



People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research



Echahid Hamma Lakhdar University – El Oued

Faculty of Exact Sciences

Department of Computer Science

Final Year Project Report

In partial fulfillment of the requirements for the degree of:

Bachelor's Degree

Specialty: Computer Science

Realized by:

BEY Heyam

ZOUARI Ahmed Lina

-ZIAD Rounek

Under the supervision of: -

SOLTANI Khaled -

THEME

Development of a Web-Based Final Year Project Management System for

Licence and Master's Programs

Publicly defended on: 25 /05/2025 before the Jury composed of:

The chair of the jury	د. مفتاح شرفه الدين	University of El Oued
Under the supervision of	د. سلطانزي خالد	University of El Oued
Examiner	د. بوحامد محمد منير	University of El Oued

Academic Year: 2024/2025

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dedication

We want to thank God for blessing us with good luck.

We are deeply grateful to our parents for standing by us throughout this journey and for providing a supportive environment for our studies.

We also want to thank ourselves for persevering through it all.

Lastly, we extend our sincere thanks to our supervisor SOLTANI Khaled for his guidance and support.

Table Of Content

Dedication	3
الملخص	
General Introduction	1
<i>Chapter 01: Problem Context and Motivation</i>	<i>.....</i>
Introduction.....	3
1. Current Challenges.....	3
1.1. Problem Statement	3
1.2. Workflow for Topic Submission and Approval.....	3
1.3. Lack of Flexible Evaluation Criteria	4
1.4. Sub-optimal Jury Assignment Assigning.	4
1.5. Lack of Project Status Visibility	4
1.6. Complicated Documentation Process	4
1.7. Data Fragmentation.	4
2. Motivation.....	5
2.1. Digital Transformation Needs	5
2.2. Transparency and Fairness	5
2.3. Data-Driven Decision Making	5
2.4. User Experience Expectations	5
2.5. Resource Efficiency	5
2.6. Standardization with Flexibility.....	5
2.7. Reusability and Adaptability.....	6
3. Objectives and Proposed Solution.....	6
3.1. Problem: Workflow for Topic Submission and Approval	6
3.2. Problem: Lack of Flexible Evaluation Criteria	6
3.3. Problem: Lack of Project Status Visibility	6
3.4. Problem: Complicated Documentation Process.....	6
4. Theoretical Foundations and Related Work.....	7
4.1. Role-Based Access Control (RBAC)	7

4.2. Benefits of Role-Based Access Control (RBAC).....	7
Conclusion	9
<i>Chapter 02</i>	3
<i>Modeling and Design</i>	3
Introduction.....	11
1. UML Definition	11
1.1. Use Case Diagram.....	11
1.2. Activity Diagram.....	11
1.3. Sequence Diagram.	11
1.4. Class Diagram.	11
2. Use case diagrams Section.....	12
2.1. Admin use-case.....	12
2.2. Super-admin use-cases	13
2.4. The relationship between the users... خطأ! الإشارة المرجعية غير معرّفة.	
3. Sequence Diagrams Section	16
4.1. The activity diagrams	22
5. Class Diagram	34
5.1. Relational Model.....	35
Conclusion	36
<i>Chapter 03</i>	11
<i>Implementation</i>	11
Introduction.....	36
1. Components of the work environment	36
1.1. Languages.....	36
1.2. Frameworks and Libraries	36
1.3. Tools and Utilities	37
1.4. Other Tools.....	38
2-The Workflow of Our System: MERN Stack Explained.....	38
2.1. Frontend (React).....	39
2.2. Backend (Node.js + Express)	39
2.3. Database (MongoDB)	39
2.4. How it works	39

3.Database Schemas	40
4.The Main Interfaces of the Application	43
Figure 3.26: Pages In Dark Mode	62
Conclusion	63
General Conclusion	62
References	63

List of Figures

N°	THE ADDRESS	PAGE
01	Figure 2.1: Admin's use-case diagram	13
02	Figure 2.2: Super Admin use-case diagram	14
03	Figure 2.3: Teacher use-case diagram	15
04	Figure 2.4: Sign-in sequence diagram	17
05	Figure 2.5: Sign-out sequence diagram	18
06	Figure 2.6: Create Teachers/Admins accounts sequence diagram	19
07	Figure 2.7: Forgot password sequence diagram	20
08	Figure 2.8: Reset Password sequence diagram	21
09	Figure 2.9: Change Password Activity Diagram	23
10	Figure 2.10: Update Email Activity Diagram	24
11	Figure 2.11: Submit Topics Diagram	25
12	Figure 2.12: Edit/Delete Topic diagram	26
13	Figure 2.13: Submit Evaluation Form diagram	27
14	Figure 2.14: Edit/Delete Evaluation Form Activity diagram	28
15	Figure 2.15: Delete User Activity Diagram	29
16	Figure 2.16: Update Teacher's Academic Level	30
17	Figure 2.17: Change Topic's status Activity diagram	31
18	Figure 2.18: Generate Juries Activity Diagram	33
19	Figure 2.19: Actions related to the generated juries	34
20	Figure 2.20: Class Diagram For Users	35
21	Figure 3.1: MERN Stack work flow	38
22	Figure 3.2: Home page (before signing in)	43
23	Figure 3.3: Sign-in pop up	44
24	Figure 3.4: Profile Page	44
25	Figure 3.5: Update Email Page	45
26	Figure 3.6: Change Password Page	45
27	Figure 3.7: Forgot-Password Page	46
28	Figure 3.8: Reset Password Page	46
29	Figure 3.9: Access Denied Page	47
30	Figure 3.10: Super-admin's Dashboard	48
31	Figure 3.11: Super-admin User management	49
32	Figure 3.12: Create Admin Page	49
33	Figure 3.13: Admin's Dashboard	50
34	Figure 3.14: Create Teacher Page	51
35	Figure 3.15: Topics Management Page	51
36	Figure 3.16: Topics Status Window	52
37	Figure 3.17: Jury Management Page	53
38	Figure 3.18: Chosen Juries Schedule Page	54
39	Figure 3.19: Teacher's Dashboard	55
40	Figure 3.20: Topics Management Page For Teachers	56

41	Figure 3.21: Submit New Topic Page	57
42	Figure 3.22: Edit / Delete Topic Page	58
43	Figure 3.23: Teacher's Evaluations Forms	59
44	Figure 3.24: Submit Evaluation Form	60
45	Figure 3.25: Edit Evaluation Form	61
46	Figure 3.26: Pages In Dark Mode	62

الملخص

في العديد من المؤسسات الاكاديمية، لا تزال إدارة مشاريع نهاية الدراسة تتم بشكل يدوي، مما يؤدي إلى ضعف الكفاءة وإهدار كبير للوقت. يهدف هذا المشروع إلى تقديم نظام إلكتروني لإدارة مشاريع نهاية الدراسة كبديل للنظام اليدوي الحالي، وذلك من أجل تحسين وتحديث سيرورة العمل. تم تصميم هذا النظام لدعم الاساتذة والاداريين في تنظيم مشاريع نهاية الدراسة لطلبة الليسانس والماستر بجامعة الشهيد حمة لخضر - الوادي، كلية العلوم الدقيقة.

يجمع النظام كل الاطراف في منصة واحدة عبر الانترنت، مما يُسهل عملية التسيير. كما يتيح للاساتذة تقديم مشاريعهم وتقييم طلابهم مباشرة عبر المنصة، ويُمكن الاداريين من تعيين لجان المناقشة (التحكيم) بشكل تلقائي وسهل، مما يُسهل في تسريع وتبسيط الاجراءات.

كلمات مفتاحية: مشاريع نهاية السنة، منصة رقمية، الويب.

Abstract

In our academic systems, the Final Year Project Management System is usually done manually, which often leads to inefficiency and significant time consumption. That is why, in this project, we are proposing a web-based Final Year Project Management System to replace the current manual one. The main goal of this project is to help professors and administrators manage Final Year Projects for both Licence and Master's students at Echahid Hamma Lakhdar University – El Oued, Faculty of Exact Sciences. Our system will also bring everyone together on one online platform, making it easier to manage Final Year Projects. It will allow professors to easily submit their graduation projects and evaluate their students directly on the platform. It will enable administrators to automatically assign juries within the system easily.

.Keywords: End of year projects, digital platform, web

Résumé

Dans notre système académique, la gestion des projets de fin d'études (PFE) se fait généralement de manière manuelle, ce qui entraîne souvent une inefficacité et une perte de temps considérable. C'est pourquoi, dans le cadre de ce projet, nous proposons un système de gestion des projets de fin d'études en ligne pour remplacer le système manuel actuel.

L'objectif principal de ce projet est d'aider les enseignants et les administrateurs à gérer les projets de fin d'études des étudiants de Licence et de Master à l'Université Echahid Hamma Lakhdar – El Oued, Faculté des Sciences Exactes. Notre système réunira tous les acteurs sur une seule plateforme en ligne, facilitant ainsi la gestion des projets de fin d'études.

Il permettra aux enseignants de soumettre facilement leurs projets de fin d'études et d'évaluer leurs étudiants directement sur la plateforme. Il offrira également aux administrateurs la possibilité d'assigner automatiquement les jurys au sein du système, de manière simple et efficace.

Mots clés : Projets de fin d'année, plateforme numérique, web.

General Introduction

General Introduction:

Accessing information in today's world has taken a leap with the development in technology, which has given way to online systems becoming the new management systems that people rely on. The need for online systems has increased rapidly during these past years since they have been doing a very good job at eliminating the need for hard-copies, physical space, and repeating manual protocols. EHLU also needs to benefit from the use of a time-efficient online management system. The exchange of information among FYP administrators, lecturers, supervisors and examiners needs effective communication to ensure there is no misunderstanding or lack of information that could cause delays in completing and managing projects. Thus, the FYP Online management System is ideal since it provides the best solutions for previous FYP management system problems. This system is proposed due to the need to enhance the process and improve management system efficiency. In this project, a web-based software solution will be introduced to provide a comfortable environment for FYP management. Since making academic administration digital helps make things easier and more organized, while still following the rules of the institution. When tasks like submitting topics, approving them, choosing jury members, and doing evaluations are done online, all the information is stored in one place. This makes it easier for the right people to find what they need, make better decisions, and keep everything running smoothly. This project explains how the system works technically, and also looks at how it could help Professors and the organization as a whole. By looking at how the system is designed and how people use it, we show that digital tools can improve both education at EHLU generally, and the Faculty of Exact Sciences specifically.

The paper is divided into three parts, each covering both theory and practical implementation in detail.

Chapter 1: Problem Context and Motivation: introduces the main problems in the current workflow for managing graduation projects, such as delays and lack of organization. It also presents the idea of a digital system as a simple and effective solution to improve the overall process.

Chapter 2: Modeling and Design: In this chapter, we started by creating a plan for the project and designing diagrams that illustrate the system's workflow, the services it provides, and how it interacts with the users.

Chapter 3: Implementation: In this chapter, we present the results of the project, including the tools used for implementation and an explanation of how they were applied.

Chapter 01:
Problem Context and Motivation

Introduction

Graduation projects play a major and important role in higher education at our university. They represent the final outcome of what we, as students, have learned and the skills we have developed throughout our studies. These final year projects (FYPs) usually take several months to complete and involve many people in the process before they are officially assigned to students. Managing these projects usually includes several steps: First, professors propose topics/thesis, which then need to be approved and validated by administrators. Once approved, the topics are assigned to students. After that, the administrators are responsible for forming a jury to evaluate each project. This jury must be selected carefully based on different conditions, such as academic level, group, and gender diversity, especially for Master and Licence students in the Faculty of Exact Sciences — specifically in the Computer Science department. Supervisors are also involved throughout the process. They guide their students and later evaluate their progress and their final work, including how well the student built and designed the project. However, the current workflow is not well-organized. Many of the tasks are still done manually, and proposed topics are often submitted through separate emails, which can easily get lost or overlooked. This leads to delays, confusion, or even the need for re-submissions sometimes. For these reasons, we decided to develop this project. Our goal is to create an online system that simplifies and organizes the entire workflow. The system will help manage topic proposals, approvals, jury assignments, and evaluations — all in one place. This will reduce errors, save time, and make the graduation project process much smoother and more efficient for everyone involved.

