b>7</b> <b>Conclusions .....</b>	<b>27</b>
<b>8</b> <b>User Documentation.....</b>	<b>28</b>
<b>8.1</b> <b>User Guide.....</b>	<b>28</b>
<b>8.1.1</b> <b>Introduction.....</b>	<b>28</b>
<b>8.1.2</b> <b>Getting Started.....</b>	<b>28</b>

8.1.3	Using the Website .....	31
8.1.4	Troubleshooting.....	43
8.2	Maintenance Guide .....	45
8.2.1	Overview .....	45
8.2.2	Technical Requirements.....	45
8.2.3	System Administration.....	52
8.2.4	Debugging and Logs.....	54
8.2.5	Future Maintenance.....	56
References.....		58

## **Abstract**

In academic settings, final projects are essential for students to demonstrate the skills and knowledge gained during their degree. However, students and lecturers often face challenges in accessing and organizing relevant project information. Students struggle to find available topics, identify potential partners, and select suitable supervisors, while lecturers have difficulty presenting their project topics, managing student inquiries, and overseeing collaborations. Without a centralized and efficient system, these tasks become disorganized, resulting in delays, lower project quality, and frustration for both students and faculty.

To address these issues, a centralized website was developed specifically for managing final projects. The platform offers a user-friendly interface that enables lecturers and project coordinators to streamline grading workflows, manage submission tracking, and provide structured feedback efficiently. Focusing on the grading and evaluation phase, the system includes tools for managing evaluation forms, tracking project progress, and automating grade calculations. These features significantly reduce administrative burdens for project coordinators and lecturers while improving communication with students.

The project involved designing and building the website, integrating advanced search functionality to enhance the user experience, and incorporating features that supported the needs of project coordinators and lecturers. Developed through a collaborative process that incorporated stakeholder feedback, this platform significantly improves the organization, efficiency, and quality of final project processes. Given the scope of the project and time constraints, development efforts were focused exclusively on the grading and evaluation phase to ensure the delivery of high-quality, impactful features.

### **Keywords:**

Task management, Project management, User-Centered Design, Knowledge management, Web application development.

# 1 Introduction

The final project is a critical component of the undergraduate curriculum, enabling students to apply theoretical knowledge to practical challenges. However, managing these projects effectively has long been a challenge due to the absence of a centralized system. Manual processes, such as email communication and the use of Excel spreadsheets, have created inefficiencies and significant obstacles for students, supervisors, and administrative staff.

Students often struggle to access project information, identify potential partners, and select supervisors, while faculty members face difficulties managing inquiries, coordinating schedules, and providing timely feedback. Administrative staff, burdened with manual data entry and error-prone workflows, experience increased workloads that divert resources from essential duties. These fragmented processes result in delays, disorganization, and frustration for all stakeholders, ultimately compromising the quality of the final project experience.

To address these challenges, we began developing **ProjectHub**, a centralized web-based platform designed to streamline the final project management process. Due to project scope and time constraints, development efforts in this phase focused exclusively on the grading and evaluation component of the system. This part of the platform supports supervisors and project coordinators by providing tools for grading workflows, submission tracking, and structured feedback management. These features address critical inefficiencies in managing evaluations and ensure data accuracy while significantly reducing administrative burdens.

Built with modern technologies such as React, Node.js, and Firebase, ProjectHub incorporates advanced features like real-time screens and customizable evaluation forms. The system facilitates effective collaboration between supervisors, coordinators, and students by automating repetitive tasks, enhancing communication, and centralizing grading processes.

By focusing on the grading and evaluation phase, ProjectHub addresses the most pressing pain points identified during stakeholder interviews. While the platform is ultimately intended to support the entire lifecycle of final projects, this phase prioritizes the needs of supervisors and coordinators, offering a practical and impactful solution to improve efficiency, streamline communication, and elevate the overall quality of final projects.

## 2 General Description

### 2.1 Capstone Project Process

Our college's capstone project is divided into two phases: **Phase A** and **Phase B**, both critical for successful project completion.

In Phase A, students select a project supervisor and either propose their own idea or choose from a list of supervisor-provided topics. During this phase, students conduct research, interviews, and data collection, documenting their work in a comprehensive project book. This is the basis for a formal presentation to a panel, including their supervisor and other faculty members. The evaluation for Phase A consists of assessing the project book and presentation, with feedback provided for revisions before advancing to Phase B.

Phase B focuses on developing and testing the system or solution conceptualized in Phase A. Students refine their work by implementing the system, testing its functionality, and addressing any issues. Upon completion, the final system is submitted along with a formal presentation, which is evaluated by the supervisor and Panel Members. Grading in this phase considers the quality of the system, the updated project book, and the presentation.

The **ProjectHub** platform was designed to address inefficiencies throughout the entire lifecycle of final projects, from topic selection to evaluation and grading. By centralizing project management, it aims to improve accessibility, foster collaboration, and reduce administrative burdens for all stakeholders.

In Phase B, development efforts focused exclusively on the grading and evaluation phase. This stage was identified as a critical yet often overlooked area, plagued by inefficiencies in manual workflows, such as grading on paper forms, managing deadlines with spreadsheets, and the lack of real-time tracking tools. These challenges not only created delays but also increased the likelihood of errors and administrative burdens. To address these issues, ProjectHub introduces tools to automate repetitive tasks, track submission statuses, and improve the accuracy and efficiency of evaluations.

The capstone project process involves multiple stakeholders, each playing a vital role in ensuring smooth progression and successful completion of projects. While ProjectHub is ultimately designed to support all stakeholders throughout the lifecycle of final projects, the Phase B development specifically prioritized the needs of supervisors, project coordinators, and panel members involved in the grading and evaluation phase. The roles and contributions of these stakeholders are detailed below (see Table 1) to clarify their importance and provide context for the functionalities discussed later in the document.

## 2.2 Key Stakeholders

Stakeholder	Role	Key User Role in ProjectHub
Students	Students are responsible for selecting a research or project topic, conducting research, and developing a solution. They use the system to manage various aspects of their projects, ensuring successful completion of their academic requirements.	No – Students are not direct users of the grading-focused features developed in Phase B.
Supervisors	Faculty members who mentor, review proposals, and assess project books and presentations of students.	Yes – Supervisors are primary users, utilizing grading and evaluation tools for streamlined workflows.
Final Project Coordinators and Administrative Staff	Primary users oversee the entire project process, manage registration, approve proposals, coordinate scheduling, and handle administrative tasks.	Yes – Coordinators are key users, leveraging tools for tracking submissions, monitoring deadlines, and managing grading processes.
Panel Members	Faculty members evaluate student presentations and provide feedback during the final assessment.	Yes - Panel members who are supervisors use the system to fill out evaluation forms, assign grades, and review submissions.

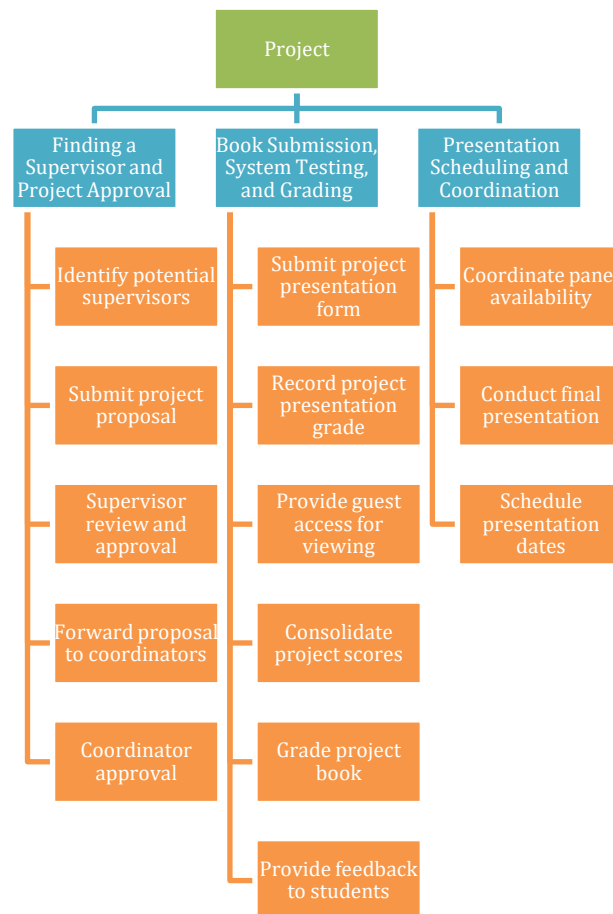
**Table 1:** Key stakeholders and their roles

## 3 Materials and Methods

### 3.1 ProjectHub Application

The purpose of the ProjectHub platform is to streamline and centralize the project cornerstone process, address inefficiencies, and improve collaboration across all phases of project management. The system is structured into three main parts, as shown in the Work Breakdown Structure (WBS) (Figure 1)[1], each targeting a critical phase of the capstone project lifecycle:

1. **Part 1: Project Selection and Initiation** – Facilitating the selection of topics, assignment of supervisors, and project planning.
2. **Part 2: Grading and Evaluation** – Supporting the submission of project books, grading, feedback management, and performance tracking.
3. **Part 3: Final Review and Reporting** – Enabling final project archiving, review of outcomes, and comprehensive reporting.



**Figure 1: Work Breakdown Structure of the Project**

While the platform’s overarching goal is to address all three stages, the development in Phase B focused exclusively on the second stage—Grading and Evaluation—due to time constraints. This phase aims to address specific bottlenecks identified during interviews and problem analysis. Traditionally, grading workflows relied heavily on fragmented tools such as Google Docs, Excel spreadsheets, and email communications. These tools often led to delays in grade submission, errors in data management, and communication gaps. For example, missing grades or feedback were frequently identified only during final grade compilation, creating additional workloads for coordinators. Additionally, students lacked a structured way to access feedback, hindering their ability to reflect on and improve their work.



To address these challenges, the development efforts in Phase B focused on delivering the following key functionalities:

- **A Centralized System for Submitting Evaluation Forms and Grades**  
Supervisors and panel members can submit grades directly through the platform. The system includes automated weighting of grades based on predefined criteria, eliminating the need for manual calculations by project coordinators. This feature reduces human error and significantly improves administrative efficiency. Additionally, coordinators can manage grading deadlines and ensure no assessments are overlooked (Figure 18).
- **A screen for Tracking Submissions**  
Coordinators have access to a real-time submission tracking screen (Figure 24), which provides updates on the status of submitted evaluation forms and pending assessments. This screen allows proactive management of grading deadlines and minimizes delays by highlighting missing grades or incomplete evaluations.
- **Evaluation Form Creation and Editing Tools**  
Administrators can create new evaluation forms or edit existing ones to ensure they align with current requirements. For example, questions and their respective weights can be adjusted as needed, providing flexibility in adapting to different evaluation criteria. These tools are integral to maintaining consistency and adaptability across grading processes (Figure 22) (Figure 23).

Implementing these features, ProjectHub significantly reduces administrative burdens, minimizes delays, and ensures that grading and evaluation processes are conducted more accurately and efficiently. The inclusivity of the platform, designed to accommodate users with varying technical proficiencies, ensures its accessibility and ease of adoption.

### 3.2 Development Process and Tools

To achieve these goals and meet the project requirements, we adopted a User-Centered Design (UCD) model [2]. This approach prioritized understanding user needs, incorporating feedback into iterative refinements, and ensuring the system was both user-friendly and functional for key stakeholders, including supervisors and project coordinators. By placing end users at the center of the development process, we ensured the platform effectively addressed their most critical pain points.

The UCD model aligned seamlessly with the WBS used during the project planning phase, which divided the system into three distinct parts: Project Selection and Initiation, Grading and Evaluation, and Final Review and Reporting. This structured framework guided our focus on Part 2: Grading and Evaluation, identified as the most impactful phase through stakeholder

consultations. It provided a clear roadmap for incremental development, enabling us to prioritize high-value features while remaining flexible to adapt based on stakeholder feedback. By concentrating on Part 2, we directed our efforts toward delivering the system's most critical components within the available timeframe.

### 3.2.1 Stakeholder Interviews and Insights

To ensure the ProjectHub platform effectively addressed user needs, the development process began with in-depth interviews involving key stakeholders, including supervisors, project coordinators, and administrative staff. These discussions were instrumental in identifying challenges within existing workflows and prioritizing features for development in the grading and evaluation phase. By placing end users at the center of the process, we ensured the platform was designed to meet their specific requirements.

The interviews revealed several key challenges and corresponding insights:

- **Centralized Grading and Submission Tools:**
  - **Challenge:** Supervisors struggled with fragmented tools for submitting grades and providing feedback, often leading to delays and inconsistencies.
  - **Insight:** A centralized system with automated reminders and weighted grading calculations would streamline grading tasks and reduce errors.
- **Tracking and Monitoring:**
  - **Challenge:** Project coordinators found it difficult to track submission statuses and follow up with late graders.
  - **Insight:** A real-time screen to monitor grading progress and submission statuses was essential for efficient management.
- **Flexible Evaluation Forms:**
  - **Challenge:** The grading process relied on static forms that lacked the flexibility to adapt to varying project requirements.
  - **Insight:** Administrators required tools to easily create or modify evaluation forms to fit specific project needs.

Weekly meetings with our supervisor, who also served as a project coordinator, established a continuous feedback loop that guided the refinement of system components. Early prototypes, such as the grading interface, were reviewed and improved based on suggestions. For instance, clearer deadlines and notifications were added to help users meet their responsibilities. Similarly, the submission tracking screen was enhanced to include real-time updates and alerts on missing evaluations, addressing coordinators' needs for effective monitoring and supervision. Additional feedback from other supervisors and project coordinators during consultations further refined these features, ensuring alignment with stakeholder priorities.

This feedback-driven, user-focused process exemplifies UCD by prioritizing usability, aligning features with user requirements, and refining functionality based on direct input from stakeholders. We created a platform that effectively resolved critical inefficiencies by incorporating these insights into the development workflow. The centralized grading tools, real-time tracking screen, and flexible evaluation form editor significantly enhanced the grading and evaluation process, improving user efficiency and satisfaction.

### **3.2.2 Incremental Planning and Feature Development**

The prioritization of the Grading and Evaluation phase led to the definition of clear functional and non-functional requirements for the ProjectHub platform. These requirements addressed inefficiencies in manual grading workflows, improved tracking capabilities, and enhanced user experience. Key functional requirements included features like centralized grade submission, a real-time tracking screen, and customizable evaluation forms. Non-functional requirements focused on scalability, usability, and compatibility to ensure a seamless and reliable system.

#### **Functional Requirements**

- The system allows supervisors and panel members to submit student grades via a centralized interface.
- The system shall automatically calculate, and weight grades based on predefined criteria.
- The system shall provide project coordinators with a real-time screen to monitor grading progress and submission statuses.
- The system shall enable project coordinators to send automated reminders to supervisors and panel members for missing grades or pending tasks.
- The system shall allow administrators to create, edit, and customize evaluation forms for different projects.
- The system shall support uploading Excel files containing project data, ensuring consistency in the database.
- The system shall notify users about upcoming deadlines and incomplete tasks.
- The system should allow coordinators to export Excel files.

#### **Non-Functional Requirements**

- The system ensures high usability, allowing users with varying technical skills to navigate and complete tasks efficiently.
- The system should maintain scalability, supporting multiple concurrent users without performance degradation.