1. Current Challenges:

1.1. Problem Statement: Final Year Project management has been in a more traditional way for all these years. It is done manually where all the registration, allocation and markings are in hard-copies and forms and then managed by FYP coordinators. While the coordinators are doing a great job, this management system has several disadvantages. To be more exact of the problems that this manual management system face. we mention some of them:

1.2. Workflow for Topic Submission and Approval: The process of submitting and approving/validating topics often relies on paper forms or

on emails, which are then manually collected. This outdated method causes delays and it increases the chances of errors that can lead to the loss of those topics. Additionally, approval committees don't have proper tools to evaluate topics consistently, meaning decisions can vary and lack uniformity. This reliance on paper and manual processes makes it harder to manage and track the approval process efficiently.

1.3. Lack of Flexible Evaluation Criteria : Professors may feel limited by rigid evaluation standards when grading students' projects. This can make it so much harder for them to provide feedback that reflects the unique aspects of each project because without a flexible system evaluations might not capture the full scope of a student's work, and professors may feel pretty much restricted in offering more tailored or personalized feedback. Additionally, a lack of clear tracking systems can make it difficult to ensure consistent evaluations across all projects

1.4. Sub-optimal Jury Assignment Assigning: a balanced jury for project evaluation is a challenging task because it relies on different factors like the supervisor's academic level, gender, and the jury members' expertise. While manual methods do try to consider these factors, it's much harder to apply the rules correctly. This can result in an uneven distribution of work and take up a lot of time, as manually matching the right jury members to the right projects is both confusing and slow.

1.5. Lack of Project Status Visibility: Admins and department heads find it hard to track the status of all projects. This makes it harder to plan resources and catch issues early. Also, supervisors and other Professors don't have a clear, shared place to see deadlines or requirements.

1.6. Complicated Documentation Process: Topics proposals are usually collected through email or in person, making it hard to organize, find, and access past documents. This also makes it difficult to categorize and search for older projects.

1.7. Data Fragmentation: Project information is often scattered across different systems, like spreadsheets in different departments, email threads, and personal notes. This fragmentation makes it hard to evaluate the overall progress of projects, which could help improve curriculum planning and resource management.

These challenges cause delays in administration, unfair evaluations, a poor management and a lot of wasted time that could be spent on improving education.

2. Motivation

There are several reasons why this research matters and why we need to tackle these challenges:

2.1. Digital Transformation Needs: More and more, universities and educational institutions are realizing the need to modernize their administrative processes. Going digital can help improve efficiency and make services better for everyone involved. Graduation project management, with its many moving parts and people, is one area where this change could be especially helpful.

2.2. Transparency and Fairness: With a growing focus on fairness in education, there's increasing pressure to make project evaluations more transparent. Digital systems can help by standardizing the process, making sure evaluation criteria are clear, and reducing bias, which can build trust among students and faculty.

2.3. Data-Driven Decision Making: A lot of universities are looking to use data to improve continuously. By moving project management online, we can collect useful data on evaluation methods, and student outcomes. This info can help improve the curriculum, allocate resources better, and improve the overall quality of education.

2.4. User Experience Expectations: Today's admins and faculty workers are used to digital interfaces; thanks to all the consumer apps they use daily. If they have a bad experience with a system, it can hurt their engagement and satisfaction with the process. A user-friendly platform can make things smoother for everyone and make it easier for them to adopt the system.

2.5. Resource Efficiency: With the rising cost pressures in higher education, it's important to make the most of available resources. A digital final year project management system can help reduce administrative work for faculty and staff, so they can spend more time focusing on teaching and learning, rather than getting bogged down with paperwork.

2.6. Standardization with Flexibility: It's important to have a set way of doing things, but also allow departments to make changes to fit their

needs. A digital system can make things uniform, but still give departments the option to adjust it as needed.

2.7. Reusability and Adaptability: If we create a system for managing graduation projects, it could be adapted for other educational activities too, like internships, research projects, or group learning. Using a modular approach means it could be easily adapted for different situations, even beyond the original use.

These reasons highlight how a comprehensive digital solution can not only solve immediate problems but also support long-term goals like improving quality, ensuring fairness, and increasing efficiency.

3. Objectives and Proposed Solution:

The main goal of this research was to tackle the challenges we identified by designing, implementing, and evaluating a digital solution. Specifically, we aimed to address each problem with a targeted solution:

3.1. Problem: Workflow for Topic Submission and Approval

Objective: Build an efficient and user-friendly platform that allows professors to easily submit, track, and manage graduation project topics in one centralized place, reducing manual work and improving clarity.

3.2. Problem: Lack of Flexible Evaluation Criteria

Objective: Create a flexible evaluation framework that allows professors to adjust criteria to suit different project types while maintaining overall consistency. Include tools to help track evaluations and feedback for transparency and fairness.

Problem: Suboptimal Jury Assignment
Objective: Develop a smart, automated system that helps generate diverse and balanced evaluation committees by considering key factors such as academic level, groups, gender diversity and other conditions to ensure fairness and equal workload distribution.

3.3. Problem: Lack of Project Status Visibility

Objective: Build a dashboard or tracking tool that allows admins, department heads, and professors to monitor project progress, and stay informed about all related tasks in one shared space.

3.4. Problem: Complicated Documentation Process

Objective: Implement a centralized digital repository for submitting, storing, and retrieving topic proposals and related documents. It should support easy organization, categorization, and search functionality and filtering for both master and license final year projects.
Problem: Data Fragmentation
Objective: Create a unified digital system that brings

together all project-related information in one place. This will make it easier to track progress, support better curriculum planning, and help departments manage their resources more effectively.

These objectives serve as a roadmap for our development process and a way to measure how well we've done. Our goal is to create a solution that provides real value to educational institutions and helps us better understand academic administration. By aligning these objectives with the challenges and motivating factors we identified earlier, we're ensuring that our research is focused on solving real problems in education, not just creating new technology for the sake of it.

4. Theoretical Foundations and Related Work:

Before we dive into the technical stuff like designs and diagrams, we need to understand the ideas and theories that guided our approach. Since our system touches areas like access control, academic workflows, and educational technology, This background helped us make smarter decisions and create a system that actually fits real-world academic environments.

4.1. Role-Based Access Control (RBAC)

RBAC is an access control method for managing access to resources by assigning permissions based on a user's role within an organization.

In our system, we identified three primary roles: "Teachers, Administrators, and Super Administrators" each with specific permissions and responsibilities. RBAC provides a scalable and maintainable approach to authorization that aligns naturally with organizational structures in educational institutions.

4.2. Benefits of Role-Based Access Control (RBAC):

Better Security: RBAC can dramatically reduce the effect of human vulnerabilities because many access privileges that could be exploited are eliminated.

Less Work for Admins: With role-based access control, an admin can easily change access privileges for individuals or entire teams all at once. This not only saves time but also ensures that users get the access they need when they need it— without manual, repetitive work.

Improved Operational Efficiency: By assigning roles instead of individual permissions, RBAC simplifies day-to-day operations. New users can be on boarded quickly by assigning them to a role, and system changes are easier to manage without disrupting the workflow.

Clear Responsibility and Accountability: Since each role comes with specific permissions, it's easier to track who accessed what and when. This creates a clear trail of responsibility, making it simpler to audit user behavior and spot unusual activity when needed.

Conclusion

In Chapter One, we looked at the main challenges universities face when managing graduation projects, like issues with submitting topics, flexibility in evaluations, assigning juries, tracking project status, and handling documents. These problems often arise from outdated and fragmented systems, which slow things down and make things less consistent so in order to tackle these issues, we've set clear goals for creating a digital solution that simplifies topic management, automates jury assignments, moves evaluations online, centralizes project data, and ensures secure access.

Chapter 02:
Modeling and Design

Introduction:

After completing the initial study and understanding the full set of tasks that the system will provide to users, this chapter uses UML language to present the modeling and design phase. This step is considered crucial for showcasing the progress of the project by incorporating new information gathered through detailed study and real-world observation. It helps us gain a deeper and clearer understanding of the system's nature and requirements. To achieve this, we'll present various diagrams that are commonly used in modeling and designing software systems.

1. UML Definition:

Unified Modeling Language (UML) is a standardized visual modeling language that is a versatile, flexible, and user-friendly method for visualizing a system's design. Software system artifacts can be specified, visualized, built, and documented with the use of UML. [1]

- We use UML diagrams to show the behavior and structure of a system.
- UML helps software engineers, businessmen, and system architects with modeling, design, and analysis.

We will use UML language in this chapter to present three main diagrams:

1.1. Use Case Diagram: This diagram simply describes the activities and tasks performed by users or external systems, as well as the different interactions they have with the system being developed.

1.2. Activity Diagram: This diagram simply outlines the sequence of activities and interactions that take place within the system, showing how data flows and how control is managed throughout these activities.

1.3. Sequence Diagram: This diagram simply shows the interaction between objects in the order they occur, highlighting the sequence in which these interactions take place. It just describes how and in what order the objects in a system function.

1.4. Class Diagram: This diagram identifies the different objects within the system and the relationships between them. It includes their attributes, methods, and how these objects interact with each other.

By using these diagrams, it becomes easier to understand the system's requirements, design it accurately and efficiently, and focus on the key areas that need attention during the upcoming stages of development.

2. Use case diagrams Section:

We designed our system to support three main types of users: Super-admin, Admin, Teacher. These are the use-cases for each user with a table that explains it on general:

2.1. Admin use-case: The following table provides a brief overview of the main Admin use cases and their descriptions:

<i>use-case</i>	<i>Description</i>
Authentication & Profile	Allows admins to log in securely and manage their profile information
Create Teachers	Allows admins to create teachers accounts.
View all topics	Viewing all of the systems topics and with filtering
Review Topics	Validate or reject submitted graduation project topics. (change status)
Manage Juries	Automatically generate and edit juries while ensuring diversity constraints.
Export Data	Export selected juries and topics in PDF or Word format for official use

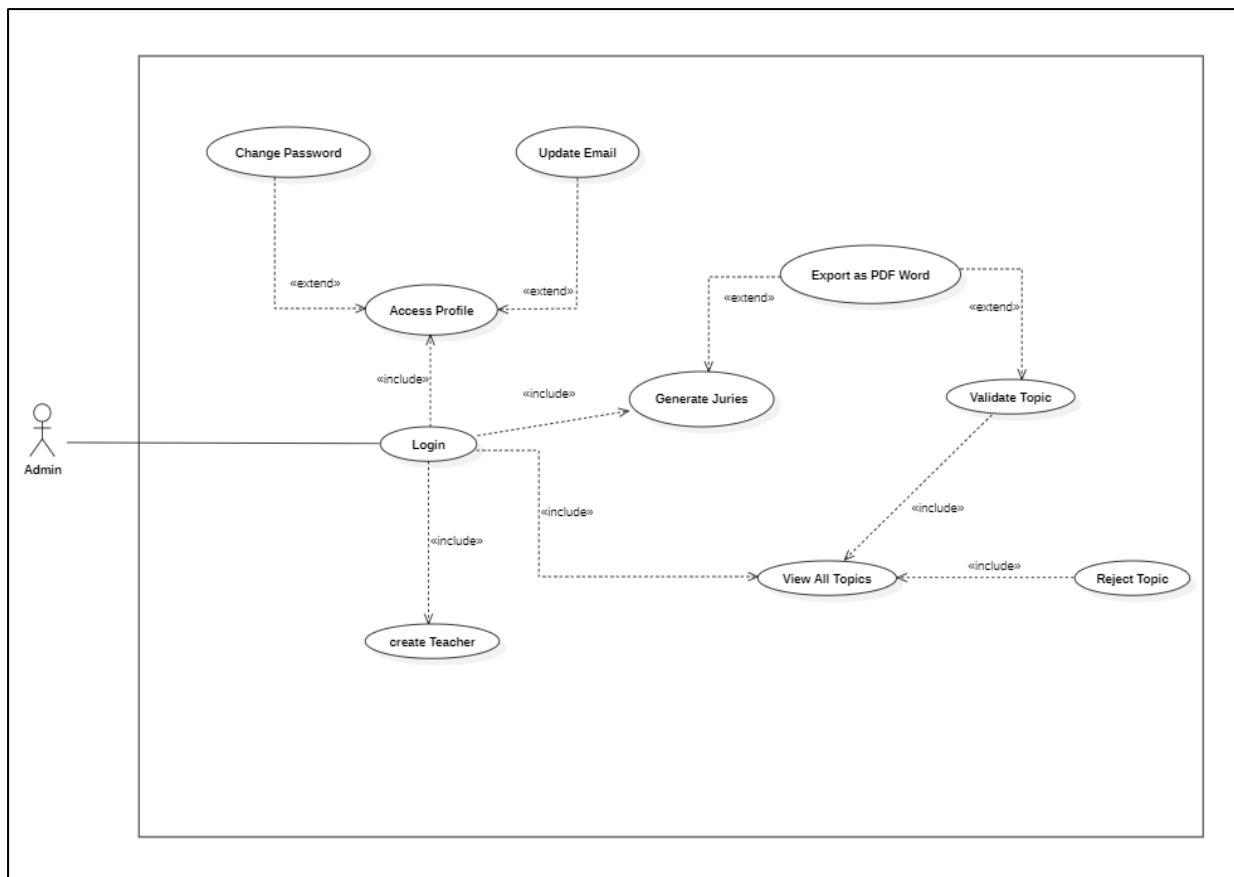


Figure 2.1: Admin's use-case diagram

2.2. Super-admin use-cases: The following table provides a brief overview of the main Super-admin use cases and their descriptions:

<i>use-case</i>	<i>Description</i>
Authentication & Profile	Log in securely and manage personal account settings.
Create Admins	Allows Super-admin to create Admin's accounts.
Modify Academic Levels	Update the academic level of any teacher in the system.
User Deletion	Removes any user account from the system when necessary.
System-Wide Control	Access all Admin features, plus full oversight of all user accounts.