- The system should provide real-time updates for screens and notifications, ensuring data is accurate and current.
- The system shall ensure data integrity and security with user authentication and role-based access control.
- The system shall support compatibility with major web browsers and devices, ensuring seamless user experience.
- The system will achieve an average response time of under 2 seconds for all major actions.
- The system shall allow for easy maintainability, enabling updates to evaluation forms and workflows without disrupting ongoing processes.

After defining these requirements, the development process focused on gradually building and improving individual components of the system. Specific pages or features were developed individually, with feedback incorporated after each step. For example, the grading interface was initially developed with basic functionality for submitting grades. After receiving feedback from supervisors, additional improvements were implemented, including features such as the ability to write personal and project-specific notes and tools to send email notifications to students. Similarly, the submission tracking screen was gradually enhanced to include real-time updates and automated alerts for missing assessments, ensuring supervisors had the tools to efficiently track progress, meet grading deadlines, and maintain oversight of project evaluations.

This feedback-driven, user-centered approach aligns with UCD principles, ensuring functionality directly addresses stakeholder requirements while maintaining usability. By focusing on the most critical needs and refining features based on user input, we developed a platform that significantly improved grading workflows, tracking efficiency, and overall user satisfaction.

### 3.2.3 Tools and Technologies

The development of the ProjectHub platform relied on a carefully selected set of tools and technologies that supported user-focused development, effective collaboration, and the refinement of features based on stakeholder feedback. Each tool was chosen to align with the platform's modular architecture and to facilitate usability, real-time updates, and user-centered workflows.

#### Development Tools

##### Frontend Development:

- **React:** A JavaScript library used for building dynamic, component-based user interfaces. React enabled the creation of reusable components, improving development efficiency and consistency across the platform.

#### Backend Development:

- **Node.js:** A scalable, high-performance runtime environment for building server-side applications. Node.js facilitated the integration of APIs and real-time data processing.
- **Express.js:** A lightweight web application framework used alongside Node.js to simplify routing and backend logic.

#### Database:

- **Firestore:** Selecting the right database for the ProjectHub platform was critical to ensuring scalability, real-time functionality, and ease of deployment. Initially, we began with MySQL, a relational database. While MySQL supports robust data structuring and remote access when properly configured, our initial setup relied on a local server. This created challenges for deployment, as multiple users required consistent access to the database over a network, highlighting the need for a cloud-based solution.

We then transitioned to **MongoDB**, a NoSQL database, which offered greater flexibility in handling unstructured data. However, during testing, we identified limitations in its ability to handle real-time synchronization natively. Although MongoDB supports real-time capabilities through additional configurations (e.g., Change Streams), implementing this functionality added complexity that did not align with our project timeline and requirements.

Ultimately, we adopted **Firestore**, a cloud-hosted NoSQL database, due to its ability to meet the platform's real-time and scalability requirements. Firestore provided:

- **Real-Time Synchronization:** Updates to grading, tracking, and feedback data were instantly reflected for all users without additional middleware.
- **Ease of Deployment:** Firestore's cloud-based infrastructure simplified deployment, eliminating the need for server configuration or maintenance.
- **Seamless Integration:** Its compatibility with the existing architecture ensured smooth adoption and accelerated development.

Despite initial challenges with database selection, Firestore allowed us to deliver the dynamic, real-time functionality necessary for the grading and evaluation workflows central to the **ProjectHub** platform.

#### Collaboration Tools

##### Version Control:

- **Git:** Used for version control, enabling efficient tracking of changes and collaboration among team members.

GitHub: The project repository was hosted on GitHub, providing a centralized platform for issue tracking, pull requests, and team discussions.

## Prototyping and Design

- **Figma:** Figma played a critical role during the early stages of development, enabling the creation of interactive mockups that stakeholders could visualize and provide feedback on. These prototypes facilitated the refinement of user interface designs, ensuring they aligned with stakeholder expectations and usability standards.

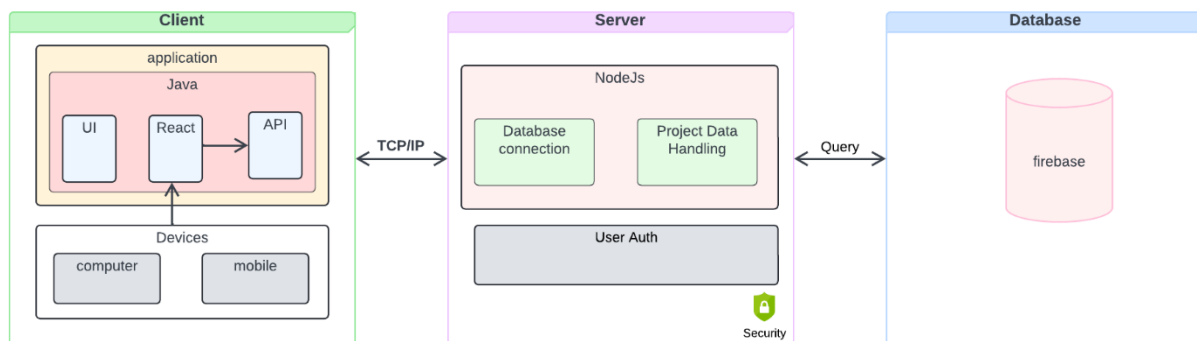
## Development Environment

- **Visual Studio Code (VS Code):** Our preferred code editor for development, offering an extensive range of extensions to enhance productivity and debugging.

## System Architecture

The ProjectHub platform follows a three-tier architecture (Figure 2), dividing responsibilities between the Client (Frontend), Server (Backend), and Database (Firestore). Each layer is responsible for different functions:

- **Client Layer:** Built with React, 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.



**Figure 2:** System Architecture of ProjectHub

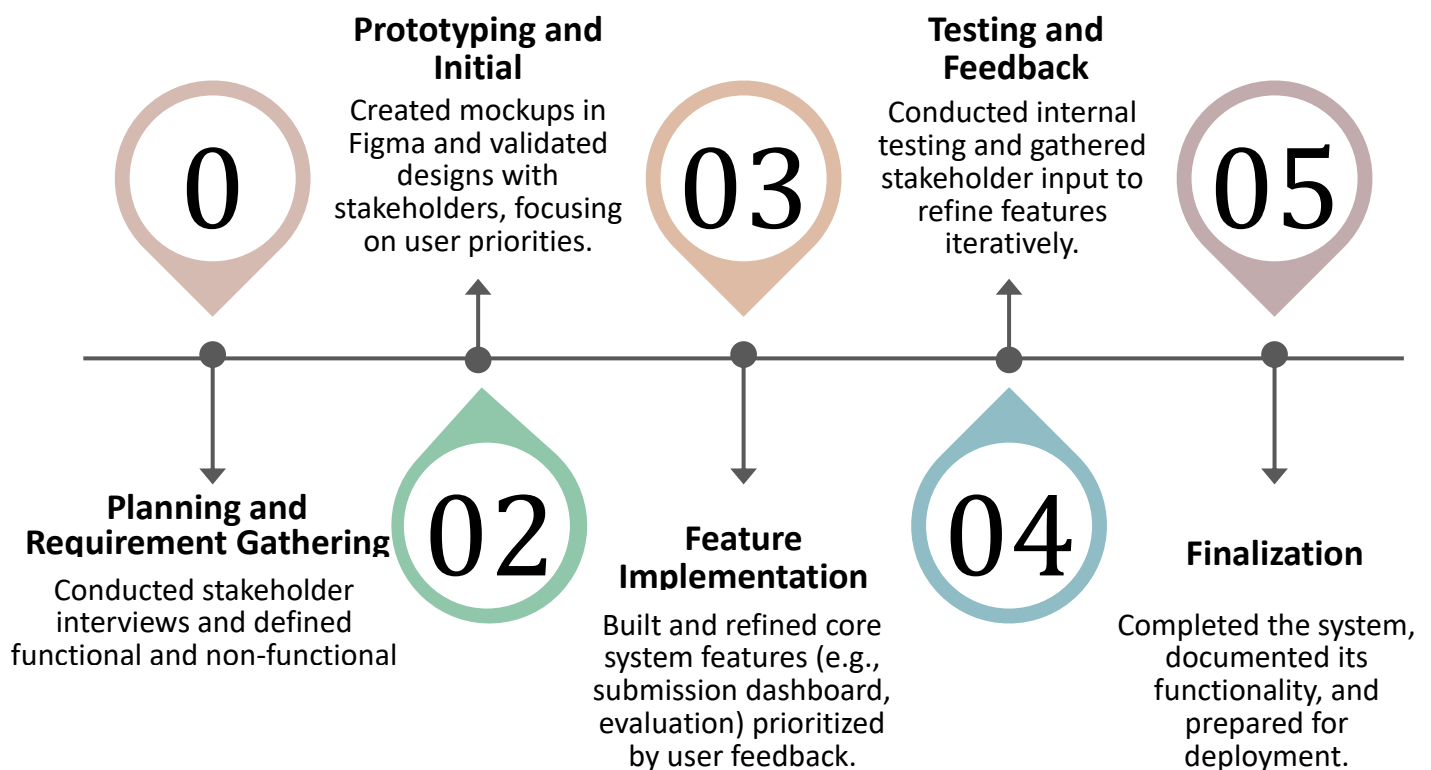
These tools collectively supported the User-Centered Design (UCD) process by enabling user-focused development and stakeholder collaboration. Tools like React and Firebase ensured that features were modular, usable, and capable of delivering real-time updates, while GitHub facilitated efficient teamwork. Figma bridged the gap between stakeholders and developers, allowing design decisions to reflect user expectations from the outset. Together, these technologies enabled the delivery of a platform that met user needs effectively while remaining adaptable to future refinements.

### 3.2.4 Timeline

The timeline for the development of ProjectHub provides an overview of the key activities and milestones achieved during the project. Adopting a User-Centered Design (UCD) approach, the development process prioritized understanding user needs, integrating stakeholder feedback, and refining features to address the most critical challenges supervisors and project coordinators face.

The timeline is divided into distinct phases, each with clear objectives to ensure progress remains aligned with the project's goals and deadlines (Figure 3). Early stages focused on gathering requirements and designing prototypes, while subsequent phases centered on feature development, testing, and feedback-driven refinements. This structured progression ensured that every process step was guided by user priorities, culminating in a platform that effectively addresses key pain points in grading and evaluation workflows.

The timeline below highlights these phases, illustrating how stakeholder input and iterative refinements shaped the platform's development.



**Figure 3:** Use Case Diagram for ProjectHub

## 4 System Diagrams

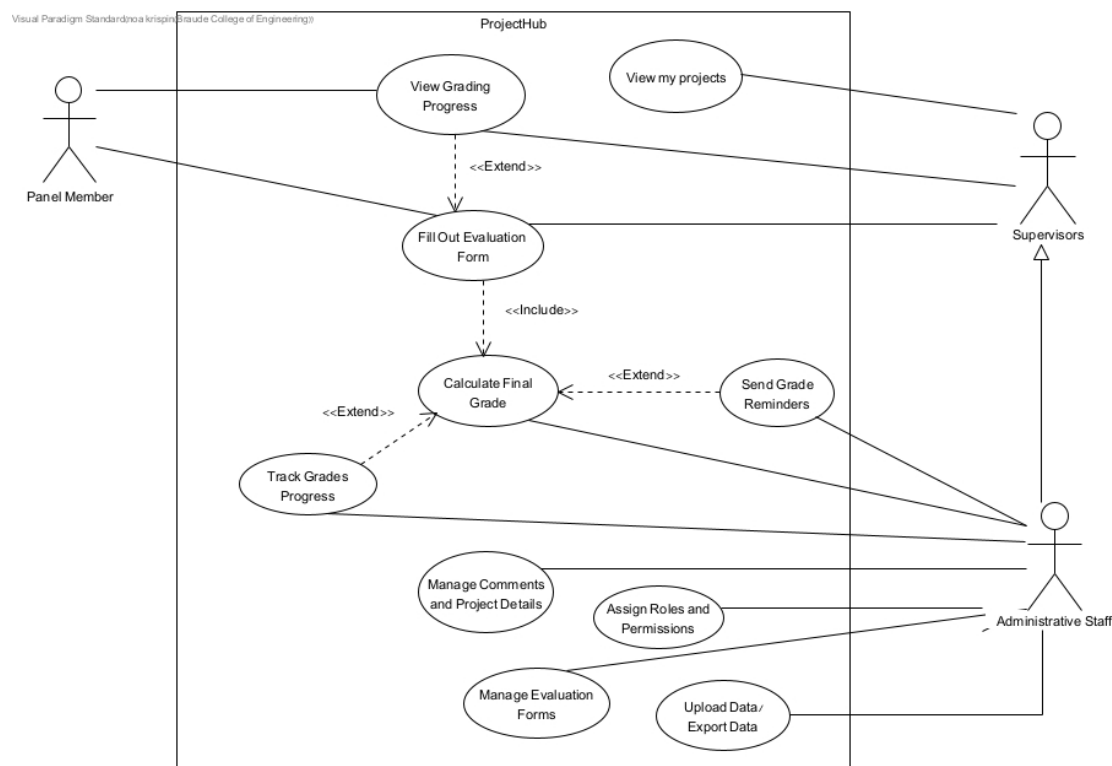
### 4.1 Use Case

The use case diagram below represents the primary interactions and functionalities of the ProjectHub system. The diagram illustrates how various users—Panel Members, Supervisors, and Administrative Staff—interact with the system to manage and assess projects effectively.

**The Panel Members** are responsible for reviewing project books and presentations, completing evaluation forms, and tracking their grading progress.

**Supervisors** can monitor grading progress, review their assigned projects, and provide feedback. They also assist in the grading process and ensure that evaluation forms are completed.

**Administrative Staff** is tasked with administrative operations such as assigning roles and permissions, managing evaluation forms, and exporting project-related data. They can also send grade reminders to ensure timely evaluations and monitor system-wide grading progress. The use case focuses on enabling seamless stakeholder collaboration while maintaining efficient and transparent grading workflows. Key functionalities include completing evaluation forms, managing project details, and calculating final grades.



**Figure 4: Use Case Diagram for ProjectHub**



## 4.2 Package Diagram

The following diagram illustrates the three main components of the ProjectHub system:

**Frontend:** Responsible for handling user interactions and managing the interface. It includes UI components, state management, and services that connect to the backend and Firebase.

**Backend:** Hosts the application logic and provides API endpoints. It manages requests from the frontend, processes data, and integrates with Firebase Services.

**Firebase Services:** Manages authentication, database operations, and file storage. It serves as the central storage and service layer for both the frontend and backend.

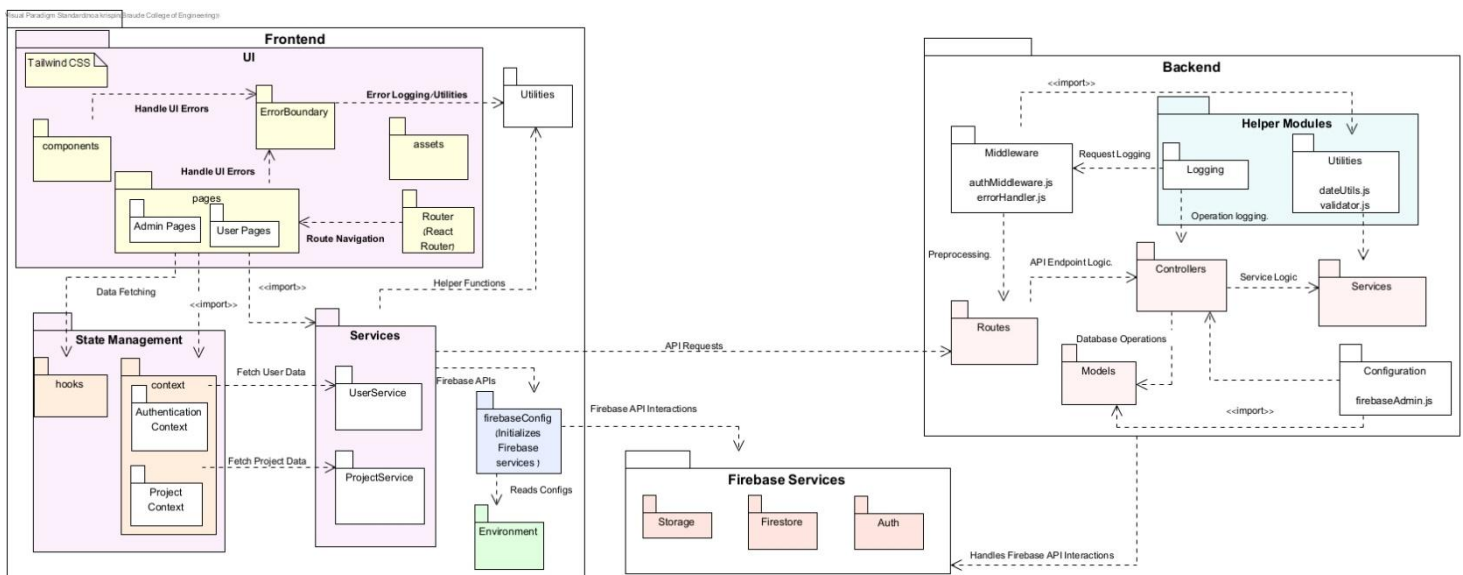


Figure 5: Package Diagram for ProjectHub

## 5 Prototype

Prototyping plays a crucial role in bridging the gap between conceptual designs and functional systems. This chapter outlines the iterative development of the ProjectHub prototype, from early design concepts to the final version. Each stage of the process—initial design, interface development, feedback incorporation, and validation—is detailed to showcase how user needs inform every journey step.

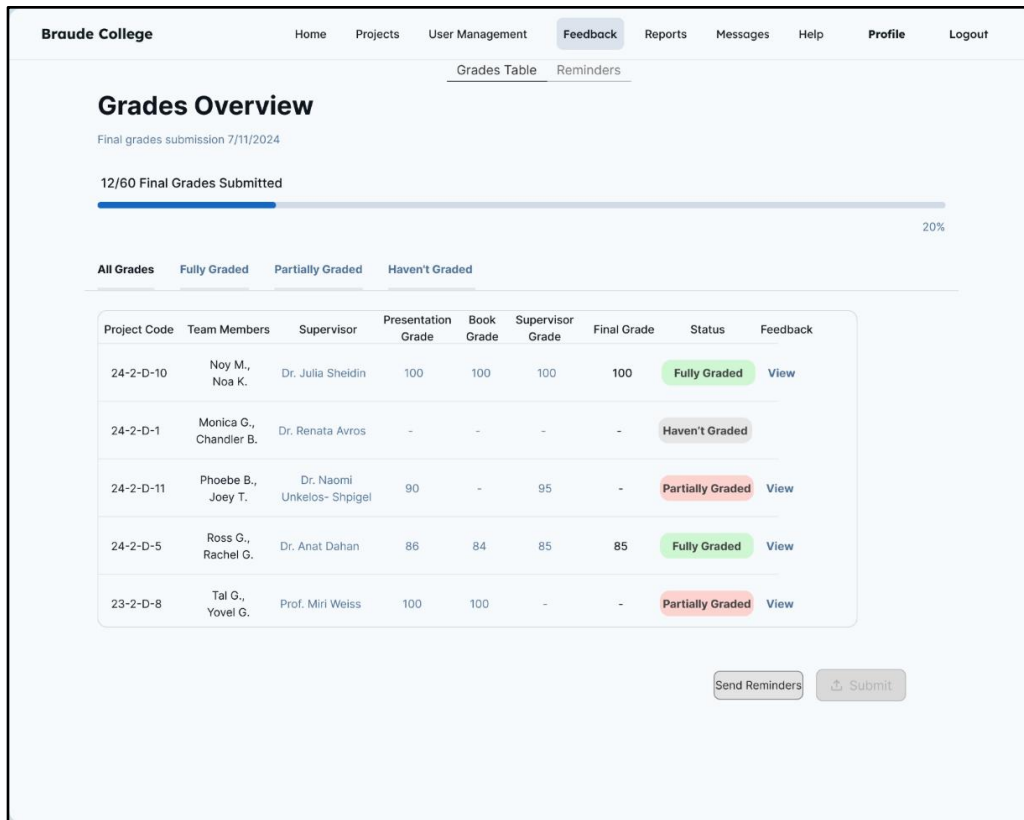
### 5.1 Initial Design Concepts

#### 5.1.1 Design Process

The design of ProjectHub evolved through an iterative process, starting with conceptual sketches in Part A and progressing to prototypes of a polished system in Part B. Early designs focused on meeting the needs of supervisors and coordinators, addressing key inefficiencies such as tracking

assessments, grade management, and creating evaluation forms. Figma was used to create multiple interactive prototypes, refined at each stage based on stakeholder feedback. This approach ensured that the design evolved to align closely with the workflows of supervisors and coordinators.

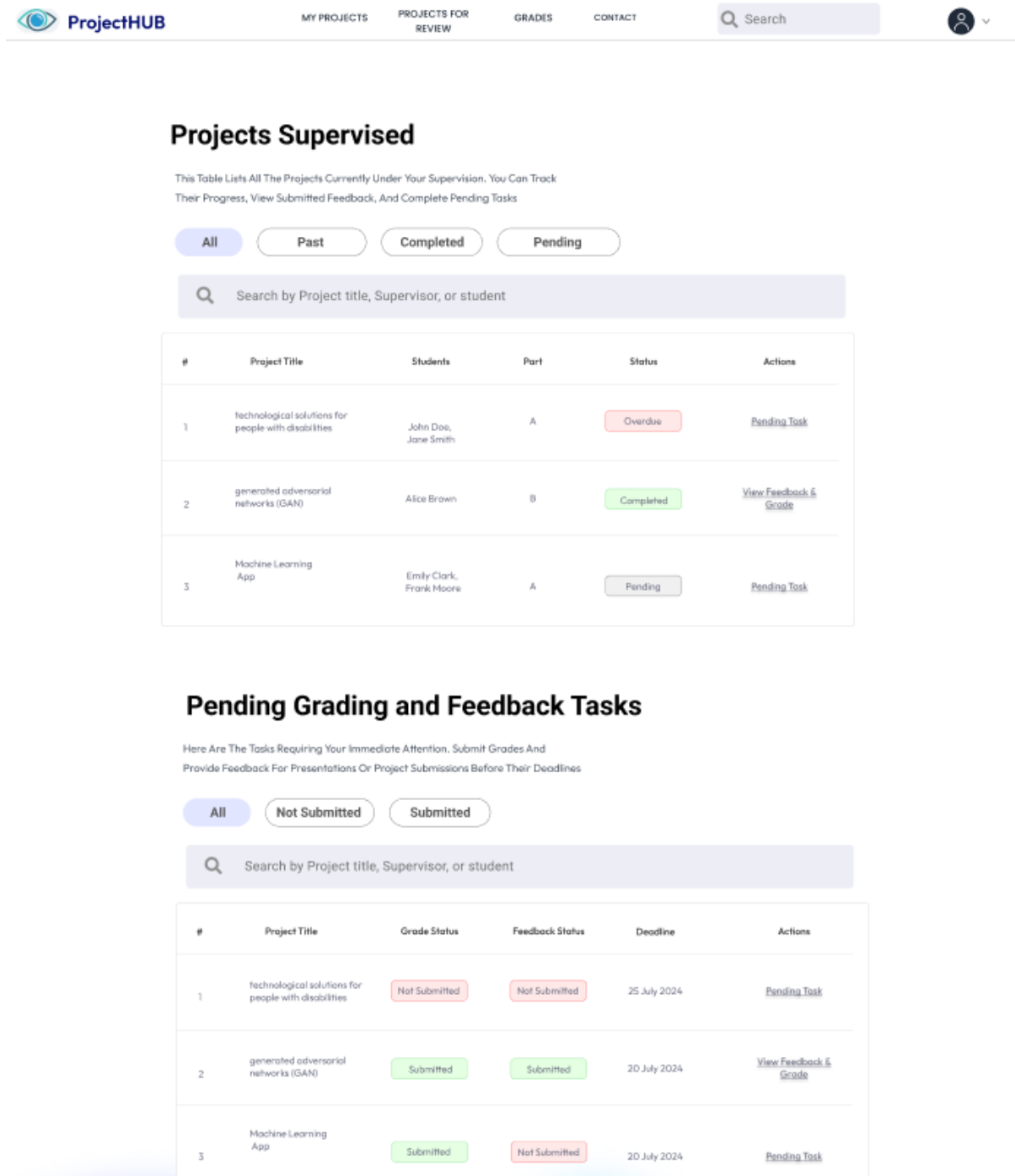
Part A laid the foundation for ProjectHub by establishing key functionalities such as grade tracking and progress management. However, early designs were limited in scope and usability. For instance, the Reports and Notifications for Project Coordinators page (Figure 6). was developed in Part A to provide coordinators with tools for monitoring project submissions and grading progress. This design clarified the importance of real-time updates and centralized tracking. Following its construction, it became clear that a similar page tailored for supervisors was necessary to meet their specific needs.



**Figure 6:** Reports and Notifications for project coordinators

These gaps not only limited the ability of supervisors to manage grades efficiently but also created ambiguities in task ownership and progress visibility. The need for role-specific workflows and dynamic, real-time features highlighted the importance of further development and iterative feedback to make the platform more dynamic and user-friendly.

In Phase B, the design and implementation of the Grade Review for Supervisors feature underwent significant changes based on stakeholder feedback. The initial Figma prototype focused on creating a comprehensive view for supervisors to track grading progress and access detailed project information (Figure 7). This page included a list of projects under review by the supervisor, showing grade statuses and providing links to submission details.



**Figure 7:** Review of grades for supervisor projects and projects he or she grades

A structured table format that displays key details, such as project code, title, students involved, and grade progress (Figure 16) (Figure 17), includes:

- Real-time status updates showing whether grades have been submitted, partially submitted, or are pending.
- Direct links to Git repositories for easy access to project files, streamlining the review process.
- A clear due date column to ensure supervisors are aware of submission schedules briefly.
- A pop-up plane providing essential project details in a concise and accessible format.

Additional improvements were made following facilitation conversations with our supervisor, who also serves as the project coordinator. These discussions helped identify key needs and informed changes such as:

- Filters and tabs (e.g., “All,” “Part A,” and “Part B”) to segment projects by their stage, allowing supervisors to narrow their focus.
- A progress indicator at the top of the page, providing a quick glance at the overall grading completion rate, encouraging timely submissions and improving efficiency.
- Actionable links (e.g., “Grade Presentation”) offering direct access to specific grading forms, reducing the number of steps required to complete reviews.

These refinements demonstrate the transition from a static prototype to a fully interactive and functional system page. Each improvement was guided by user-centered design principles, ensuring that the platform addresses the practical needs of its users while maintaining a streamlined and intuitive interface.

Figma prototypes served as detailed blueprints for system development, guiding the transition from conceptual ideas to practical solutions. These prototypes allowed for iterative refinement based on stakeholder feedback, ensuring alignment between design and user needs. The progression from initial mockups to an interactive, real-time system demonstrates how design decisions were driven by usability and functionality considerations.

The Projects for Review page was refined iteratively to address the specific needs of supervisors and coordinators. Each stage of feedback improved the interface and addressed the practical needs of supervisors and coordinators. Moreover, the design principles and insights gained during this development informed similar refinements across ProjectHub, resulting in a cohesive and intuitive system.

### **5.1.2 Development with Simulated Data**

The development process for ProjectHub began by focusing on the front-end, using simulated data to build and test the user interface before connecting to the backend. This approach enabled rapid iterations on the front-end design, ensuring that client feedback could be incorporated early without dependency on back-end progress. Simulated data was structured as JSON files and mock API responses to replicate real-world scenarios, such as grading submissions and tracking project statuses. This approach enabled independent testing and refinement of key workflows, including real-time screens, grading processes, and submission tracking.

The simulated data was created based on insights from Part I of the project book and iteratively refined throughout development. It included mock user roles, such as students, supervisors, and administrators, complete with corresponding credentials, as well as project details like titles, statuses, deadlines, and attendees. These data sets were paired with a mock API simulating back-end responses, including realistic delays, to ensure accurate testing of workflows. This process clarified the system's database requirements, such as defining tables for projects, users, and evaluations, along with their relationships.

Working with simulated data streamlined the incorporation of client feedback and minimized potential errors. The ability to test workflows independently of the backend allows for early validation of features and informed back-end development. The transition to the live backend was seamless, as the mock structures were mirrored for compatibility, ensuring consistency and reliability. Insights gained during this phase also highlighted additional needs, such as specific API endpoints and error-handling mechanisms.

Using simulated data allowed the team to deliver a polished front-end, refine the system's architecture, and facilitate a reliable back-end integration. This method ensured flexibility for changes and delivered a final product that met high standards of usability and reliability.

## **5.2 Presenting the Prototype**

The ProjectHub prototype was critical step in transitioning from conceptual designs to a fully implemented system. Unlike static design concepts, the prototype provided an interactive platform where stakeholders could test key workflows, evaluate usability, and offer actionable feedback. The prototype enabled early validation of core functionalities and ensured alignment with user expectations by simulating components such as the Grade Review Page, Assessment Form Editor, and Dashboard Overview.

The primary objective of the prototype was to validate usability and ensure that workflows were intuitive and efficient. By allowing stakeholders to interact with working models of the system, usability challenges were identified and addressed, resulting in refinements that enhanced both functionality and user experience. This iterative approach ensured that the final implementation was not only user-centric but also practical and reliable.

Key features of the prototype included:

- **Grade Review Page:** A real-time tool for supervisors to track grade statuses, filter assignments, and manage submission deadlines (Figure 24).
- **Assessment Form Editor:** A dynamic interface enabling coordinators to create, edit, and preview customizable forms (Figure 28) (Figure 29).

Navigation and workflows were designed for simplicity and ease of use. Users could:

- Supervisors accessed detailed project tables with progress indicators and direct links to grade assignments.
- Coordinators utilized a drag-and-drop interface in the Assessment Form Editor to rearrange criteria, adjust weights, and save templates.
- Users could track submission progress via the dashboard, highlighting overdue tasks with actionable prompts.

Stakeholder feedback played a pivotal role in refining the prototype. Supervisors and the project coordinator emphasized the need for clear visual indicators, such as color-coded statuses, to improve task prioritization. Navigation was streamlined with direct links to critical pages, reducing unnecessary steps, while workflows were tailored to the unique responsibilities of each user role. For example, tabs were added to filter projects by stage (e.g., "All," "Part A," "Part B"), and pre-populated templates were introduced in the Assessment Form Editor to expedite form creation.