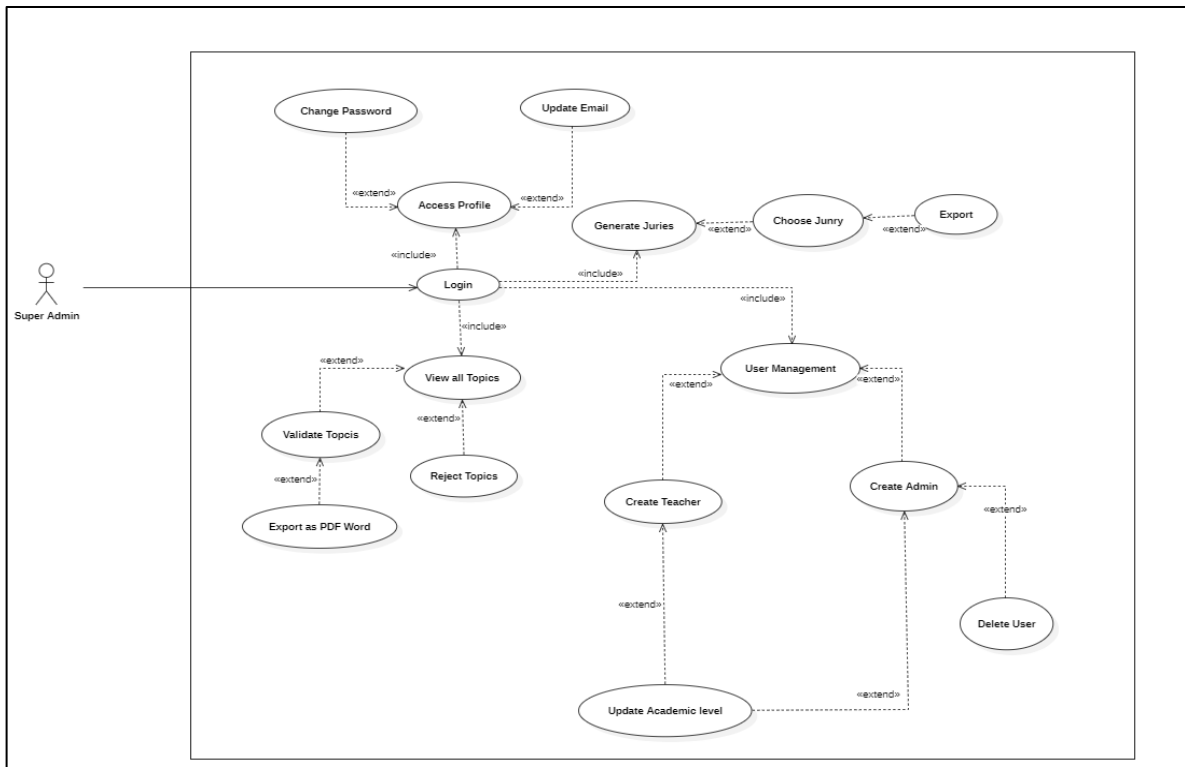


Figure 2.2: Super Admin use-case diagram

2.3. Teacher use-case: The following table provides a brief overview of the main Teacher use cases and their descriptions:

<i>use-case</i>	<i>Description</i>
Authentication & Profile	Secure login and personal account management.
Topic Management	Submit new topics, edit existing ones, or delete them.
View Topics & Status	Access and track the status of their own submitted topics
Evaluations student	Fill out and submit evaluation forms for assigned Submit projects
Export Evaluation/Topics	Export completed evaluation forms or topics to PDF or Word formats

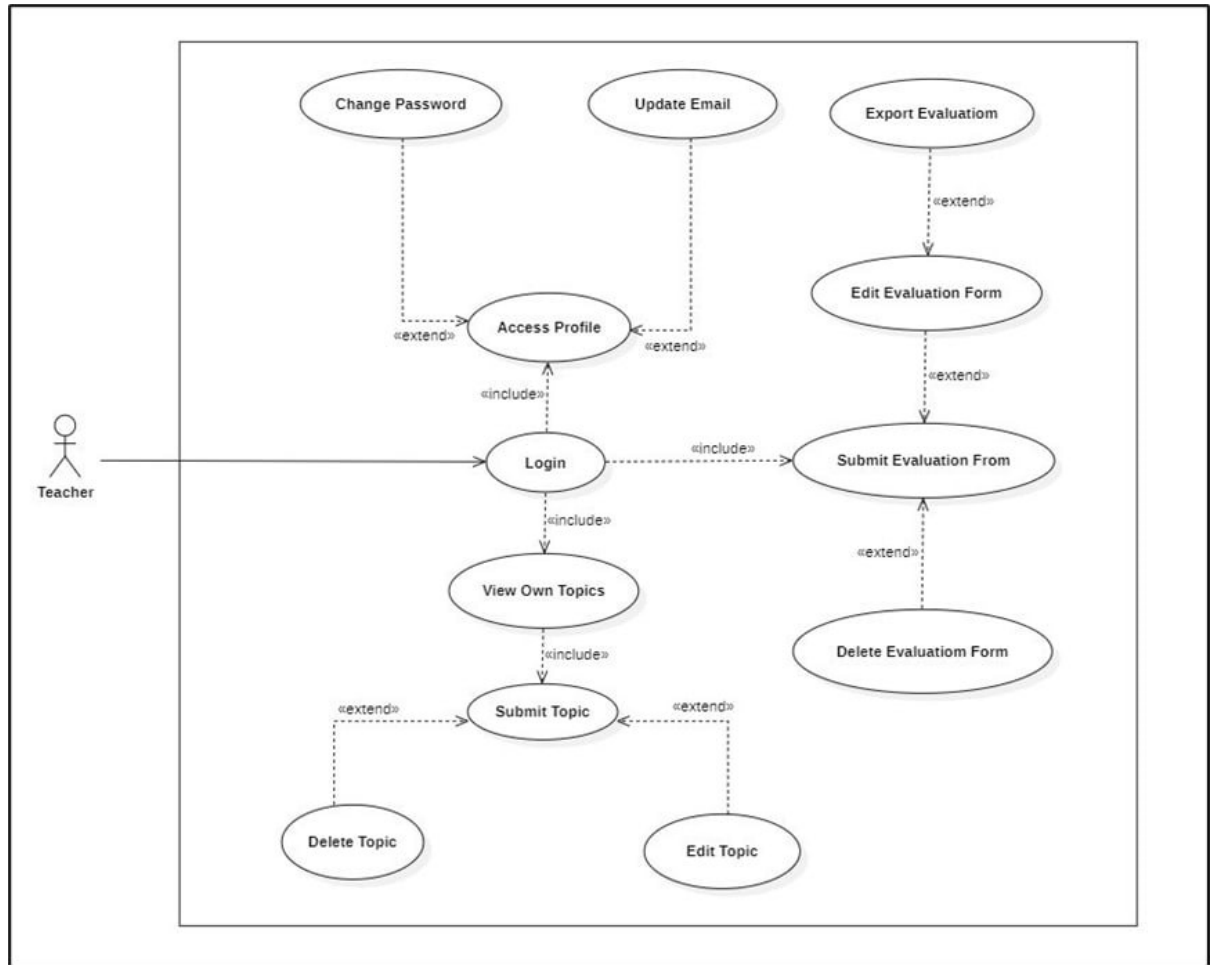


Figure 2.3: Teacher use-case diagram

3. Sequence Diagrams Section:

In the following section, we present sequence diagrams for all the users:

- Signing in
- Signing out
- Creating accounts
- Forgot password
- Reset Password

Each diagram provides a clear view of how requests are processed and how different system elements