While the prototype successfully validated workflows, certain advanced features, such as cross-platform compatibility and detailed reporting, were beyond its scope and reserved for future development. However, the prototype's interactive elements directly informed the final system, ensuring continuity between design and implementation. Features like the streamlined Grade Review Page and dynamic Assessment Form Editor became integral components of the final platform, preserving the user-centric improvements tested during the prototype phase.

By bridging the gap between conceptual designs and implementation, the ProjectHub prototype ensured that the final product met high standards of usability and practicality. Its development was instrumental in delivering a cohesive, intuitive system that addressed the needs of its users.

### **5.3 Stakeholder Feedback and Iterations**

The stakeholder review phase played a vital role in ensuring that ProjectHub aligns with the needs and expectations of its users. Throughout this phase, key stakeholders, including supervisors, coordinators, and panel members, engaged with the system's core functionalities to evaluate its usability and effectiveness. Their feedback was instrumental in identifying areas for improvement, refining workflows, and addressing usability challenges. The review process focused on real-world validation, allowing users to interact with the system and provide insights contributing to meaningful refinement.

Regular feedback during the development phase, particularly from the project supervisor and second coordinator, helped shape essential features such as navigation enhancements, progress indicators, and customizable evaluation forms. These iterative adjustments ensured that ProjectHub evolved in response to real user needs. In addition to early feedback, the formal stakeholder review phase served as a critical checkpoint, allowing further refinement based on actual usage. Key insights from this process emphasized the importance of clarity in the grading and review process, accessibility improvements for evaluation forms, and the need for a more intuitive interface to streamline project assessments.

By incorporating these insights, ProjectHub continues to evolve as a user-centric platform that prioritizes accessibility, efficiency, and usability. The iterative process ensures the system remains adaptable to changing user needs and future enhancements. As the platform moves toward full implementation, ongoing feedback will be crucial in further refining functionalities and ensuring seamless integration with Parts 1 and 3 of the projects, solidifying ProjectHub as a comprehensive and efficient project management solution.

### **5.4 Final Prototype and Features**

The final prototype of ProjectHub has been refined through continuous development and stakeholder feedback, resulting in a more intuitive and efficient platform for managing evaluations. The system now fully supports supervisors and coordinators in streamlining the grading and review process while maintaining a clear and structured interface. Throughout development, several key features were prioritized to ensure users could navigate the platform seamlessly and complete their tasks efficiently. The supervisor grading interface provides a structured page for evaluating students' projects, offering clear descriptions and accessible controls. Evaluation and review pages facilitate a comprehensive workflow for assessing project submissions, tracking progress, and finalizing grades. Automated notifications and reminders enable the system to notify evaluators who have yet to submit their reviews, ensuring timely grading. A personal notes feature was incorporated within the project pop-up to allow evaluators

to leave private notes related to a submission. Additionally, administrative oversight tools were implemented, allowing coordinators to manage deadlines, monitor submissions, and oversee the overall evaluation process.

Based on stakeholder recommendations, several refinements were made to improve clarity, usability, and efficiency:

- **Clearer Supervisor Grades Page** – The title and description were revised to reflect the page's purpose better.
- **Clarified Personal Notes Feature** – The project pop-up now provides a clearer explanation of what "personal notes" means.
- **Enhanced Grade Visualization** – Grades on the grading page are now color-coded, making it easier to distinguish between projects.
- **Unlocked Evaluation Tabs After Deadline** – Evaluators can still access and review their evaluations once the deadline has passed.
- **Validation Confirmation for Reminders** – A pop-up confirmation was added on the admin side of the reminders page to prevent accidental notifications.
- **Refined Reminder System** – Reminders now only target evaluators who have not yet submitted grades, avoiding unnecessary notifications.
- **New "Book Reviewer" Column** – A dedicated column was added to the admin side of the grades page to track book reviewers.

These refinements ensure that ProjectHub remains a user-friendly, adaptable, and effective platform for managing final project evaluations. By incorporating user feedback and continuously improving its functionality, the system is now better equipped to meet user needs, enhance workflow efficiency, and provide a seamless evaluation experience for supervisors and coordinators. Moving forward, further refinement will be made based on real-world usage to ensure long-term usability and effectiveness.

## 5.5 Testing Plan

A structured testing plan was implemented to ensure the accuracy, reliability, and usability of ProjectHub. The evaluation focused on core functionalities, including grade submission, automated reminders, evaluation tracking, and system performance.

Testing was conducted in two phases:

- **Phase 1: Internal Testing** – Verified grading workflows, feedback processing, and system performance under controlled conditions.
- **Phase 2: Final Validation** – Conducted real-world testing with supervisors and coordinators to evaluate usability and efficiency.



The plan (see Table 2) outlined key testing areas, objectives, and success criteria for assessing system functionality.

Testing Area	Objective	Title	Result
Grade Submission	Ensure grades are correctly stored and displayed	Submission and Retrieval	Grades were saved, updated, and retrieved successfully
Automated Reminders	Verify notifications are sent on time	Notification System	Email reminders triggered correctly
Evaluation Form Completion	Ensure evaluators can submit forms without errors	Form Submission Flow	Forms were completed, saved, and accessible with no errors
System Performance	Validate response time under load	Load and Stress Testing	System maintained <1min response time under multiple concurrent users
Usability Testing	Ensure intuitive workflows for supervisors	Supervisor Task Execution	Supervisors successfully navigated and completed key tasks

**Table 2:** Summary of Evaluation Plan Metrics and Methods

To ensure comprehensive validation, the testing process incorporated the following methodologies:

- **Performance Benchmarking** – Measured system response times, load handling, and database efficiency.
- **Simulated Use Cases** – Evaluated workflows such as grade submission, feedback retrieval, and automated notifications to ensure seamless functionality.
- **Stakeholder Testing** – Gathered feedback from supervisors and coordinators to assess usability and improve user experience.

Additionally, selected lecturers will use the system throughout the semester to evaluate its convenience and efficiency in an academic setting. Their feedback will be collected, and any necessary improvements will be addressed in a follow-up project, which may introduce additional features and refinements.

The system successfully met all criteria, confirming its ability to streamline the evaluation process while ensuring a seamless user experience. Future updates will incorporate additional enhancements based on real-world feedback to improve the platform further.

## 6 Challenges - Description and Solutions

Throughout the development process, we encountered several challenges that required innovative problem-solving and adaptations to meet the project objectives. This chapter highlights key difficulties faced during the project and the solutions implemented to overcome them.

### Starting with Part 2 Instead of Part 1

The development process began with Part 2 (Book Submission, System Testing, and Grading) instead of Part 1 (Finding a Supervisor and Project Approval). Since Part 1 provides critical information that Part 2 relies on, such as project topics, supervisors, and approvals, the absence of Part 1 created a significant gap in the workflow.

To overcome this issue, we added the ability to upload an Excel file containing all the necessary fields for Part 2, including supervisor information, student details, and project topics (Figure 8).

This approach allowed the system to function seamlessly without Part 1 being implemented.

Student 1 Last Name	Student 1 ID	Student 1 Email	Student 2 First Name	Student 2 Last Name	Student 2 ID	Student 2 Email	Supervisor 1	Supervisor 2	Project Title	Project Description	Part	Type
Malka	318980919	noy.malka@e.braude.ac.il	Noa	Stanislavsky	987654321	noa.stanislavsky@e.braude.ac.il	noa.krispin@e.braude.ac.il	Naomi.Lavi@e.braude.ac.il	projectHUB	project managment platform	A	Development

*Figure 8: Example of the Excel File Format Used for Project Data Upload*

### Incomplete Implementation of Part 3

Part 3 (Presentation Scheduling and Coordination) was partially implemented due to time constraints and its reliance on functionality from Part 1. A specific challenge in this area was determining which panel members were responsible for reviewing specific projects, as the system did not define this information.

As a solution, we added the option to upload an Excel file listing reviewers and their assigned projects (Figure 9). However, the solution remains incomplete, as panel members who are not supervisors cannot currently access the system. This limitation resulted from both the time constraints and the dependency on further development of Part 3 functionality.

Project Code	Presentation Evaluator	Book Evaluator
24-2-D-11	Naomi.Lavi@e.braude.ac.il	noa.krispin@e.braude.ac.il
24-2-D-11		
24-2-D-11	charlie.davis@e.braude.ac.il	

*Figure 9: Excel File Format for Assigning Evaluators to Projects*

### Exporting Grades to the College System

Initially, the system was intended to integrate directly with the college's platform to export grades. However, due to restrictions imposed by the college's security policies, it was not possible to establish a connection to the college system. To overcome this, we developed an Excel file format compatible with the college's system. Grades are now exported from our system into this file, allowing administrative staff to upload them directly to the college's platform. This solution eliminated the need for manual organization of grades and streamlined the process.

### Sending Future Reminders

Project coordinators requested the ability to schedule reminders in advance by selecting a future date and time for the system to automatically send notifications to supervisors. This feature would allow better planning and reduce the need for manual intervention.

However, implementing scheduled reminders posed several technical challenges:

- It required a dedicated database table to store scheduled reminders.
- An automated scheduler would be necessary to trigger reminders at the specified time.
- Maintaining an ongoing scheduling service would increase system complexity and require additional infrastructure, which is not supported by Vercel's serverless environment.

To avoid these complexities, we implemented only manual reminder sending instead of scheduling future reminders. Project coordinators can send reminders whenever necessary, ensuring evaluators receive notifications without the overhead of managing a scheduling system. This decision simplified system maintenance while still providing the essential functionality needed for timely reminders.

## 7 Conclusions

This project aimed to develop an efficient and user-friendly system for managing key workflows in the project lifecycle, including grading, feedback collection, and automated notifications. The system was designed to streamline processes for supervisors, coordinators, and panel members, ultimately reducing administrative workload and improving usability.

The project successfully achieved the following objectives:

- **Grading and Feedback Submission:** Implemented features to enable supervisors and panel members to submit grades and feedback efficiently.
- **Automated Notifications:** Developed a reminder system to reduce delays in grade submissions, ensuring timely evaluations.
- **Data Integration:** Enabled data import through Excel files to overcome the absence of certain functionalities in Parts 1 and 3.

- **Performance and Usability:** Validated the system's performance and usability through rigorous testing, meeting benchmarks for response times and user satisfaction.

Despite these achievements, the project faced several challenges, including incomplete functionality for Parts 1 and 3, as well as database selection issues during development. However, these challenges were addressed through innovative solutions, such as using Firebase for the database and implementing Excel-based workflows to bridge gaps.

The project provided valuable learning opportunities, including:

- The importance of iterative testing and feedback loops during development.
- Effective problem-solving when dealing with integration challenges and time constraints.
- The significance of stakeholder communication in refining system workflows.

While the system meets the core objectives of Part 2, there are opportunities for future improvements:

- **Part 1 Implementation:** Adding functionality for Finding a Supervisor and Project Approval to provide seamless integration with Part 2 workflows.
- **Part 3 Enhancements:** Expanding access to panel members and implementing complete scheduling features for presentations.
- **Advanced Integrations:** Exploring the possibility of integrating directly with external systems, such as the college's platform, to reduce reliance on Excel-based workflows.

In conclusion, the system successfully addressed the primary objectives of Part 2, delivering a functional, reliable, and intuitive platform for managing project workflows. The lessons learned and the foundation established by this project will guide future developments and improvements.

## 8 User Documentation

### 8.1 User Guide

#### 8.1.1 Introduction

Welcome to ProjectHub, a platform created to assist supervisors and project coordinators in managing final projects efficiently. Designed to streamline workflows and improve coordination, ProjectHub provides an intuitive interface to handle evaluation processes and ensure smooth project management.

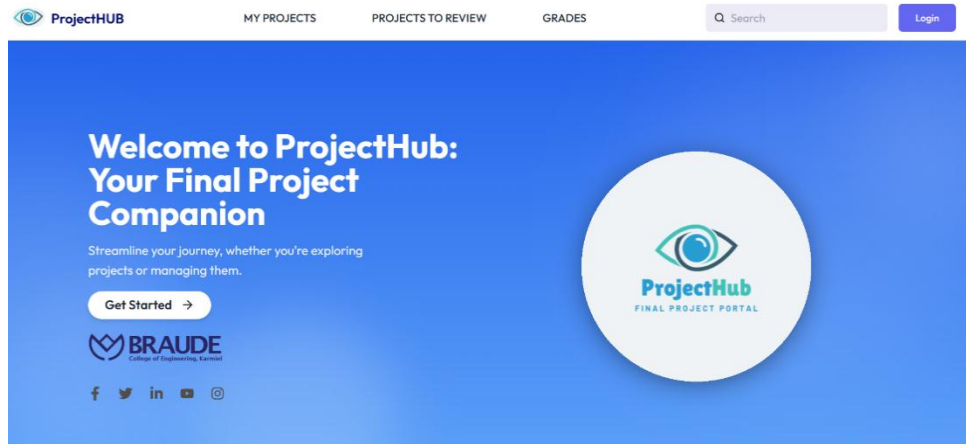
This guide will help you understand how to use the platform effectively, from navigating the interface to performing essential tasks related to project evaluations.

#### 8.1.2 Getting Started

Before using ProjectHub, follow these steps to log in and access your profile.

## 1. Accessing ProjectHub:

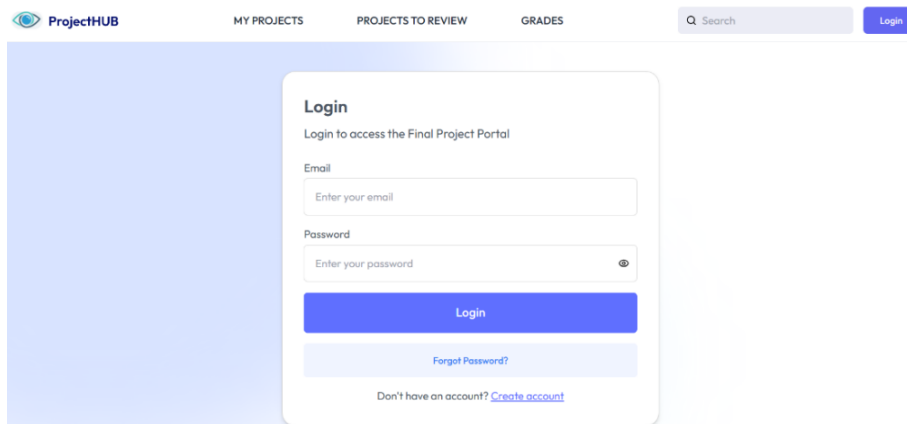
- Open your web browser and go to [ProjectHub](#) platform.
- The homepage will appear, welcoming you to the system.
- You can proceed in two ways:
  - Click "Get Started" on the homepage.
  - Click the "Login" button in the top navigation bar.