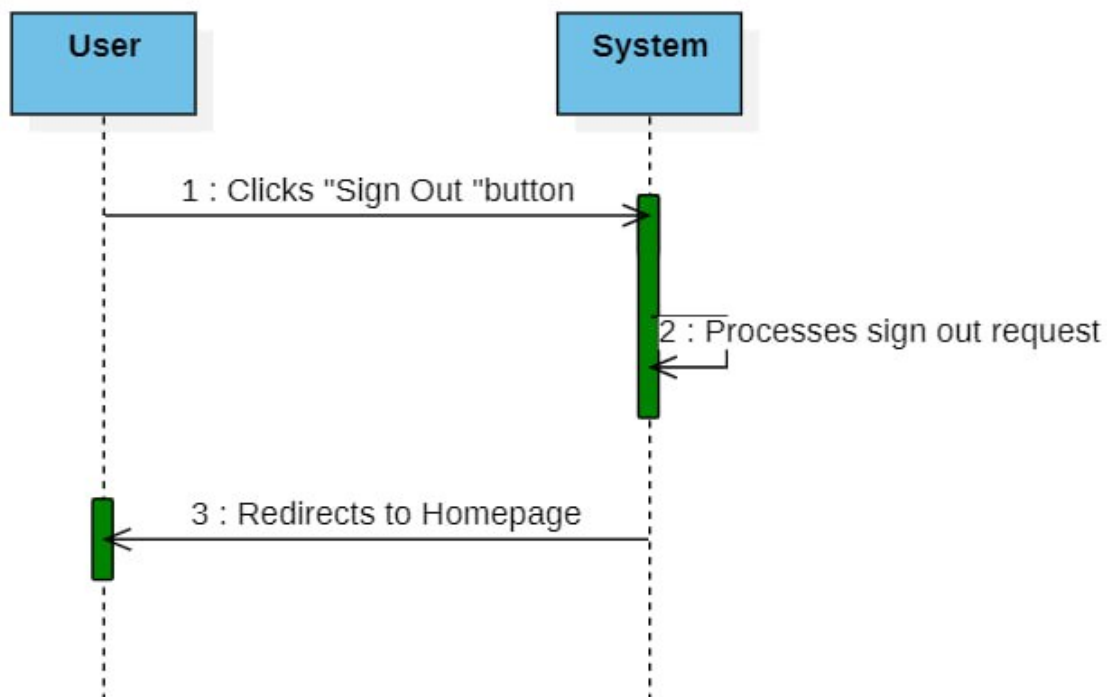


Figure 2.6: sign-out sequence diagram

Figure 2.5: Sign-in sequence diagram

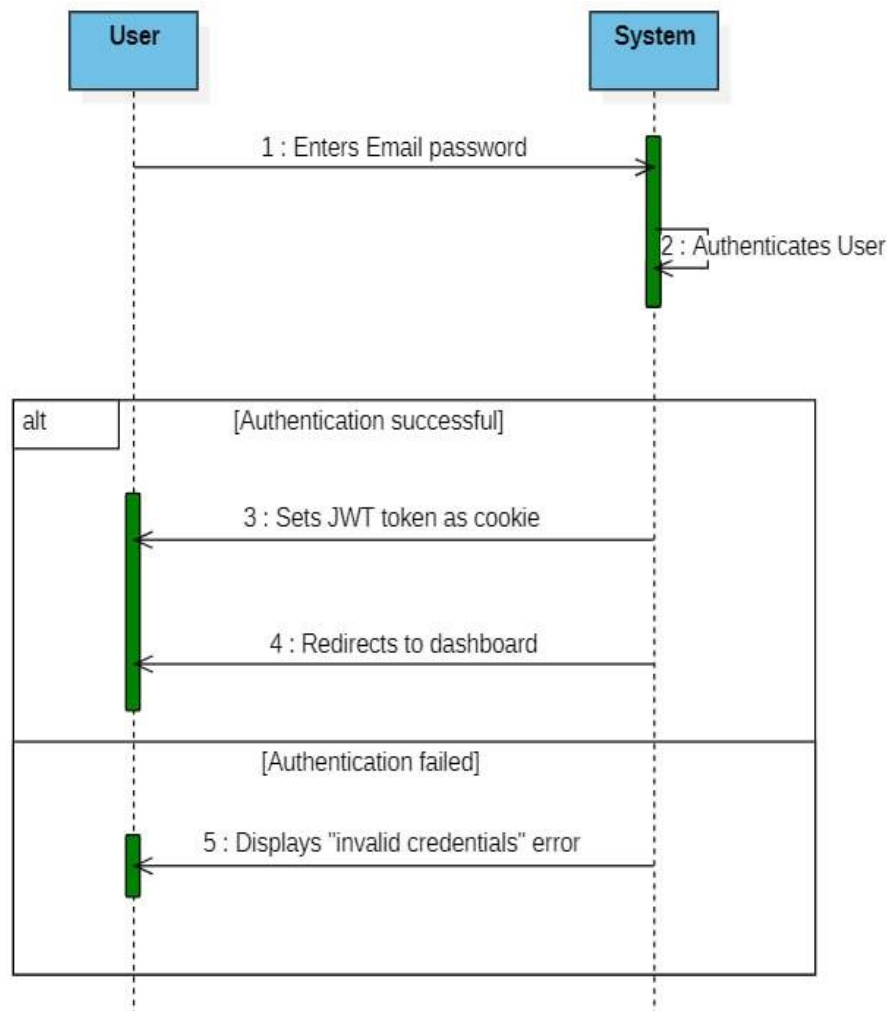


Figure 2.5: Sign-in sequence diagram

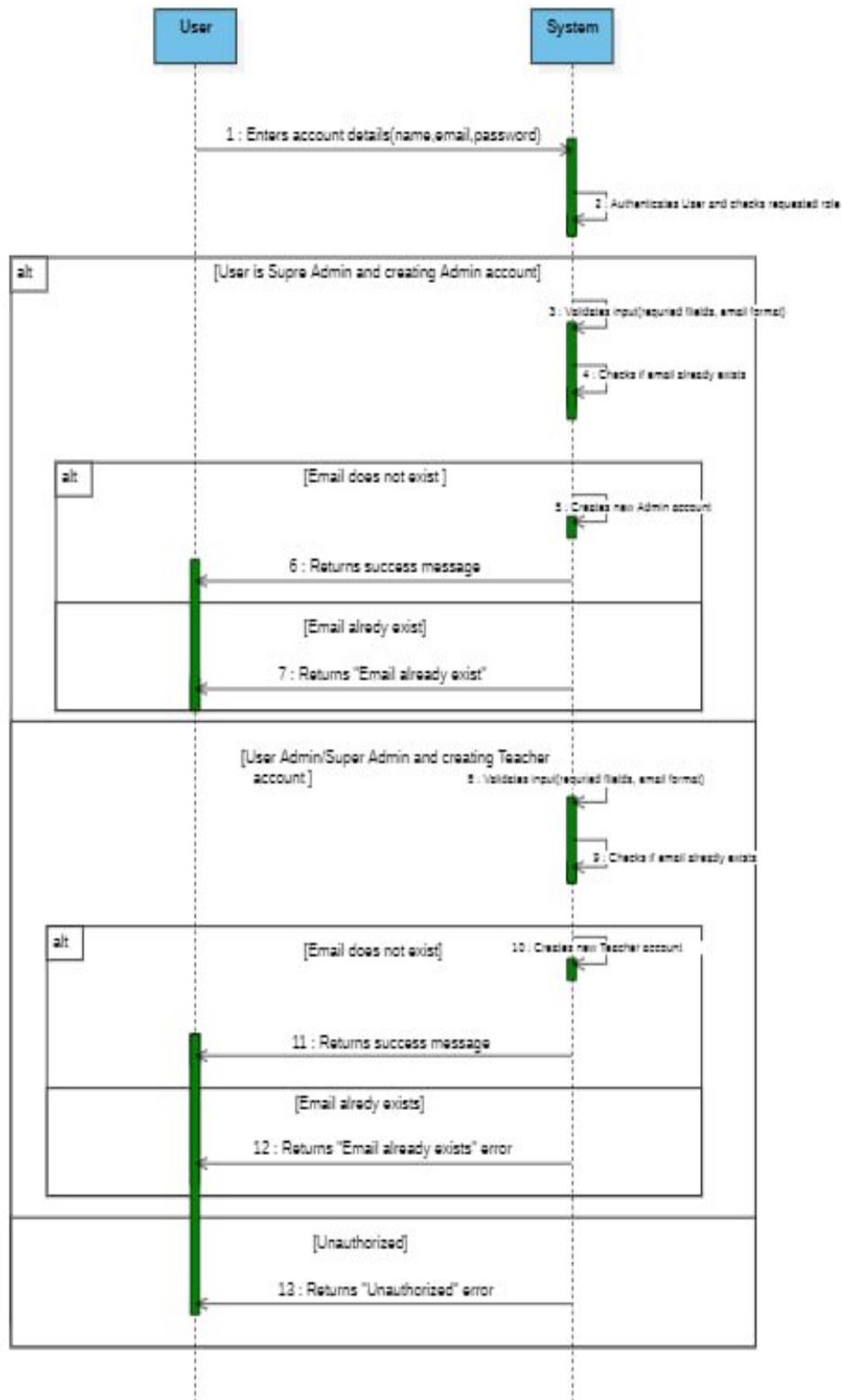


Figure 2.7: Create Teachers/Admins accounts sequence diagram

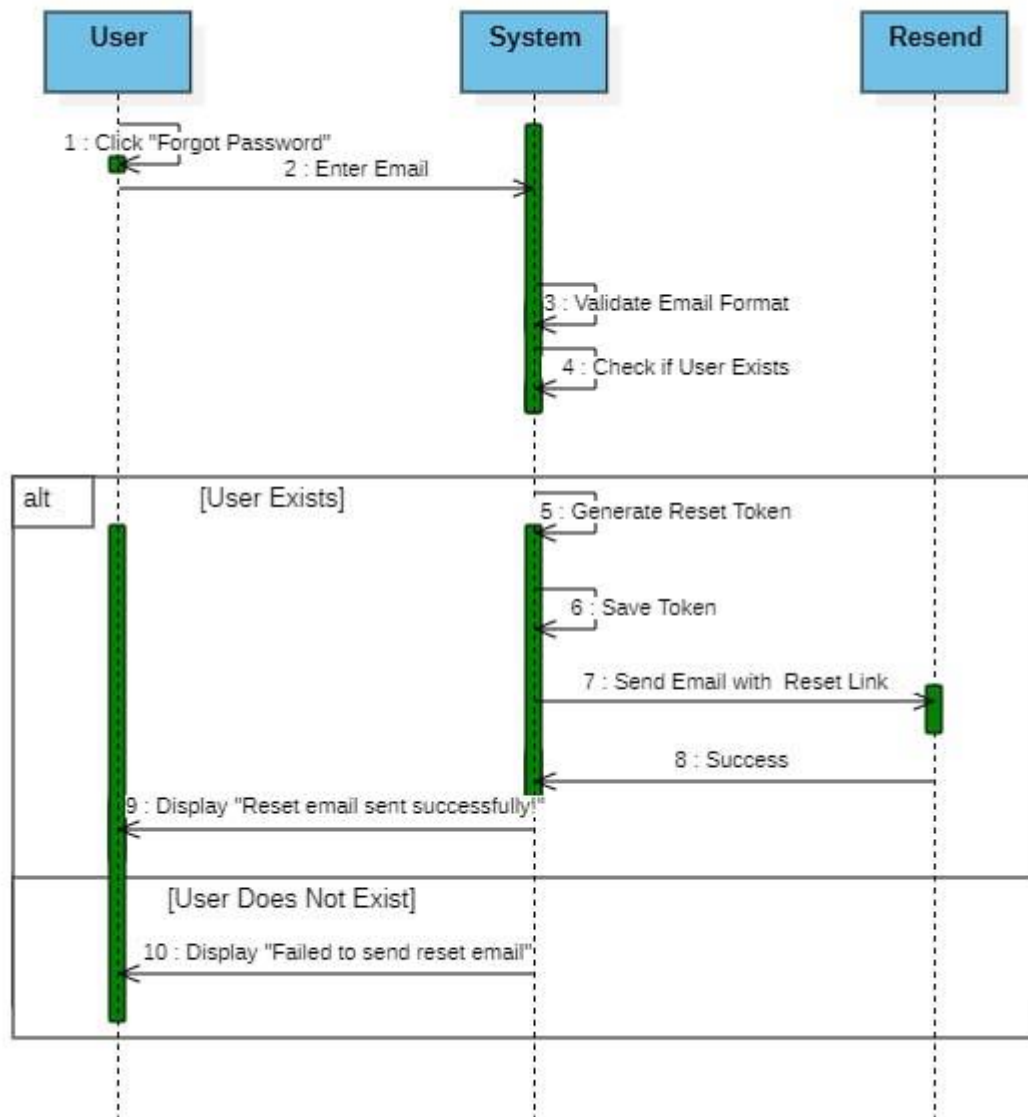


Figure 2.8: Forgot password sequence diagram

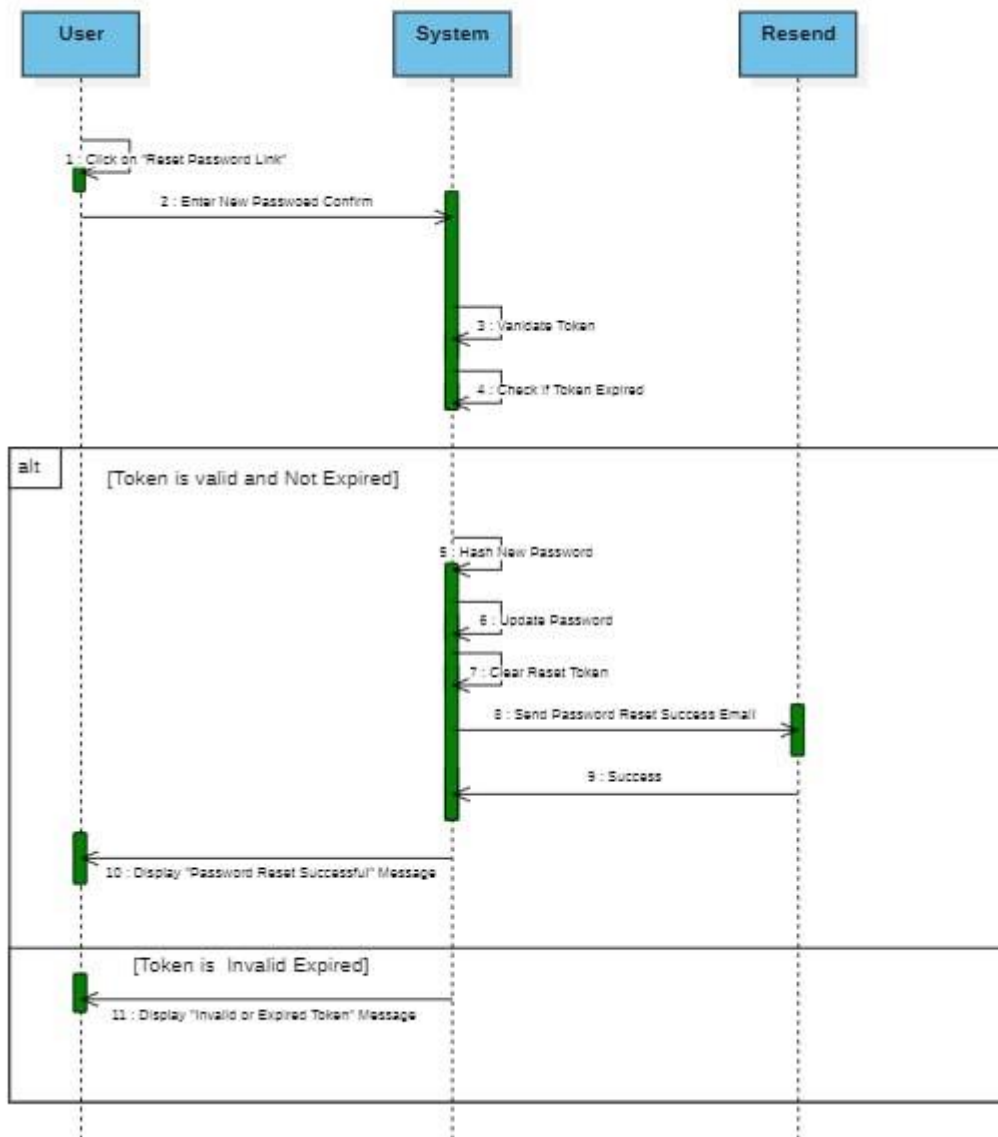


Figure 2.9: Reset Password sequence diagram

4. Activity diagrams Section:

Before we show the activity diagrams, we start with a table that highlights the main responsibilities of each user type. This helps give a better understanding of how users interact with the system.

2.1. User Role Permissions Table:

<i>Category</i>	<i>Feature</i>	<i>Super Admin</i>	<i>Admin</i>	<i>Teacher</i>
Account Management	Update Email	✓	✓	✓
	Change Password	✓	✓	✓
	Retrieve Password	✓	✓	✓
	Create Admin	✓	✗	✗
	Create Teacher	✓	✓	✗
	Delete Users	✓	✗	✗
	Update Teacher's Level	✓	✗	✗
Topic Management	Propose Topics	✗	✗	✓
	Edit/Delete Topics	✗	✗	✓
	View All Topics	✓	✓	✗
	View Own Topics	✗	✗	✓
	Filter Through Topics	✓	✓	✓
	Export Topics PDF/Word	✓	✓	✓
Jury Management	Generate Juries	✓	✓	✗
	View/Edit Chosen Juries	✓	✓	✗
	Export Chosen Juries PDF/Word	✓	✓	✗
Evaluations Management	Submit evaluation Forms	✗	✗	✓
	Edit/Delete Evaluation Forms	✗	✗	✓
	Export Evaluation Forms PDF/Word	✗	✗	✓

4.1. The activity diagrams: The following activity diagrams demonstrate how different users interact with the system through various core functions. Each diagram represents a specific task carried out by a designated user role, providing a clear visualization of the process flow:

Activity diagrams for all users:

- Change Password
- Update Email

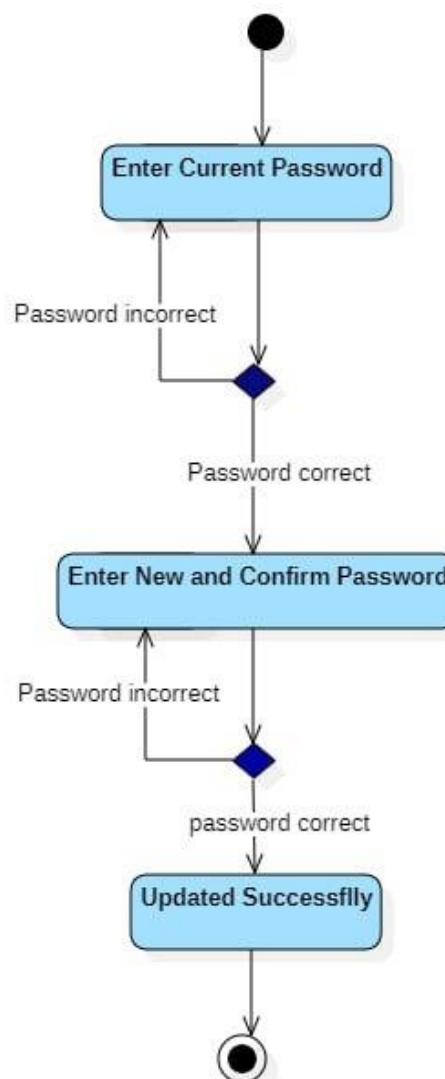


Figure 2.10: Change Password Activity Diagram

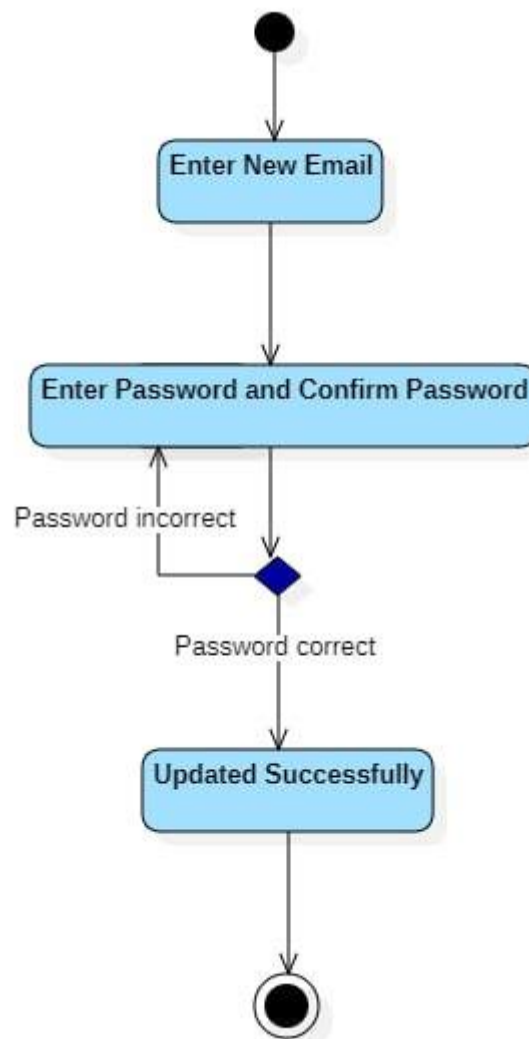
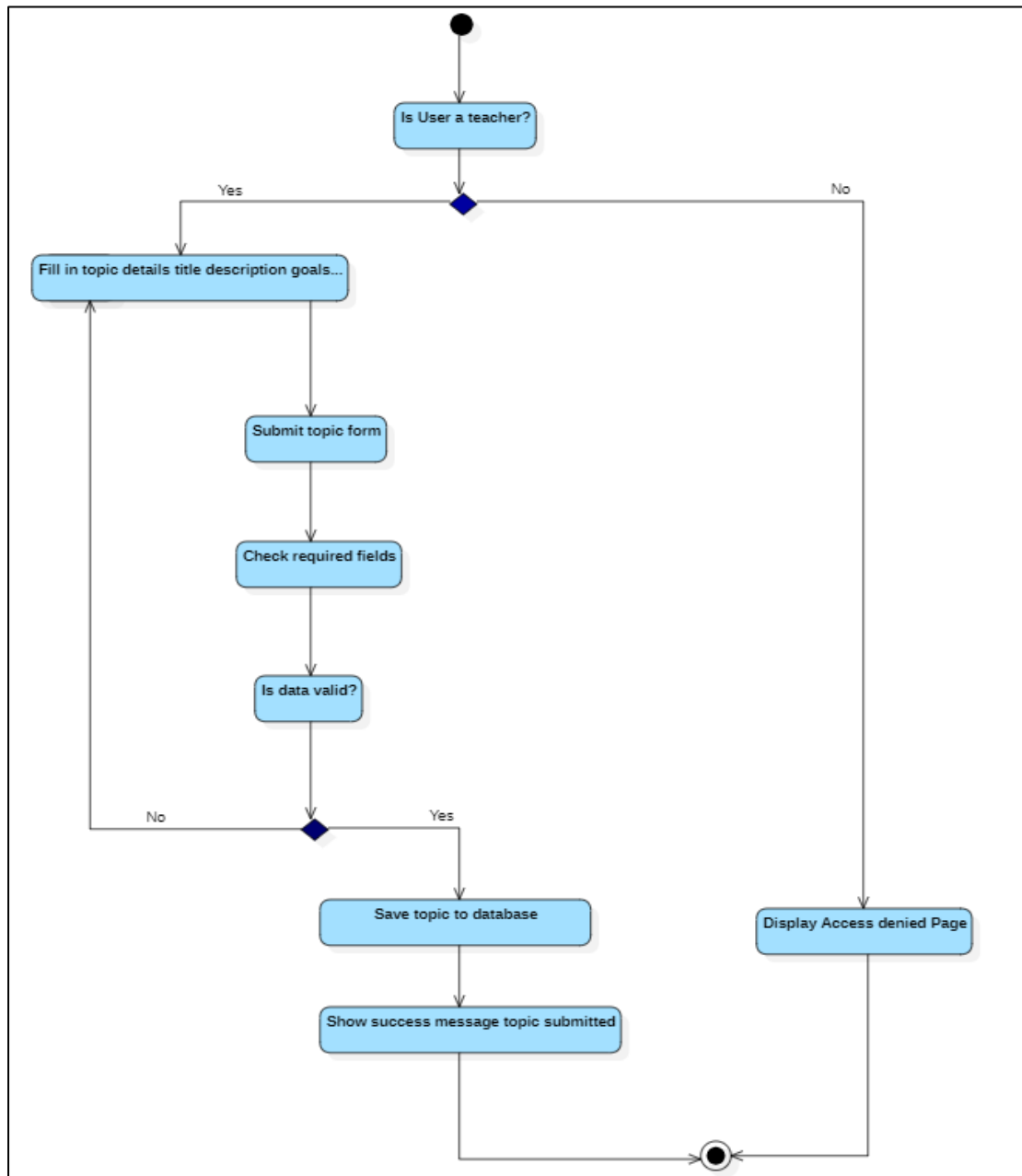


Figure 2.11.: Update Email Activity Diagram

Activity Diagrams for Teachers

- Submit Topics diagram
- Edit/Delete Topic diagram
- Submit Evaluation Form diagram
- Edit/Delete Evaluation Form diagram