**Figure 10:** ProjectHub Homepage - Accessing the Platform

## 2. Logging In:

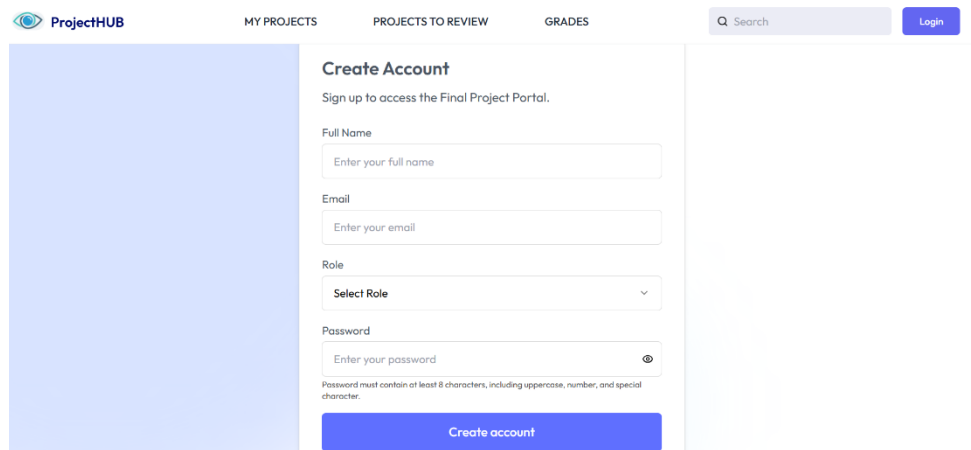
- On the login page, enter your Braude email address (must end with braude.ac.il) and password.
- Click the Login button to access your account.
- If you forgot your password, click Forgot Password? and follow the instructions to reset it.
- If you don't have an account, click "Create Account" to sign up.



**Figure 11:** ProjectHub Login Page – Enter Credentials to Access the System.

### 3. Creating an Account

- On the registration page, enter:
  - Full Name
  - Braude Email Address (must end with braude.ac.il)
  - Role
  - Password (Must be at least 8 characters long and include one uppercase letter, one number, and one special character)
- Click "Create Account" to complete the process.

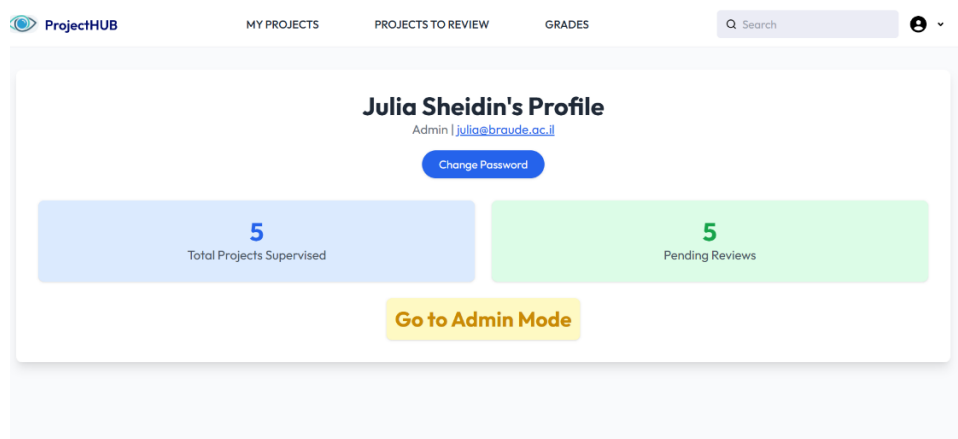


The screenshot shows the 'Create Account' page on ProjectHub. The page has a header with the ProjectHub logo and navigation links: MY PROJECTS, PROJECTS TO REVIEW, and GRADES. There is a search bar and a 'Login' button. The main content area is titled 'Create Account' and includes the instruction 'Sign up to access the Final Project Portal.' Below this are four input fields: 'Full Name' (with placeholder 'Enter your full name'), 'Email' (with placeholder 'Enter your email'), 'Role' (a dropdown menu with 'Select Role' and a downward arrow), and 'Password' (with placeholder 'Enter your password' and a password strength icon). A note below the password field states: 'Password must contain at least 8 characters, including uppercase, number, and special character.' At the bottom is a blue 'Create account' button.

**Figure 12:** ProjectHub Registration Page – Creating a New Account

### 4. Navigating Your Profile

- After logging in, you will be taken to your profile page.  
Here, you can:
  - View the number of projects you supervise.
  - See pending project reviews.
  - Change your password.
  - Enter Admin Mode (if applicable).



The screenshot shows the profile page for 'Julia Sheidin'. The header is the same as in Figure 12. The main content area is titled 'Julia Sheidin's Profile' and includes the text 'Admin | [julia@braude.ac.il](mailto:julia@braude.ac.il)'. Below this is a blue 'Change Password' button. There are two large colored boxes: a blue one on the left showing '5 Total Projects Supervised' and a green one on the right showing '5 Pending Reviews'. At the bottom is a yellow 'Go to Admin Mode' button.

**Figure 13:** ProjectHub Profile Page – Supervisor/Admin Dashboard Overview

### 8.1.3 Using the Website

This section provides an overview of the ProjectHub interface for both supervisors and project coordinators. It explains how to navigate the platform, track projects, perform evaluations, manage grading, and access administrative features.

- The first part of this section covers functionalities available to all users (supervisors).
- The second part details additional tools and actions available exclusively to project coordinators (admins).

- **First Part: Supervisor Functionalities**

- 1. Navigating the Dashboard**

- After logging in, all users (supervisors and project coordinators) can access different sections of the platform via:

- The navigation bar at the top, which includes:

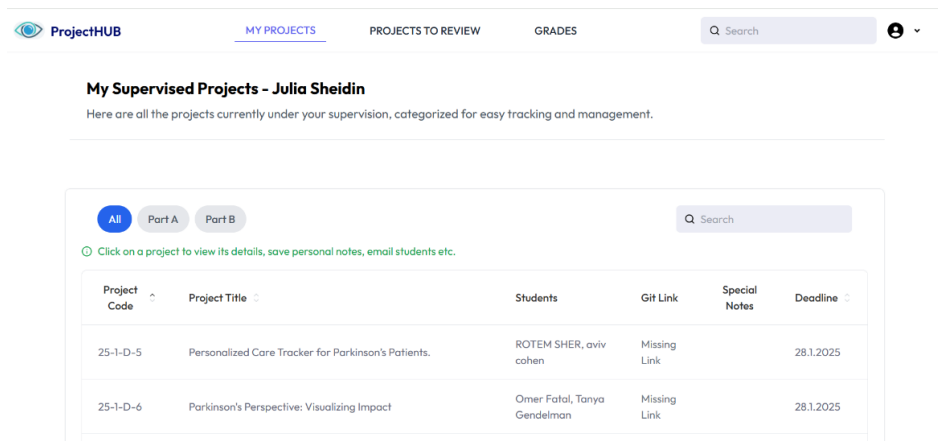
- My Projects – View and manage supervised projects.
    - Projects to Review – Access projects requiring evaluation.
    - Grades – Track project grading progress.
    - User Icon Menu (top-right corner):
      - My Profile – Access personal profile settings and view supervised projects.
      - Admin Dashboard (only for coordinators) – Switch to the admin interface.
      - Logout – Sign out of the platform.

- Note:** Project coordinators have access to all supervisor functionalities in addition to their administrative tools.

- 2. My Projects**

- Accessible via the My Profile page or the My Projects tab at navigation.

- Displays a table of all supervised projects.
    - Projects can be filtered by:
      - All Projects
      - Part A Projects
      - Part B Projects
    - You can search for projects using any column on the table.

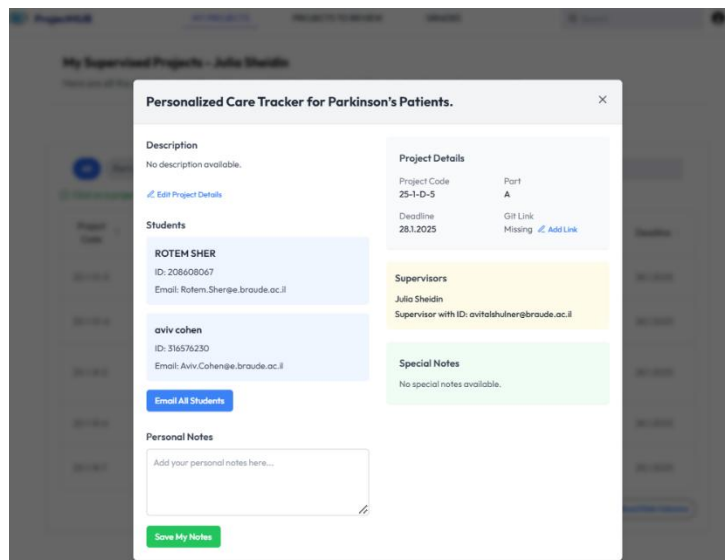


**Figure 14:** My Projects Page – Viewing and Filtering Supervised Projects

- **Viewing a Project's Details**

Clicking on a project opens a pop-up window where supervisors can:

- Add a project description.
- Attach a Git repository link.
- View information about the students.
- Email students directly.
- See special notes from project coordinators.
- Write personal notes (visible only to the supervisor).



**Figure 15:** Project Details Pop-up – Viewing and Managing Project Information

### 3. Projects to Review

Accessible via the My Profile page (clicking "Pending Reviews") or the Projects to Review tab.



- **Contains two sub-tabs:**
  - My Projects** – Projects assigned to the supervisor for grading.
  - Other Projects** – Projects where the supervisor is required to provide additional evaluations.

- **My Projects**

Displays a table of projects requiring a supervisor grade.

- Clicking on a project opens a pop-up window with project details.
- The Supervisor Grade column:
  - If no grade is given → The "Supervisor Grade" button appears (clicking it opens the grading form).
  - If a grade is given → The grade appears as a button (clicking it allows editing the evaluation).

**Note:** If no projects are available for review, the system will display "No data to display."

ProjectHUB MY PROJECTS **PROJECTS TO REVIEW** GRADES Search Projects...

My Projects Other Projects

**My Projects To Review – Julia Sheidin**  
Projects you need to grade as a supervisor

0/5 Forms Submitted

All Part A Part B Search

Click on a row to view project details and on a grade to add or edit evaluations.

Project Code	Project Title	Students	Git Link	Supervisor Grade	Deadline
25-I-D-5	Personalized Care Tracker for Parkinson's Patients.	ROTEM SHER aviv cohen	Missing Link	Grade Supervisor	4.2.2025
25-I-D-6	Parkinson's Perspective: Visualizing Impact	Omer Fatal Tanya Gendelman	Missing Link	Grade Supervisor	Passed

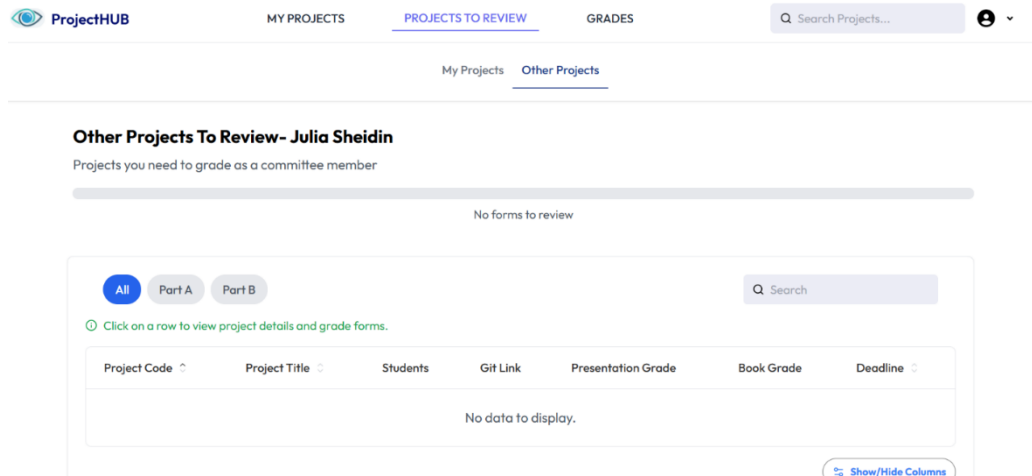
**Figure 16:** ProjectHub "My Projects" Review Page – Evaluating supervisor Grades

- **Other Projects**

Displays projects where the supervisor must give a book or presentation grade.

- Clicking on a project opens a pop-up panel with project details.
- The Book Grade and Presentation Grade columns:
  - If no grade is given → The grade button appears (clicking it opens the respective evaluation form).
  - If a grade is given → The grade appears as a button (clicking it allows editing the evaluation).

**Note:** If no projects are available for review, the system will display "No data to display."



**Figure 17:** ProjectHUB "Other Projects" Review Page – Evaluating Book and Presentation Grades

#### 4. Filling Out Evaluation Forms

Evaluation forms are used by supervisors and evaluators to assess students' projects. Clicking on a Supervisor Grade, Book Grade, or Presentation Grade in the Projects to Review page redirects the user to the corresponding evaluation form.

- **Types of Evaluation Forms**

- Supervisor Evaluation Form – Used by supervisors to assess the overall project.
- Book Evaluation Form – Used to evaluate the written report (Part A or Part B).
- Presentation Evaluation Form – Used to assess the student's presentation (Part A or Part B).

- **Completing the Evaluation Form**

- Each form contains questions defined by project coordinators.
- Required fields are marked with a red asterisk (\*).
- Some fields require numeric input (scores 0-100), while others allow text responses.

- **Form Progress Indicator**

- A progress wheel tracks how much of the form has been completed.
- The form can be submitted only after all required fields are filled.

- **Editing Submitted Grades**

- Once submitted, the grade appears in the Projects to Review section.
- You can edit the form by clicking on the assigned grade at any time.

**Supervisor Evaluation Form**

This is an evaluation form for the supervisor to assess the project and the students' performance.

**Project Code \***  
25-1-D-5

**Project Name \***  
Personalized Care Tracker for Parkinson's Patients.

**Evaluator Name \***  
Julia Sheidin

**Overall Project Evaluation**

**Analysis and solution formation \***  
Quality of the work process, use of original ideas, level of questions raised during the work.

Form Progress: 0%

**Figure 18:** Supervisor Evaluation Form – Evaluating a New Grade or Editing Submitted Grades

## 5. Grades

Accessible via the Grades tab in the navigation bar.

- Displays all projects supervised by the user along with their final grades.
- Clicking on a project opens a pop-up panel showing:
  - Book project evaluation comments.
  - Presentation evaluation comments.

**Supervisor Grades Feedback**

View and manage grades for projects assigned to you.

[Refresh Grades](#)  
Click to see the updated grades

**All** **Part A** **Part B**

[Search](#)

[Weighted grades for each student. Click on a row to view detailed project assessments.](#)

Project Code	Part	Student Name	Book Grade	Presentation Grade	Supervisor Grade	Final Grade	Grade Status	Deadline
25-1-D-5	A	aviv cohen					Not graded	28.1.2025
25-1-D-5	A	ROTEM SHER					Not graded	28.1.2025
25-1-D-6	A	Omer Fatal					Not graded	28.1.2025

**Figure 19:** Grades Feedback Page – Grading Supervised Projects.

**Supervisor Grades Feedback**

View and manage grades for projects assigned to you.