**Figure 2.12: Submit Topics Diagram**

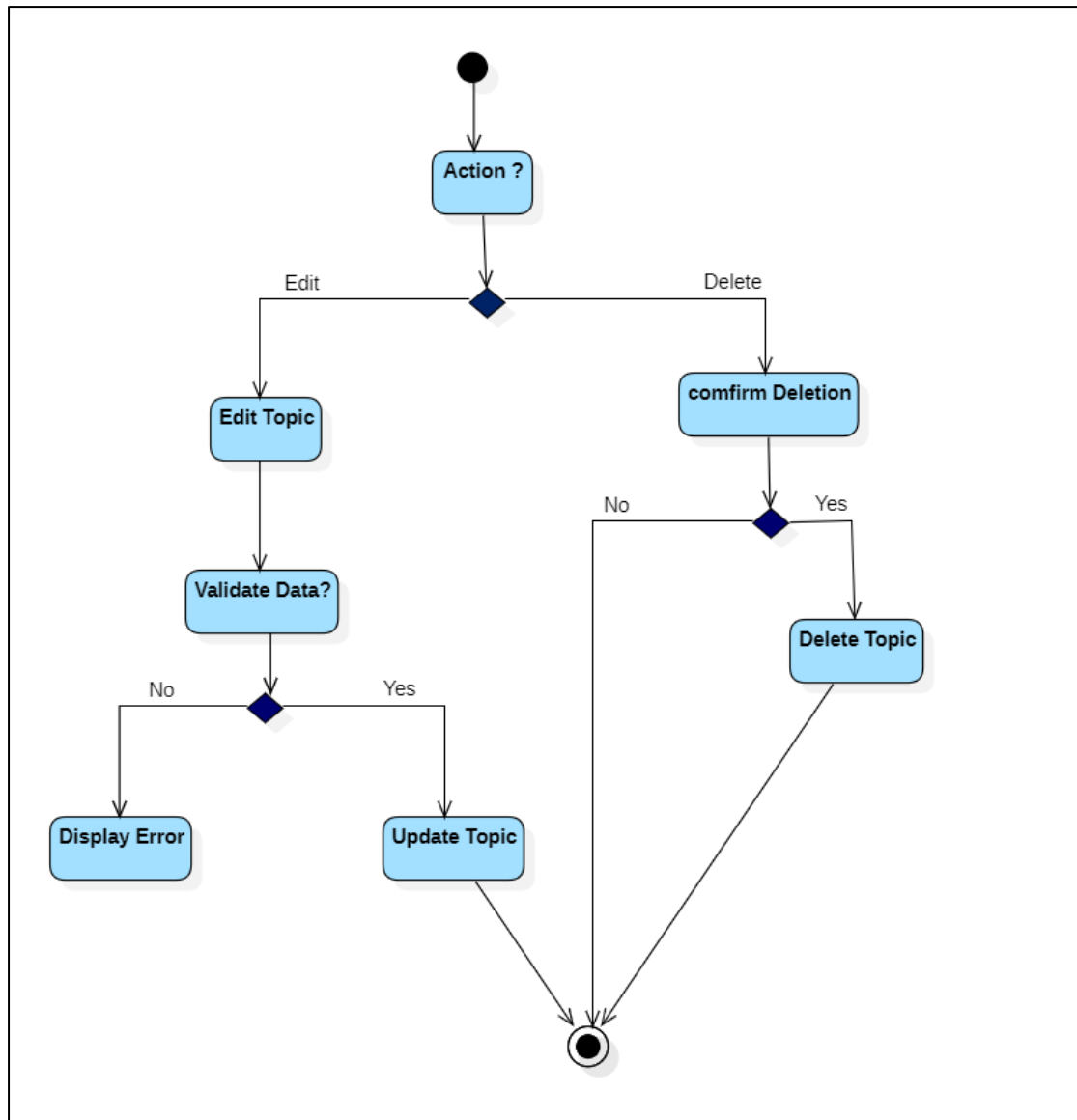


Figure 2.13: Edit/Delete Topic diagram

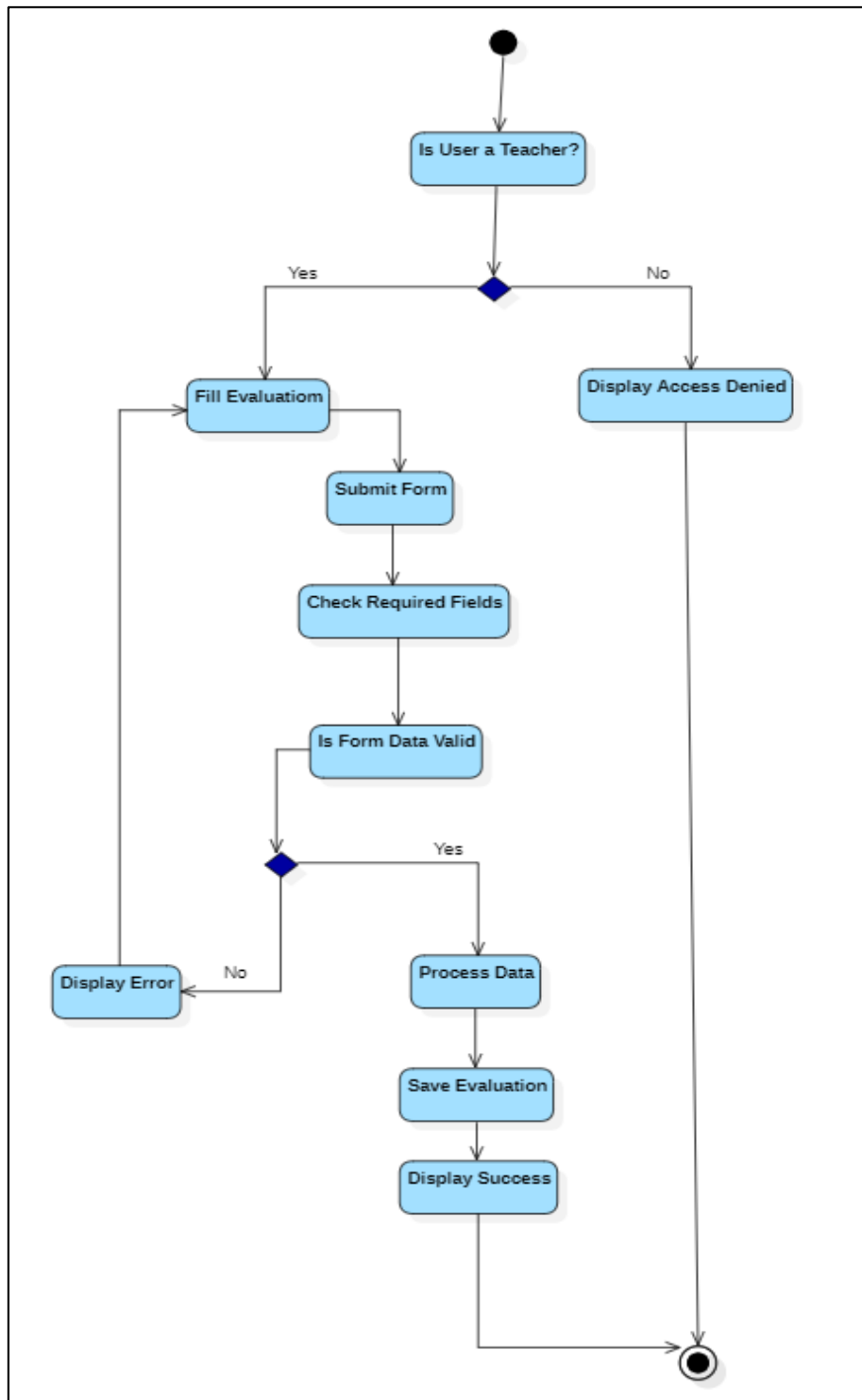


Figure 2.14: Submit Evaluation Form diagram

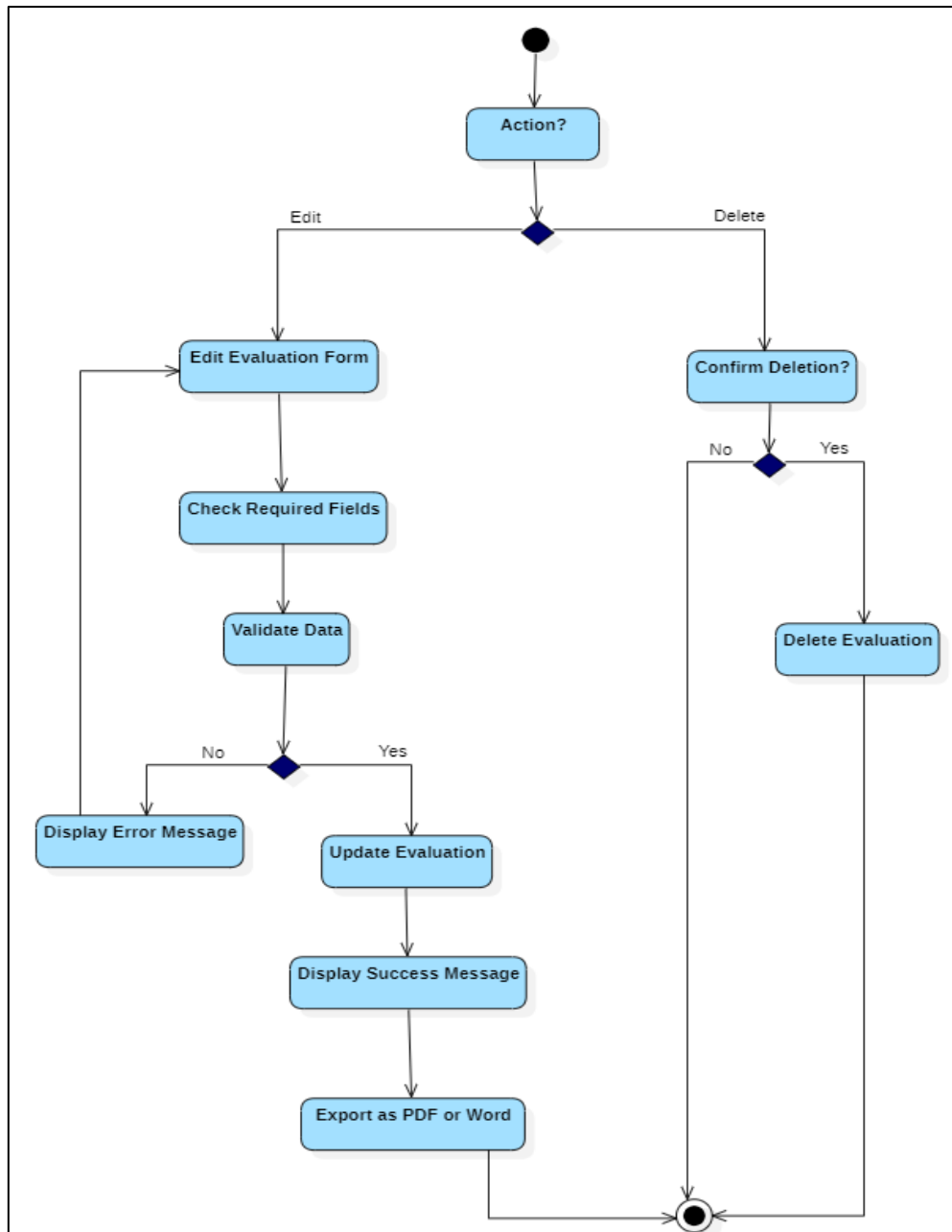
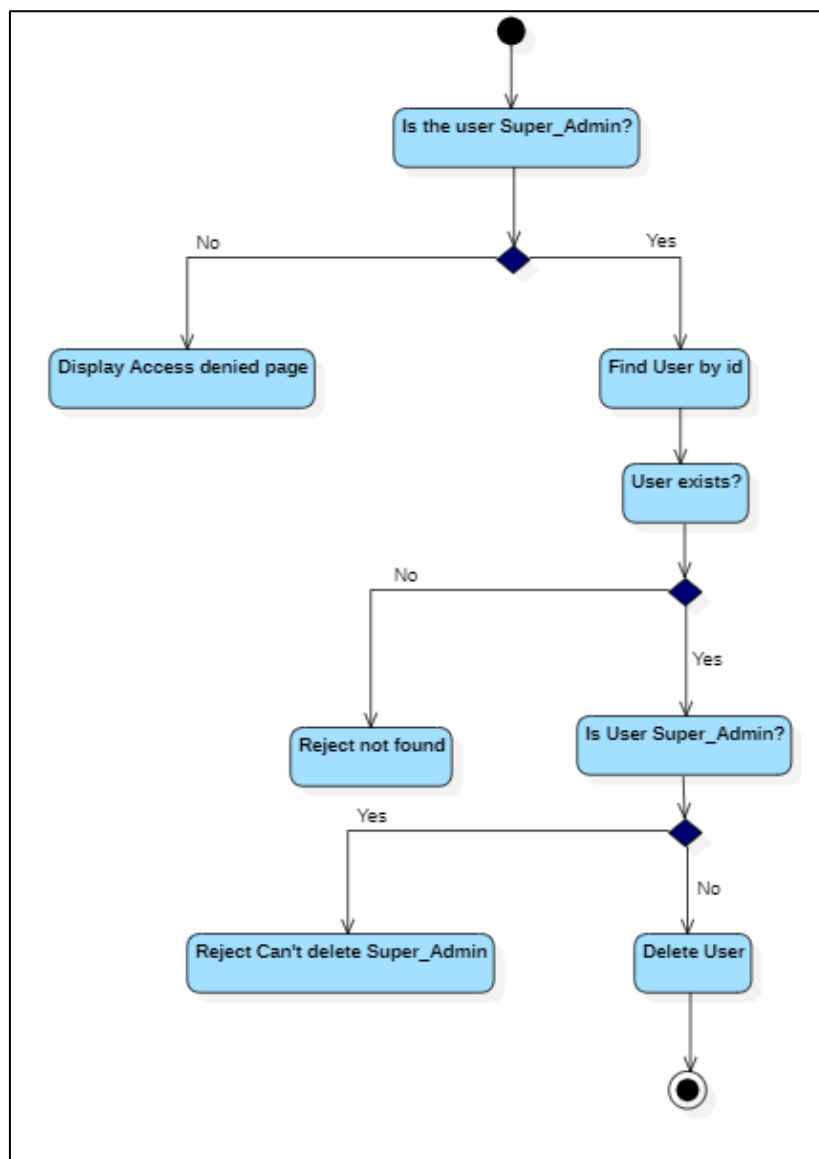


Figure 2.15: Edit/Delete Evaluation Form Activity diagram

Activity Diagrams for Super-admin/Admin:

- Delete Users diagram (Super-admin only).
- Update teacher's academic level (Super-admin only).
- Change topic's status.
- Generate juries.
- Actions related to the generated jury.