[Refresh Grades](#)  
Click to see the updated grades

**All** **Part A** **Part B**

[Search](#)

[Weighted grades for each student. Click on a row to view detailed project assessments.](#)

**"Personalized Care Tracker for Parkinson's Patients." Assessment**

[Book Form Responses](#)

[Presentation Form Responses](#)

**Figure 20:** Grades Feedback – Viewing Book & Presentation Grade Comments.

- **Second Part: project coordinators (admins)**

The Project Coordinator (Admin) Dashboard provides additional tools for managing projects, evaluations, and deadlines. Since all project coordinators are also supervisors, they have access to both supervisor and administrative functionalities.

### **1. Accessing the Admin Dashboard.**

There are two ways to switch to the Admin Mode:

- i. Via My Profile – Click the "Go to Admin Mode" button on the profile page.
- ii. Via the Navigation Bar – Click the user icon in the top-right corner and select "Admin Dashboard".

Once switched to the Admin Mode, the first page displayed is the Projects Management page.

### **2. Navigation Bar (Top Menu):**

- **Projects** – Manage all projects, including assigning supervisors, editing details, and handling deadlines.
- **Reminders** – Set deadlines and send reminders to supervisors.
- **Grades** – View and manage all project evaluations.
- **Upload Files** – Bulk upload projects and evaluator assignments.
- **Management** – Manage users and modify evaluation forms.
- **Supervisor Mode** – Return to the regular supervisor view.
- **Logout** – Sign out of the platform.

### **3. Managing Projects**

The Projects page allows project coordinators to manage all projects in the system. It includes three sub-tabs:

- i. All Projects – View and manage all projects.
- ii. Part A – Displays projects in their planning and initial development phase.
- iii. Part B – Displays projects in their implementation and completion phase.

- **All Projects Sub-tab**

If no projects exist, an "Upload Excel File" button appears, allowing project coordinators to bulk upload projects.

If projects exist, they are displayed in a table.

The following fields are editable:

- Project Title – Clicking on the field opens an editing window.
- Deadline – Can be adjusted (e.g., extended for appeals).

Other functionalities:

- Clicking a student's name opens a pop-up with student details.

- Clicking on Special Notes opens a pop-up where coordinators can write notes (visible to supervisors).
- Projects can be deleted if necessary.

ProjectHub

PROJECTS REMINDERS GRADES UPLOAD FILES MANAGEMENT

Supervisor Mode Logout

All Projects Part A Part B

Here you can view and manage all projects, including both Part A and Part B projects. Use the filters above to focus on specific project types.

49 Total Projects

Upload Excel File

Click on blue cells to edit them. Click on student name to see their details.

Project Code	Project Title	Students	Supervisors	Part	Type	Deadline	Special Notes	Actions
24-2-D-30	projectHUB	Noya Malka Noa Krispin	Alice Doe Prof. Naomi Lavi	A	Development	No Deadline	Add note	
25-1-D-1	A tool for analyzing fNIRS hyper-scanning using graph measures.	Sara Saleh Yonatan Sherer	Anat Dahan	A	Development	28.1.2025	Add note	

Search Projects

Edit Notes for projectHUB

Enter your notes here...

Close Save

Edit Project Title

projectHUB

Close Save

Student Details

Name  
Noa Stanislavsky

Student ID  
987654321

Email  
noa.stanislavsky@e.braude.ac.il

Close

**Figure 21:** ProjectHub Projects Management – Viewing, Editing, and Managing Project Information.

- **Part A & Part B Tab**

Projects in Part A can be moved to Part B only if Part B is empty.

Part B projects can be:

- Deleted if necessary.
- Exported to an Excel file.

If no projects exist, a "No projects available to display" message appears.

#### 4. Sending Reminders

The Reminders page allows project coordinators to set deadlines and send reminders about Submitting grades to supervisors.

- **Setting Deadlines**

- A global deadline for all projects can be set.
- Setting a deadline automatically sends an email notification to all supervisors.

The screenshot shows the 'ProjectHub' interface with the 'REMINDERS' tab selected. The 'Set Deadline' form is displayed, allowing users to manage the global deadline for all projects. The form includes a 'Deadline' field with a calendar icon and a 'dd/mm/yyyy' placeholder. Below this is an 'Email Message (Optional)' section with a default message: 'A new deadline has been set for project submissions. Please log in to the system to view the details.' and a text area for a custom message. A 'Save Deadline' button is at the bottom.

Figure 22: ProjectHub Reminders Page – Setting Deadlines for Project Grades Submissions.

- **Sending Reminders**

- Allow coordinators to send email reminders to all supervisors.
- A default message is provided, but a custom message can be added.
- Below the reminders section, a table shows grading status:
  - ✓ (Check Mark) – The evaluator has submitted a grade.
  - ✗ (X Mark) – The evaluator has not submitted a grade.
  - "-" (Dash) – The evaluator has not been assigned yet.

The screenshot shows the 'ProjectHub' interface with the 'REMINDERS' tab selected. The 'Project Reminders' section is visible, with a 'Reminder Message' field and a 'Send Reminders' button. Below this is the 'Projects Evaluators and Grades' section, which includes a table with columns for Project Code, Evaluator Name, Presentation Grade, Book Grade, Supervisor Grade, and Deadline. The table shows two rows of data.

Project Code	Evaluator Name	Presentation Grade	Book Grade	Supervisor Grade	Deadline
24-2-D-11	Prof. Naomi Lavi	-	-	✗	30.1.2025
24-2-D-11	Alice Doe	-	-	✓	30.1.2025

Figure 23: ProjectHub Reminders Page – Sending Reminders about Grading Status.

## 5. Managing Grades

The Grades page allows project coordinators to track and manage all project evaluations.

- **Grades Overview**

Displays a table of all projects, with filtering options for:

- All Projects
- Part A
- Part B

The table includes columns for:

- Project Code
- Project Section (Part A or B)
- Student Name
- Supervisor Name
- Book Grade
- Presentation Grade
- Supervisor Grade
- Final Grade
- Deadline

Clicking on a project opens a pop-up panel containing book and presentation evaluation comments.

- **Additional Features**

- Search for projects using any field in the table.
- Exporting Grades to Excel – Grades can be exported to an Excel file with the following required columns:
  - Student ID
  - Student First Name
  - Student Last Name
  - Final Grade
- Refresh grades to update evaluations in real time.

PROJECTS

REMINDERS

GRADES

UPLOAD FILES

MANAGEMENT

Supervisor Mode

Logout

Project Grades Management

View and manage grades for all ongoing projects.

Export to Excel

Refresh Grades

Click to see the updated grades

All

Part A

Part B

Q Search

Weighted grades for each student. Click on a row to view detailed project assessments.

Project Code	Part	Student Name	Supervisors	Book Grade	Presentation Grade	Supervisor Grade	Final Grade	Grade Status	Deadline
24-2-D-30	A	Noa Krispin	Alice Doe Prof. Naomi Lavi					Not graded	No Deadline
24-2-D-30	A	Noya Malka	Alice Doe Prof. Naomi Lavi					Not graded	No Deadline
25-1-D-1	A	Sara Saleh	Anat Dahan					Not graded	28.1.2025
25-1-D-1	A	Yonatan Sherer	Anat Dahan					Not graded	28.1.2025

**Figure 24:** ProjectHub Grades Management Page – Viewing and Managing Project Grades.

## 6. File Uploads

The Upload Files Page contains two tabs:

- Upload Project File
- Upload Evaluator File

- **Uploading Projects**

- Displays mandatory and optional columns for the Excel template.
- Uploading an invalid file triggers an error message.
- A valid file is uploaded successfully, and the user is redirected to Projects Page.

- **Uploading Evaluators**

- Displays mandatory and optional columns for the Excel template.
- Uploading an invalid file triggers an error message.
- A valid file is uploaded successfully, and the user is redirected to Grades Page.

PROJECTS

REMINDERS

GRADES

UPLOAD FILES

MANAGEMENT

Supervisor Mode

Logout

Projects

Evaluators

Project Upload Instructions

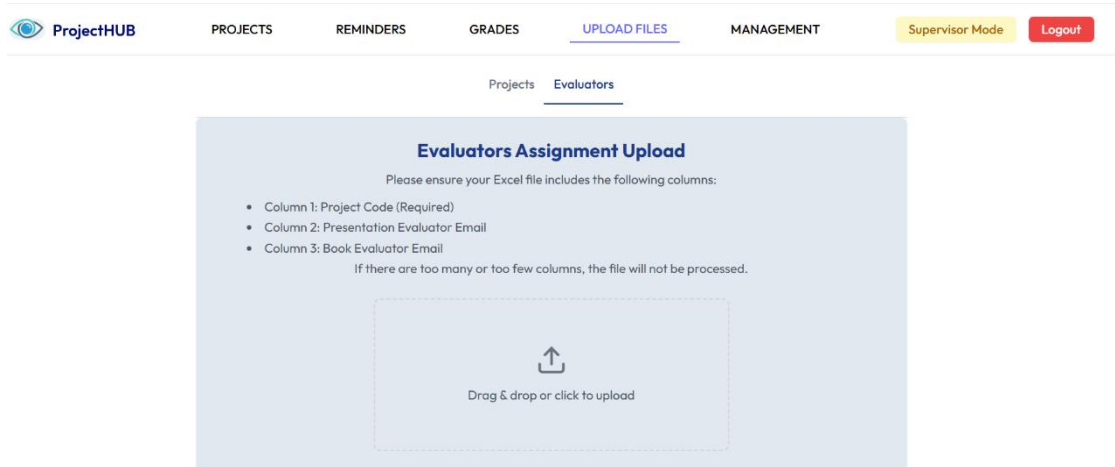
Please ensure your Excel file follows the format below before uploading.

- Column 1: Project Code (Required)
- Column 2: Student 1 First Name (Required)
- Column 3: Student 1 Last Name (Required)
- Column 4: Student 1 ID (Required - 9 digits)
- Column 5: Student 1 Email (Required - @e.braude.ac.il)
- Column 6: Student 2 First Name (Optional)
- Column 7: Student 2 Last Name (Optional)
- Column 8: Student 2 ID (Optional - 9 digits)
- Column 9: Student 2 Email (Optional - @e.braude.ac.il)
- Column 10: Supervisor 1 Email (Required)
- Column 11: Supervisor 2 Email (Optional)
- Column 12: Project Title (Required)
- Column 13: Project Description (Optional)
- Column 14: Part (Required - A or B)
- Column 15: Type (Required - Development or Research)

Drag & drop or click to upload

**Figure 25:** ProjectHub Upload Files Page – Project Upload Instructions.





**Figure 26:** ProjectHub Upload Files Page – Evaluators Assignment Upload.

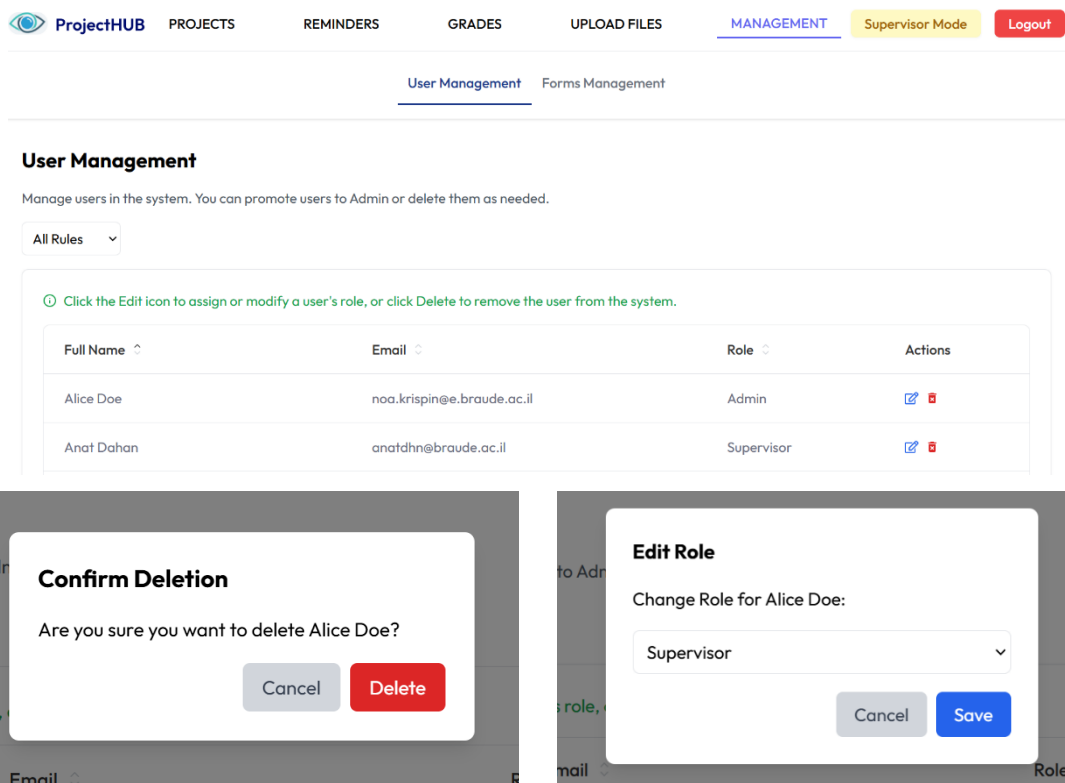
## 7. Management

The Management page provides tools for user management and form management. It consists of two tabs:

- i. User Management
- ii. Form Management

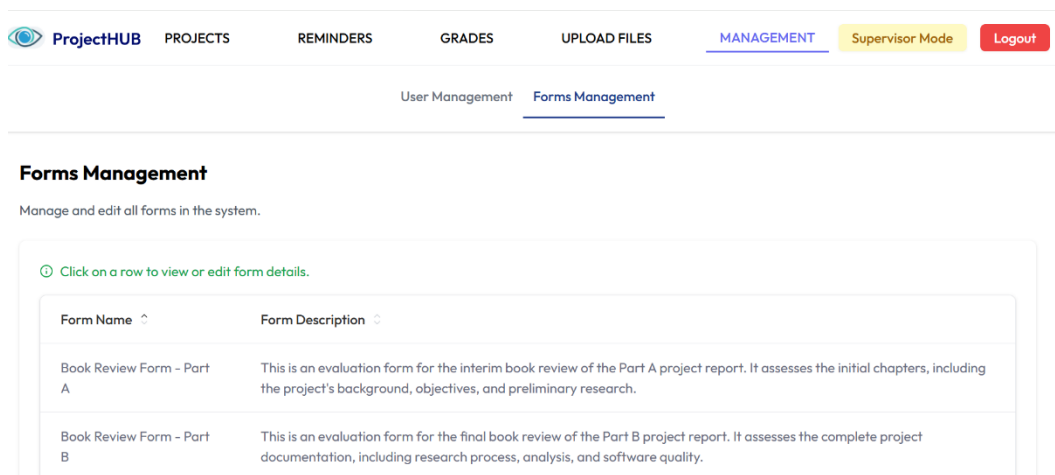
- **User Management**

- Displays a table of all registered users in the system.
- Allows project coordinators (admins) to modify user roles or remove users from the platform.
- Changing User Roles
  - Clicking the edit icon next to a user's role opens a pop-up window.
  - Users can be changed between roles:
    - Supervisor(default role).
    - Admin (admins have supervisor permissions plus additional management tools).
  - Clicking Save updates the user's role.
- Deleting a User
  - Clicking the delete icon prompts a confirmation message.
  - If confirmed, the user is permanently removed from the system.