**Figure 2.16: Delete User Activity Diagram**

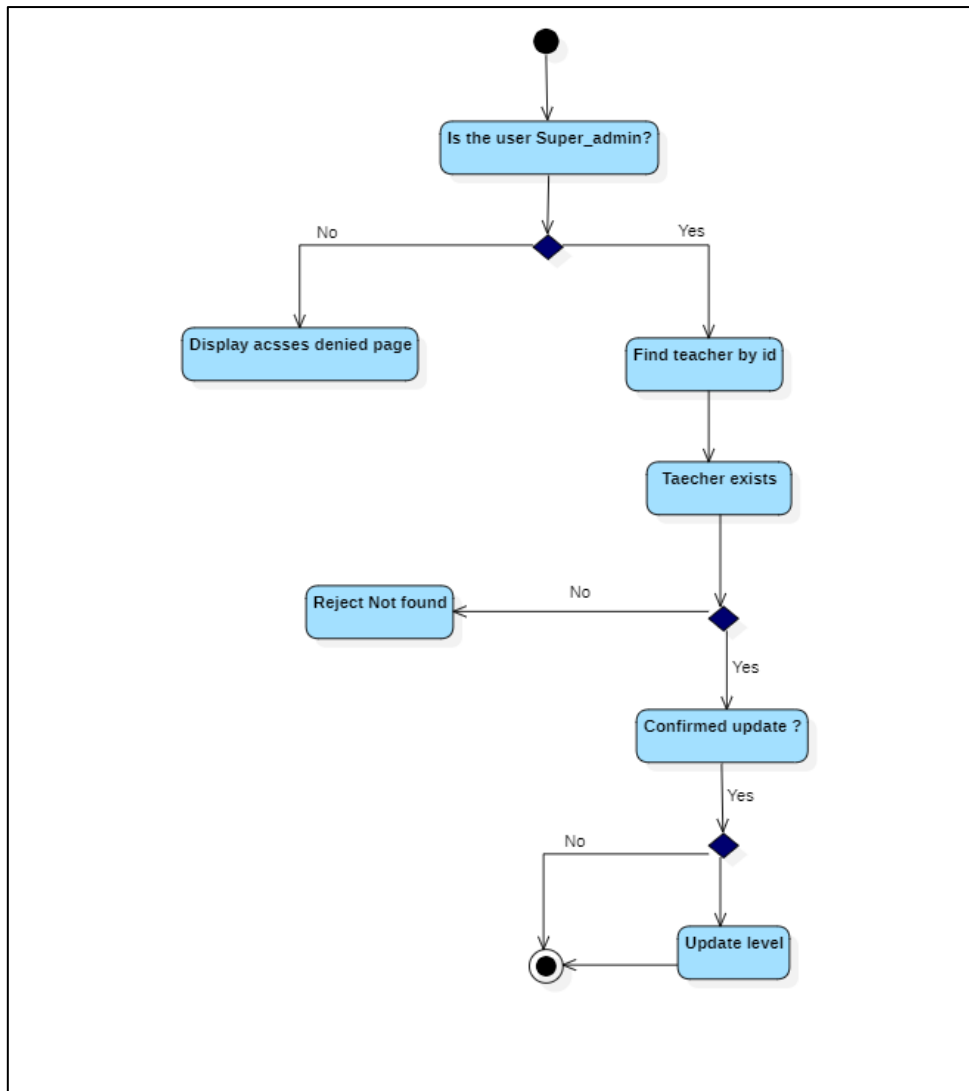


Figure 2.17: Update Teacher's Academic Level

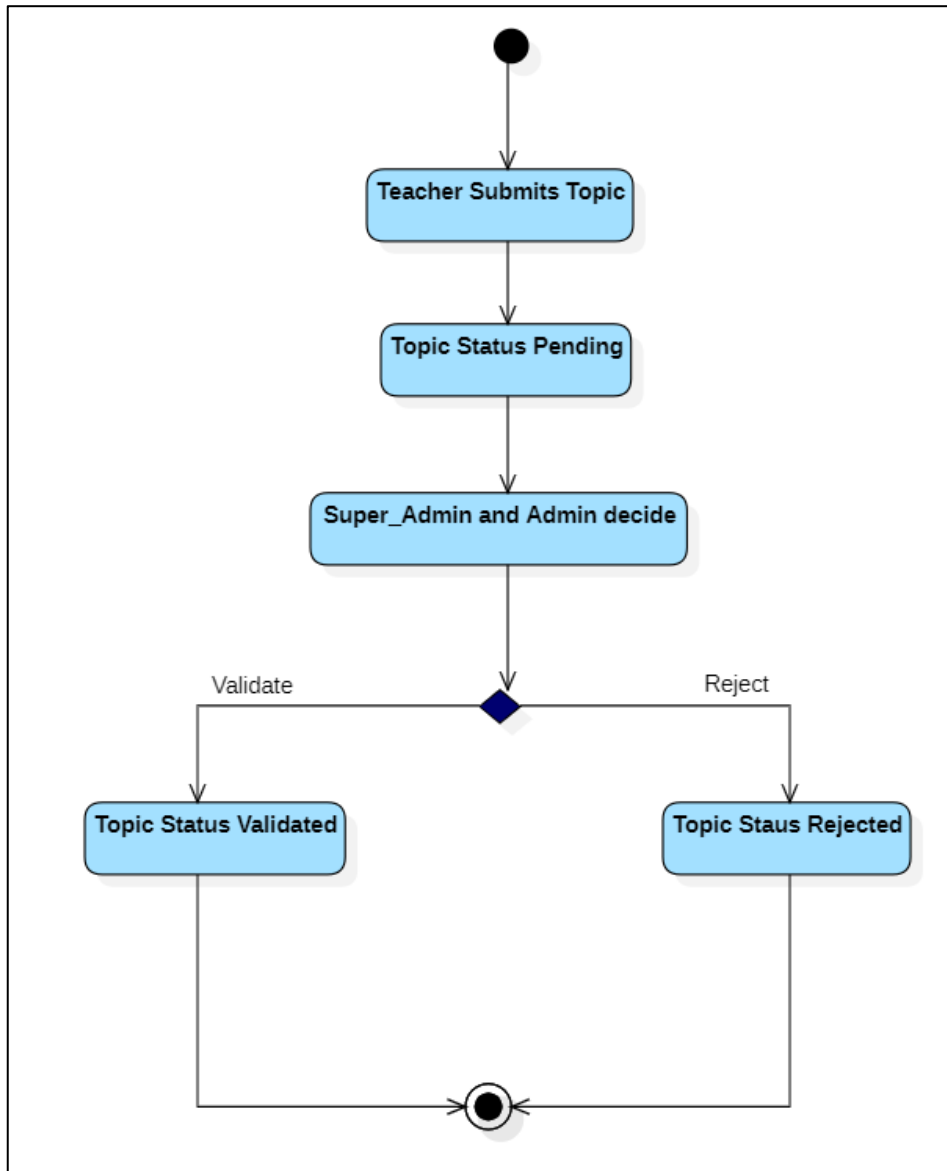


Figure 2.18: Change Topic's status Activity diagram

Before we look at the activity diagrams for generating the juries, let's first go over the rules behind jury assignment. The tables below show two key things:

- The diversity rules each jury must follow.
- A brief summary of the algorithm used to assign juries fairly.

This is to help make sure the process is balanced, inclusive, and follows our project guidelines.

Diversity Requirements:

<i>Category</i>	<i>Requirement</i>
Group Diversity	<ul style="list-style-type: none"> - Professors are in Groups 1, 2, 3 - The jury cannot be formed solely from Group 1 or 3 - Group 2-only juries are valid.
Gender Diversity	<ul style="list-style-type: none"> - Juries should include both male and female professors when possible. - Gender diversity is tracked and indicate.
Academic Level Diversity	<ul style="list-style-type: none"> - Include professors of different academic levels: Assistant Professor, Professor, Lecturer - Aim for balanced rank representation.
Workload Distribution	<ul style="list-style-type: none"> - Prefer professors with fewer than 2 assignments - Flag those with more than 2 to prevent overloading - Distribute workload evenly

Jury Assignment Algorithm:

<i>Step</i>	<i>Description</i>
Step 1: Initialize Candidate Pool	<ul style="list-style-type: none"> - Filter eligible professors. - Group by academic level, gender, and group. - Calculate current assignment count.
Step 2: Topic Prioritization	<ul style="list-style-type: none"> - Sort topics by academic level.
Step 3: Role Assignment	<ul style="list-style-type: none"> - Assign Supervisor (topic owner). - Identify Chair from higher academic levels. - Select Examiner.
Step 4: Diversity Optimization	<ul style="list-style-type: none"> - Score combinations based on the previous requirement. - Prioritize those with higher diversity scores. - Enforce group diversity rules.