**Figure 27:** ProjectHub User Management – Managing User Roles and Deletion.

- **Form Management**
  - Displays a table of all evaluation forms used in the system.
  - Clicking on a form opens its detailed view, where admins can view, edit, or delete form questions.



**Figure 28:** ProjectHub Forms Management – Viewing and Editing Evaluation Forms.

- **Editing an Evaluation Form**
  - Clicking Edit Form enables form modifications.
  - The following elements can be modified:

- Form Name
- Form Description
- Existing Questions (text, required status, response type, weight etc.)
- Adding New Questions
- Deleting Questions
- Clicking Save updates the form in the system.

**Figure 29:** ProjectHub Evaluation Form – Editing an Existing Form.

### 8.1.4 Troubleshooting

This section should provide solutions to common issues users may encounter while using ProjectHub. The goal is to help supervisors and project coordinators resolve problems efficiently without needing external support.

#### 1. Login Issues

- **Problem:** Cannot log in to the platform.
- **Solution:**
  - Ensure that you are using your Braude email (must end with `@braude.ac.il`).
  - Check that you have entered the correct password.
  - If you forgot your password, click "Forgot Password?" and follow the reset instructions.
  - Ensure that your internet connection is stable.
  - If the issue persists, contact the Project Coordinators.

#### 2. Account Registration Issues

- **Problem:** Cannot create an account.
- **Solution:**

- Ensure you are using a valid Braude email.
- Check that your password meets the requirements (at least 8 characters, one uppercase letter, one number, and one special character).
- If you already have an account, try logging in instead.

### 3. Project or Grade Not Appearing

- **Problem:** A project or grade is missing.
- **Solution:**
  - Ensure you are in the correct section (e.g., Part A / Part B).
  - Click Refresh Grades on the Grades page to update the latest evaluations.
  - Supervisors should check if they have been assigned to the project.
  - If a project was recently uploaded, wait a few minutes for it to process.

### 4. Unable to Submit or Edit Grades

- **Problem:** Cannot enter or modify a grade.
- **Solution:**
  - Ensure that the grading deadline has not passed.
  - If the grading form is locked, contact the project coordinator to request a deadline for extension.
  - Check if the evaluation form is complete; all required fields (marked with a red asterisk) must be filled before submission.

### 5. Error Uploading Files (Projects or Evaluators)

- **Problem:** Cannot upload an Excel file.
- **Solution:**
  - Ensure that the file follows the required format as listed in the Upload Files section.
  - Check that all mandatory columns are included.
  - Ensure that the file type is Excel (.xlsx or .csv).
  - If the file is valid but still not uploading, refresh the page and try again.

### 6. System Errors (Page Not Loading, Freezing, or Crashing)

- **Problem:** The platform is unresponsive or shows errors.
- **Solution:**
  - Try refreshing the page (`Ctrl + R` on Windows, `Cmd + R` on Mac).
  - Log out and log back in.
  - Clear your browser cache and cookies.
  - Use a different browser (recommended: Google Chrome or Mozilla Firefox).
  - If the issue continues, report the problem to the support team.

## 8.2 Maintenance Guide

### 8.2.1 Overview

The Maintenance Guide provides essential details for maintaining the ProjectHub platform, ensuring system stability, security, and optimal performance. It includes technical requirements, system administration procedures, debugging practices, and recommendations for future maintenance.

ProjectHub is built using a modern web-based technology stack that includes a React-based frontend, a Node.js backend, and a Firestore database. The deployment is handled through Vercel for seamless continuous integration and deployment.

This guide serves as a reference for developers, system administrators, and project coordinators responsible for maintaining the platform. It outlines the following key aspects:

- **Technical Requirements:** Describes hardware, software, and dependencies required to run the system.
- **System Administration:** Covers user management, project handling, and configuration tasks.
- **Debugging and Logs:** Explains troubleshooting procedures and log management for issue tracking.
- **Future Maintenance:** Provides recommendations for long-term improvements and security measures.

### 8.2.2 Technical Requirements

This section outlines the technical infrastructure required to deploy, maintain, and run the ProjectHub platform. It includes details on hosting, backend and frontend dependencies, supported browsers, and third-party integrations.

#### 1. Server Requirements

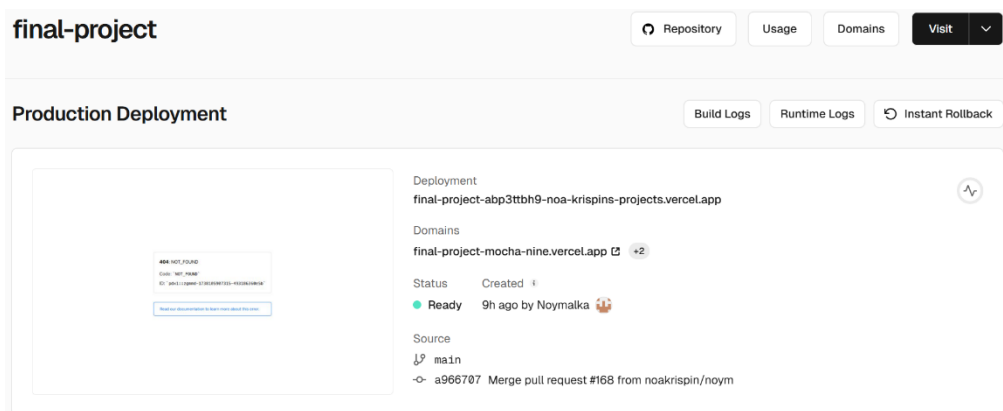
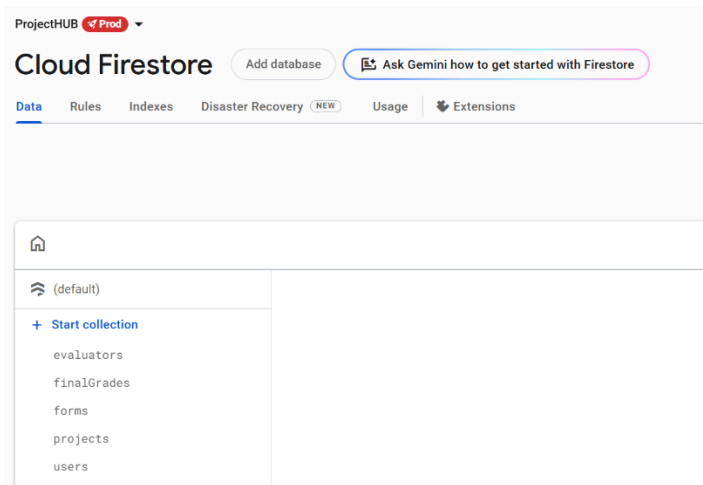
ProjectHub is deployed using modern cloud technologies, with the following setup:

- **Hosting Environment:** The platform is hosted on Vercel, providing automatic deployments, CDN support, and serverless functions for backend processing.
- **Database:** ProjectHub uses Google Firestore (Firebase Cloud Firestore) as a NoSQL cloud database to store user data, project details, evaluations, and grading records.
- **Backend Technology:** The backend is built using Node.js (version 20.x) and runs on Express.js to handle API requests.

```

backend > ▲ vercel.json > ...
1  {
2    "version": 2,
3    "builds": [
4      {
5        "src": "server.js",
6        "use": "@vercel/node"
7      }
8    ],
9    "routes": [
10     {
11       "src": "/api/(.*)",
12       "dest": "server.js"
13     }
14   ]
15 }

```



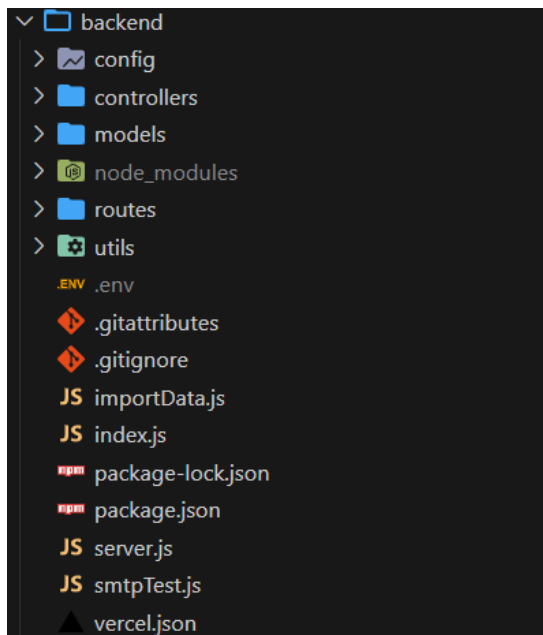
## Backend Dependencies

The backend relies on the following key dependencies:

- **Express.js (4.21.1):** Handles API routes and server logic.
- **Firebase Admin SDK (13.0.2):** Enables database interaction with Firestore.
- **Axios (1.7.9):** Handles HTTP requests.
- **Nodemailer (6.9.16):** Sends email notifications to users.
- **Bcrypt.js (2.4.3):** Encrypts passwords for security.
- **Helmet (7.1.0):** Enhances security by setting various HTTP headers.
- **Compression (1.7.4):** Improves performance by compressing HTTP responses.
- **CORS (2.8.5):** Allows secure cross-origin requests.
- **Express-rate-limit (7.1.5):** Prevents API abuse by limiting repeated requests.
- **Dotenv (16.4.7):** Manages environment variables.

## Development Dependencies

- **Nodemon (3.0.2):** Automatically restarts the server during development.
- **ESLint & Prettier:** Ensures code quality and formatting.



```
backend > npm package.json > ...
1  {
2    "name": "backend",
3    "version": "1.0.0",
4    "main": "server.js",
5    "engines": {
6      "node": "20.x"
7    },
8    "scripts": {
9      "test": "echo \"Error: no test specified\" && exit 1",
10     "start": "node server.js",
11     "dev": "nodemon server.js",
12     "vercel-build": "echo \"Building for production\""
13   },
14   "keywords": [],
15   "author": "",
16   "license": "ISC",
17   "dependencies": {
18     "axios": "1.7.9",
19     "bcryptjs": "2.4.3",
20     "compression": "1.7.4",
21     "cors": "2.8.5",
22     "dotenv": "16.4.7",
23     "express": "4.21.1",
24     "express-rate-limit": "7.1.5",
25     "firebase-admin": "^13.0.2",
26     "helmet": "7.1.0",
27     "mustache": "4.2.0",
28     "nodemailer": "6.9.16"
29   },
30   "devDependencies": {
31     "nodemon": "3.0.2"
32   },
33   "description": "Backend API for Final Project Management System"
34 }
35
```

## Environment Variables (`.env` File)

The backend of ProjectHub relies on environment variables stored in a `.env` file to manage configuration settings. These variables ensure that sensitive information is kept secure and allow flexibility in different deployment environments.

### • Key Environment Variables

- **FIREBASE\_PROJECT\_ID** – Specifies the Firebase project ID.
- **FIREBASE\_PRIVATE\_KEY** – Stores the private key for Firebase authentication.
- **FIREBASE\_CLIENT\_EMAIL** – Email associated with Firebase Admin SDK.
- **SMTP\_HOST** – Mail server address for sending email notifications.
- **SMTP\_PORT** – Port for email communication.
- **SMTP\_USER** – Email username for authentication.
- **FRONTEND\_URL** – URL of the frontend application for handling redirects.
- **JWT\_SECRET** – Secret key for JSON Web Token (JWT) authentication.

- **Security Considerations**

- The .env file should never be committed to version control (including in .gitignore).
- Use secret management tools (e.g., Vercel Environment Variables) when deploying to production.
- Rotate sensitive keys periodically to enhance security.

<>	FRONTEND_URL All Environments	👁	*****
<>	FIREBASE_PRIVATE_KEY All Environments	👁	*****
<>	NODE_ENV All Environments	👁	*****
<>	FIREBASE_CLIENT_EMAIL All Environments	👁	*****
<>	FIREBASE_PROJECT_ID All Environments	👁	*****
<>	FIREBASE_DATABASE_URL All Environments	👁	*****
<>	JWT_SECRET All Environments	👁	*****

## 2. Frontend Requirements

ProjectHub's frontend uses modern web technologies, providing a fast, responsive, and user-friendly experience. Below are the key technical requirements and dependencies.

### Frontend Technology Stack

- **Framework:** React (18.3.1) – Manages UI components and application state.
- **Routing:** React Router (6.28.0) – Handles navigation between pages.
- **Styling:** Tailwind CSS (3.4.15) – Provides a modern utility-based styling approach.
- **State Management & Caching:** SWR (2.2.5) – Optimizes data fetching and caching.
- **Authentication & Database:** Firebase SDK (11.1.0) – Integrates authentication and database operations.

### Frontend Dependencies

The frontend relies on various dependencies to enhance functionality and improve the user experience.

- **UI & Styling**
  - **Tailwind CSS (tailwindcss)** – Styling framework for responsive design.

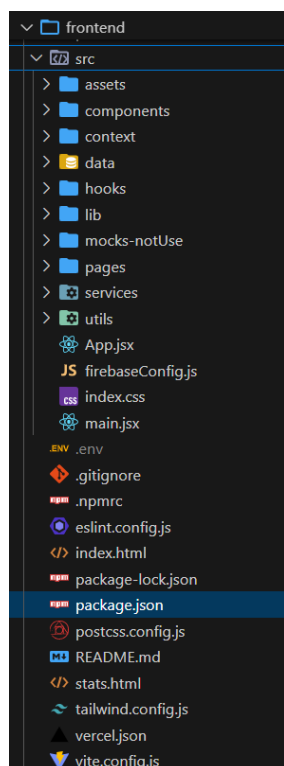


- **Lucide React (lucide-react)** – Provides icons used throughout the interface.
- **React Toastify (react-toastify)** – Displays system notifications.
- **React Tooltip (react-tooltip)** – Adds interactive tooltips for better user guidance.
- **Forms & Data Handling**
  - **React Hook Form (react-hook-form)** – Manages form validation and submission.
  - **Date-fns (date-fns)** – Handles date formatting and calculations.
  - **ExcelJS (exceljs) & XLSX (xlsx)** – Allows importing/exporting data to Excel files.
- **API & Backend Communication**
  - **Axios (axios)** – Handles HTTP requests between the frontend and backend.
  - **Firebase SDK (firebase)** – Manages authentication and integrates with Firestore.
- **Performance & Optimization**
  - **SWR (swr)** – Provides efficient data fetching and automatic revalidation.
  - **Tailwind Merge (tailwind-merge)** – Ensures better styling consistency.
  -

## Development & Build Tools

For development and deployment, ProjectHub's frontend is optimized with modern tools.

- **Vite (5.4.10)** – Enables fast builds and live reloading.
- **ESLint (9.13.0)** – Ensures code quality by enforcing best practices.
- **PostCSS (8.4.49)** – Optimizes CSS processing.
- **Autoprefixer (10.4.20)** – Adds vendor prefixes for browser compatibility.



```
backend > package.json > {} dependencies
1  {
2    "name": "backend",
3    "version": "1.0.0",
4    "main": "server.js",
5    "engines": {
6      "node": "20.x"
7    },
8    "scripts": {
9      "test": "echo \"Error: no test specified\" && exit 1",
10     "start": "node server.js",
11     "dev": "nodemon server.js",
12     "vercel-build": "echo \"Building for production\""
13   },
14   "keywords": [],
15   "author": "",
16   "license": "ISC",
17   "dependencies": {
18     "axios": "1.7.9",
19     "bcryptjs": "2.4.3",
20     "compression": "1.7.4",
21     "cors": "2.8.5",
22     "dotenv": "16.4.7",
23     "express": "4.21.1",
24     "express-rate-limit": "7.1.5",
25     "firebase-admin": "^13.0.2",
26     "helmet": "7.1.0",
27     "mustache": "4.2.0",
28     "nodemailer": "6.9.16"
29   },
30   "devDependencies": {
31     "nodemon": "3.0.2"
32   },
33   "description": "Backend API for Final Project Management System"
34 }
35
```

## Environmental Variables

The frontend of ProjectHub relies on environment variables for configuration, ensuring flexibility in different deployment environments. These variables are stored in a `.env` file and used throughout the application.`

- **Key Environment Variables**
  - **VITE\_API\_BASE\_URL** – Specifies the backend API URL.
  - **VITE\_FIREBASE\_API\_KEY** – API key for Firebase authentication and database access.
  - **VITE\_FIREBASE\_AUTH\_DOMAIN** – Domain for Firebase authentication.
  - **VITE\_FIREBASE\_PROJECT\_ID** – Unique identifier for the Firebase project.
  - **VITE\_FIREBASE\_STORAGE\_BUCKET** – Storage location for Firebase-hosted files.
  - **VITE\_FIREBASE\_MESSAGING\_SENDER\_ID** – Messaging sender ID for Firebase notifications.
  - **VITE\_FIREBASE\_APP\_ID** – Firebase application identifier.
- **Security Considerations**
  - The `.env` file must not be committed to version control (`.gitignore` should exclude it).
  - Sensitive environment variables should be managed securely in Vercel Environment Variables when deploying to production.
  - Periodic rotation of API keys and credentials is recommended to enhance security.
- **Firebase Configuration** – The frontend initializes Firebase using the stored environment variables. The configuration is managed in `firebaseConfig.js`, where Firebase services like Firestore and authentication are integrated.

<>	VITE_API_BASE_URL All Environments	⦿ .....
<>	VITE_FIREBASE_APP_ID All Environments	⦿ .....
<>	VITE_FIREBASE_MESSAGING_SEND... All Environments	⦿ .....
<>	VITE_FIREBASE_STORAGE_BUCKET All Environments	⦿ .....
<>	VITE_FIREBASE_PROJECT_ID All Environments	⦿ .....
<>	VITE_FIREBASE_AUTH_DOMAIN All Environments	⦿ .....
<>	VITE_FIREBASE_API_KEY All Environments	⦿ .....

```
frontend > src > JS firebaseConfig.js
1  import { initializeApp } from "firebase/app";
2  import { getFirestore } from "firebase/firestore";
3
4  const firebaseConfig = {
5    apiKey: import.meta.env.VITE_FIREBASE_API_KEY,
6    authDomain: import.meta.env.VITE_FIREBASE_AUTH_DOMAIN,
7    projectId: import.meta.env.VITE_FIREBASE_PROJECT_ID,
8    storageBucket: import.meta.env.VITE_FIREBASE_STORAGE_BUCKET,
9    messagingSenderId: import.meta.env.VITE_FIREBASE_MESSAGING_SENDER_ID,
10   appId: import.meta.env.VITE_FIREBASE_APP_ID,
11 };
12
13 // Initialize Firebase
14 const app = initializeApp(firebaseConfig);
15 const db = getFirestore(app);
16
17 export { db };
18
```

## Deployment

The frontend of ProjectHub is hosted on Vercel, which provides automatic deployments, serverless functions, and continuous integration.

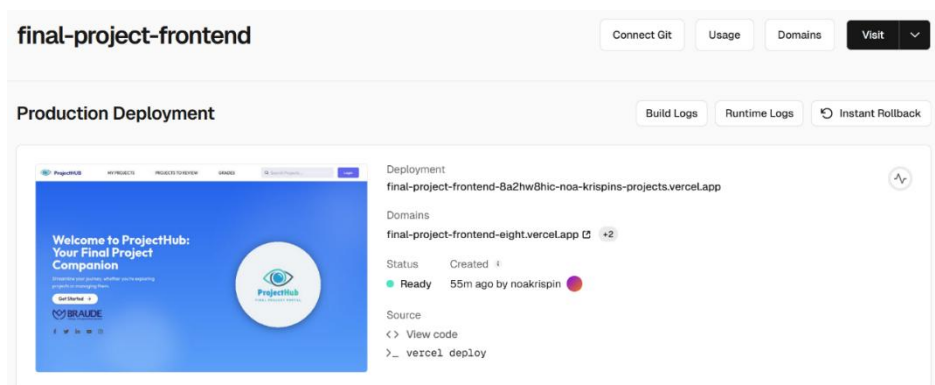
- **Hosting Provider** - Vercel. The frontend is deployed using Vercel, offering scalability, optimized performance, and a built-in CDN.
- **Vercel Deployment Configuration**

The deployment configuration is managed via a `vercel.json` file which specifies:

- **Framework:** Vite
- **Build Command:** `npm run build`
- **Output Directory:** `dist`
- **Routes Configuration:**
  - Asset catching for improved performance (`max-age=31536000`, `immutable`).
  - Filesystem handling for serving static files.
  - Catch-all routing to `index.html` for Single Page Application (SPA) support.

```
frontend > vercel.json > ...
1  {
2    "framework": "vite",
3    "buildCommand": "npm run build",
4    "outputDirectory": "dist",
5    "routes": [
6      {
7        "src": "/assets/(.*)",
8        "headers": { "cache-control": "public, max-age=31536000, immutable" },
9      },
10     {
11       "handle": "filesystem"
12     },
13     {
14       "src": "/.*",
15       "dest": "/index.html"
16     }
17   ]
18 }
```

- **Deployment Trigger**
  - **Automatic Deployments** – Any commits pushed to the main branch automatically trigger a new deployment on Vercel.
  - **Instant Rollback** – If an issue is detected in a deployment, Vercel allows instant rollbacks to a previous stable version.
  - **Custom Domains** – The application is accessible through Vercel's generated URLs, with the option to configure custom domains.
  - **Build & Deployment Logs** – Vercel provides detailed logs for debugging any issues during deployment.



### 8.2.3 System Administration

- **Introduction**

System administration in ProjectHub is essential for managing users, projects, and evaluation forms. Project coordinators (admins) have elevated permissions, allowing them to oversee system functionality, assign roles, and maintain data integrity. This section outlines the administrative tools available to project coordinators and how they can efficiently manage the platform.

- **User Role Management**

Admins can modify user roles, ensuring appropriate access levels within the platform. Supervisors can be promoted to project coordinators, and admins can also remove users if necessary (Figure 27).

**Editing a User's Role**

1. Navigate to Admin Dashboard → Management → User Management.
2. Locate the user in the list.
3. Click the Edit Icon next to the user.
4. Select the new role (Supervisor or Project Coordinator).
5. Click Save to confirm the changes.

**Deleting a User**

1. Navigate to Admin Dashboard → Management → User Management.
2. Locate the user in the list
3. Click the Trash Icon next to the user.
4. A confirmation popup will appear.
5. Click Confirm to permanently remove the user.

**Note:** Only project coordinators can modify user roles and remove accounts. Supervisors cannot make administrative changes.

- **Managing Projects**

Admins oversee all projects on the platform. Initially, the system does not contain any projects, so the admin must upload project data using an Excel file

**Uploading Initial Project Data**

Admins have two options to upload initial project data (Figure 25):

- **Option 1: From the Projects Page**

1. Navigate to Admin Dashboard → Projects
2. Click "Upload Excel file"
3. Follow the Excel file format requirements.
4. The projects will load, and the admin will be redirected to the Projects page where all projects will be displayed.

- **Option 2: From the Upload Files Page:**

1. Navigate to Admin Dashboard → Upload Files.
2. Follow the Excel file format requirements.
3. The projects will load, and the admin will be redirected to the Projects page where all projects will be displayed.

**Deleting Projects** (Figure 21).

- A single project can be deleted by clicking the Trash Icon in the table.
- Projects that are in Part B can be deleted in bulk by clicking the Delete Button in the Part B tab, with the option to export them to Excel before deletion.

**Important:** Deleting a project permanently removes all related data, including student assignments and evaluations.

- **Managing Evaluation Forms**

Admins can modify evaluation forms used for project grading. They can add or remove questions, ensuring the forms align with project assessment needs. (Figure 28)( Figure 29).

**Editing an Evaluation Form**

1. Navigate to Admin Dashboard → Management → Forms Management.
2. Click on a form row to open it.
3. Click "Edit Form" to modify the fields.
4. Add, remove, or change form questions.
5. Click Save to confirm updates.

**Note:** Form changes apply system-wide and affect all evaluations based on that form.

- **System Access & Security Considerations**

To maintain security and ensure platform integrity, the following rules are enforced:

- Only users with braude.ac.il emails can register.
- Passwords must meet security requirements:
  - At least 8 characters long
  - Must include an uppercase letter, number, and special character
- Admins should periodically review user roles and remove inactive accounts.

## 8.2.4 Debugging and Logs

- **Introduction to Debugging in ProjectHub**

Debugging is a crucial part of maintaining the stability and reliability of ProjectHub. As a platform that integrates both frontend and backend technologies, various issues may arise, including UI bugs, API failures, and database inconsistencies. Efficient debugging ensures that problems are identified and resolved quickly, minimizing disruptions for users.

ProjectHub's debugging process involves:

- **Frontend Debugging** to identify rendering issues, navigation errors, and API response failures.
- **Backend Debugging** to handle API request failures, authentication issues, and database inconsistencies.
- **System Logging** to track errors, monitor performance, and analyze user interactions.

- **Frontend Debugging**

Frontend debugging in ProjectHub involves identifying issues with UI rendering, navigation, and API communication. Developers primarily use browser developer tools to detect and resolve errors.

1. **Using Browser Developer Tools (DevTools)**

To inspect frontend issues, project coordinators and developers can use Chrome DevTools:

- **Console Tab:** Displays JavaScript errors, warnings, and logs that help diagnose issues.
- **Elements Tab:** Allows inspecting and modifying the HTML and CSS structure in real time.
- **Network Tab:** Tracks API calls and responses to detect failures.

2. **Debugging API Requests**

ProjectHub communicates with the backend through API requests. If an API fails, the following steps help diagnose the issue:

- Open Chrome DevTools (F12 or right-click → Inspect).
- Navigate to the Network Tab
- Check failed API requests (highlighted in red).
- Click on a request to examine:
  - **Status Code** (e.g., 400 Bad Request, 500 Internal Server Error).
  - **Response Body** (for error messages from the backend).

- **Request Headers** (to verify correct authentication tokens).

### 3. Handling UI Issues

- If a component does not render properly, inspect the React component tree using React Developer Tools.
- Check the state variables in React Developer Tools to ensure they hold the expected values.
- Validate that Tailwind CSS classes are applied correctly.

- **Backend Debugging**

Backend debugging in ProjectHub focuses on ensuring the API, authentication, and database interactions work correctly. The primary tools used include server logs, API response monitoring, and database validation.

#### 1. Monitoring API Requests and Errors

- All backend requests generate logs to help detect failures.
- Common API error codes and their causes:
  - 400 Bad Request → Missing or incorrect parameters in the request.
  - 401 Unauthorized → User is not authenticated or lacks permissions.
  - 403 Forbidden → User does not have access to the requested resource.
  - 404 Not Found → The requested endpoint or resource does not exist.
  - 500 Internal Server Error → An issue occurred on the server (e.g., database failure).
- Debugging Steps:
  - Open browser DevTools → Network Tab to check the failing API request.
  - Use Postman or cURL to manually test API endpoints with sample data.
  - Check Vercel logs for server-side error messages and stack traces.

#### 2. Debugging Authentication and Authorization Issues

- If users experience login failures, check:
  - Firebase Authentication logs for failed login attempts.
  - JWT token expiration to verify valid session tokens.
  - Backend API request headers to confirm the authentication token is sent correctly.

- If user roles are incorrect:
  - Ensure user role assignments are correctly stored in Firestore.
  - Verify that role-based permissions allow or restrict access properly.

### 3. Handling Database Issues

- If the data is missing or incorrect:
  - Check Firestore Database for inconsistencies.
  - Use the Firebase Console to view stored documents.
  - Review recent API updates to ensure data is properly written.

- **System Logging and Monitoring**

System logs play a crucial role in tracking errors, warnings, and performance issues. Logging mechanisms help administrators identify problems before they affect users.

#### 1. Firebase Logs for Authentication & Database

- Authentication events (login, registration failures).
- Firestore queries (read/write operations).
- Security rules violations (unauthorized data access attempts).
- Logs can be accessed via Firebase Console → Logs Section.

## 8.2.5 Future Maintenance

This chapter outlines key areas for ongoing maintenance and potential future improvements for the ProjectHub platform. Ensuring long-term stability and scalability requires continuous monitoring, updates, and enhancements.

### 1. System Monitoring and Error Handling

To maintain ProjectHub's reliability, the system should be monitored for performance, security vulnerabilities, and user-reported issues.

- **Automated Error Logging:** Continue using Firebase logs for authentication, database operations, and security rule violations.
- **API Monitoring:** Regularly check API response times and failures, ensuring backend services function as expected.
- **User Feedback Integration:** Implement a structured process for collecting and addressing user-reported issues.

### 2. Security Updates and Authentication Management

Security is critical to protecting user data and preventing unauthorized access.



- **Firestore Authentication Enhancements:** Regularly review login attempts and potential account security breaches.

### 3. Database Maintenance and Optimization

The ProjectHub platform relies on Firestore as its NoSQL database. Regular optimizations can improve query performance and prevent data inconsistencies.

- **Firestore Query Optimization:**
  - Regularly review database queries to minimize read and write costs.
  - Use Firestore indexes efficiently to improve search performance.
- **Data Archiving Strategy:** implement an archiving solution for old projects and evaluations to reduce Firestore costs and improve performance.

### 4. Feature Enhancements and Scalability

As ProjectHub grows, new features may be required to improve user experience and functionality.

- **Automated Reminder Scheduling:** Revisit the implementation of scheduled reminders to allow coordinators to set reminders for future dates.
- **Integration with College Systems:** Explore direct API integration with the college's grading system instead of manual grade exports.
- **System Expansion:** Continue to build parts 1 and 3 of the system.

### 5. Continuous Testing and Deployment Strategy

Regular testing and structured deployment strategies can prevent regressions and maintain system stability.

- **Automated Testing:** Implement frontend unit tests.
- **Deployment Best Practices:** Continue using Vercel for frontend deployment with version control.

### 6. Long-Term Maintenance Considerations

- **Documentation Updates:** Keep technical documentation, user guides, and admin manuals up to date with system changes.
- **Support and Training:** Provide periodic training sessions or materials for new project coordinators and supervisors.

## References

1. BrowserStack (2024). *Web Application Development: Process, Tools, & Examples*. [BrowserStack](#).
2. Nielsen, J. (1993). *Usability Engineering*. London, England: Academic Press.
3. *Memorial University of Newfoundland. Research Project Management Guide for Researchers. Section 7.1.3.* [Research-Project-Management-Guide-January-2018.pdf](#)
4. *Nwanakezie, H., & Ogonu, I. (2021). Task Development Procedures for Effective Educational Management.* [Based learning environment.pdf](#)