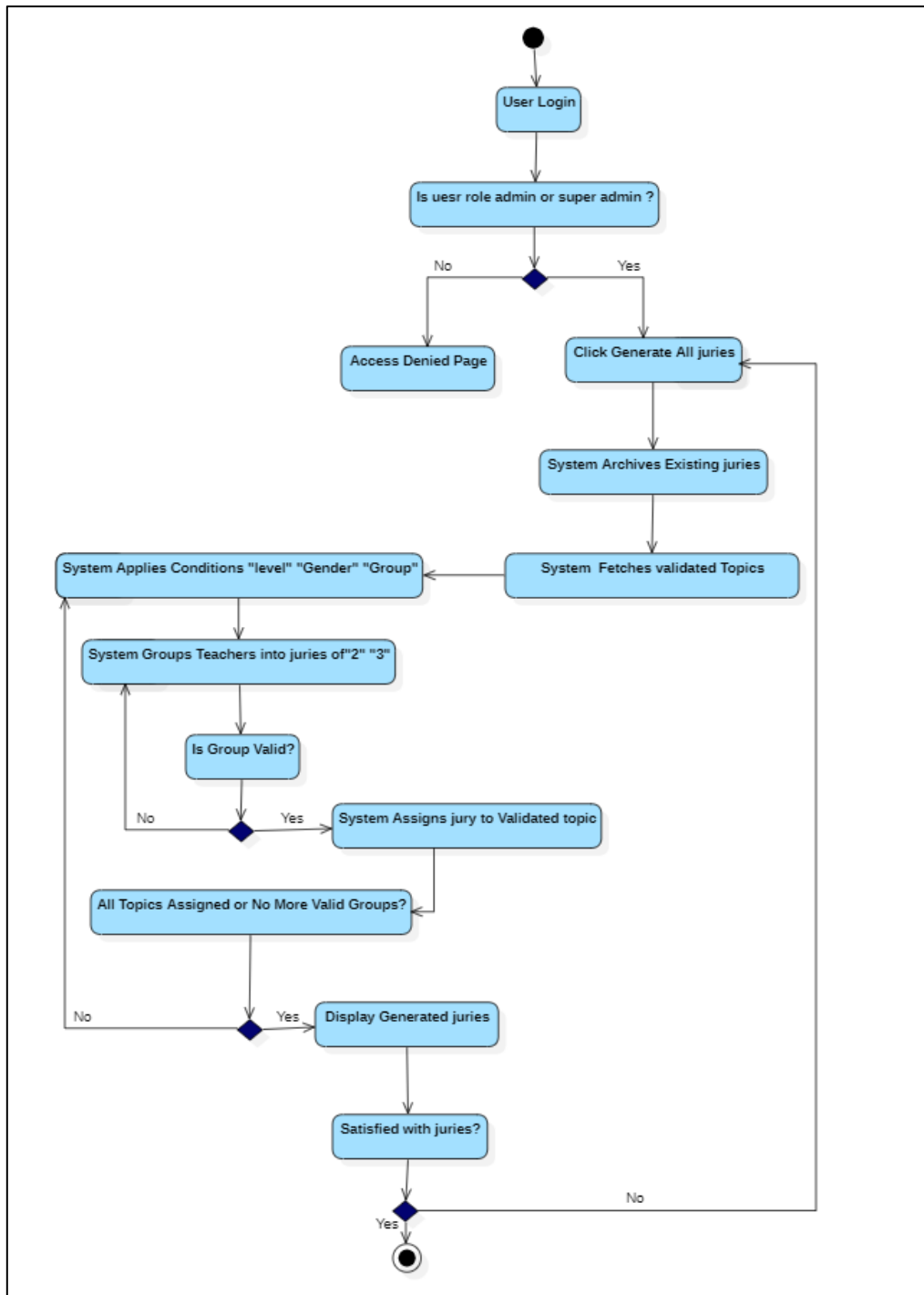


Figure 2.19: Generate Juries Activity Diagram

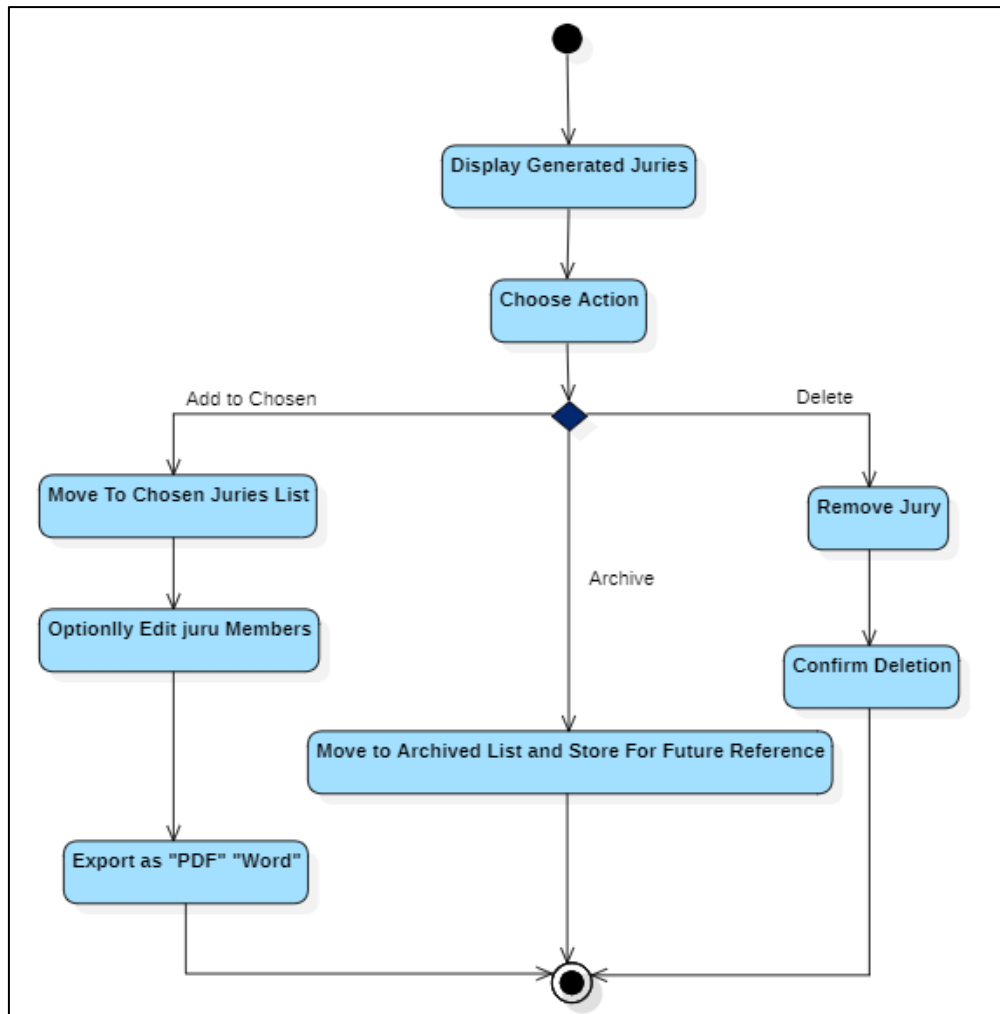


Figure 2.20: Actions related to the generated juries Activity Diagram

5. Class Diagram:

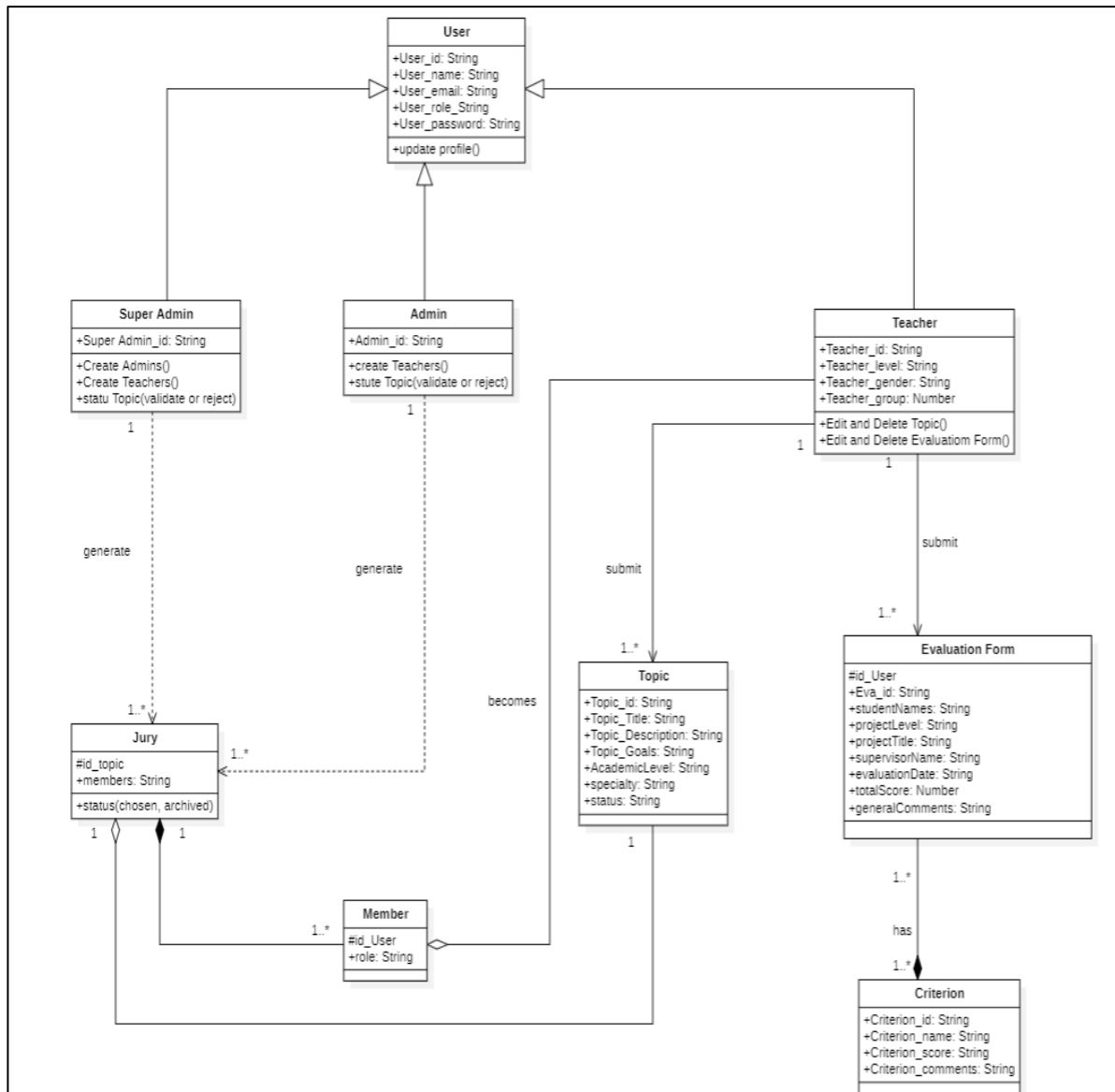


Figure 2.21: Class Diagram For Users

5.1. Relational Model:

User (User_id, User_name, User_email, User_role, User_password)

SuperAdmin (Super_Admin_id, #User_id)

Admin (Admin_id, #User_id)

Teacher (Teacher_id, #User_id, Teacher_level, Teacher_gender, Teacher_group)

Topic (Topic_id, Topic_Title, Topic_Description, Topic_Goals, AcademicLevel, Specialty, Status)

Jury (#id_topic, members, status)

Member (#id_User, #id_topic, role)

EvaluationForm (Eva_id, #Topic_id, StudentNames, projectLevel, projectTitle, supervisorName, evaluationDate, totalScore, generalComments)

Criterion (Criterion_id, #Eva_id, Criterion_name, Criterion_score, Criterion_comments)

Conclusion:

Finally, in this chapter, we conducted a comprehensive analysis and precise modeling of our project with the aim of developing the application. We provided thorough answers to the questions related to outlining and modeling, relying on the use of UML as a primary tool in designing a diverse set of diagrams, including use case diagrams, sequence diagrams, activity diagrams, and class diagrams. These models represent a crucial step toward achieving development goals efficiently and accurately.

Chapter 03:
Implementation

Introduction:

Following the system's design and modeling phase using UML, in this chapter we focus on the actual development of the application. We start by introducing the programming languages, frameworks, and technologies utilized in the process. The application is built using the MERN stack—comprising MongoDB, Express.js, React and Node.js. This technology stack was selected to deliver a modern, scalable, and maintainable application.

After that, the focus will shift to explaining the main graphical user interfaces of the website through illustrative examples.

1. Components of the work environment:**1.1. Languages:**

HTML: It stands for “Hyper-Text Markup Language”. It is the standard language for creating web pages. It structures content using elements like headings, paragraphs, links, and images.

CSS: It stands for Cascading Style Sheets. It is used to style and layout HTML elements, like colors, fonts, and spacing.

JavaScript: It is a programming language used to build interactive frontend websites and powerful backend servers. It lets developers create dynamic content, handle user events, and communicate with databases or APIs.[4]

JavaScript runs in all modern browsers and is essential for full-stack web development.

1.2. Frameworks and Libraries:

Tailwind CSS: It is a utility-first CSS framework that simplifies web development by providing a set of predesigned utility classes. These utility classes enable you to build custom designs without writing any custom CSS, promoting consistency, scalability, and efficiency.[9]

React.js: It is a JavaScript library for building user interfaces. It allows developers to create reusable UI components. It efficiently updates and renders components based on data changes.[3]

Node.js: It is a JavaScript runtime built on Chrome's V8 engine. It allows developers to run JavaScript on the server-side. It is widely used for building scalable and fast backend applications.[4]

Express.js: It is a minimal and flexible Node.js web application framework. It simplifies routing, middleware integration, and handling HTTP requests. It is commonly used for building backend APIs and web applications.[5]

MongoDB: It is a NoSQL database. It stores data in flexible, JSON-like documents. It is ideal for handling unstructured or dynamic data in modern applications.[6]

Mongoose: It is an ODM (Object Data Modeling) library for MongoDB and Node.js. It provides a straightforward way to model data, define schemas, and interact with MongoDB using JavaScript.[7]

1.3. Tools and Utilities:

Git: It is a version control system that tracks code changes and helps multiple developers collaborate.[11]

GitHub: It is a platform for hosting and sharing Git repositories. It enables collaboration, version control, and code management for developers.[12]

VS Code: It is a lightweight, open-source code editor. It supports multiple programming languages, debugging, and extensions for a customizable development environment.[13]

Star-UML: It is a powerful modeling tool used to design software architecture using UML (Unified Modeling Language) diagrams. It helps developers visualize system structure, relationships, and workflows during the planning and design phase.[14]

Thunder client: It is a tool for testing and interacting with APIs. It allows developers to send requests, inspect responses, and automate API testing.[15]

Resend: It is an email API for sending transactional and marketing emails. It integrates easily with apps and provides features like scheduling, templates, and analytics. [16]

1.4. Other Tools:

Axios: It is a promise-based HTTP client for making requests to APIs. It works in both the browser and Node.js, making it easy to handle requests and responses. And in our project, we used a custom one.

Bcrypt: It is a library used to hash and compare passwords securely. It helps protect sensitive user data by salting and hashing passwords before storing them.

JSON-Web-Token (JWT): It is a compact, URL-safe way to represent claims between two parties. It is commonly used for authentication, where the token is passed between the client and server. **React Router:** It is a library for handling routing in React applications. It enables navigation between different views or components without reloading the page.

2-The Workflow of Our System: MERN Stack Explained:

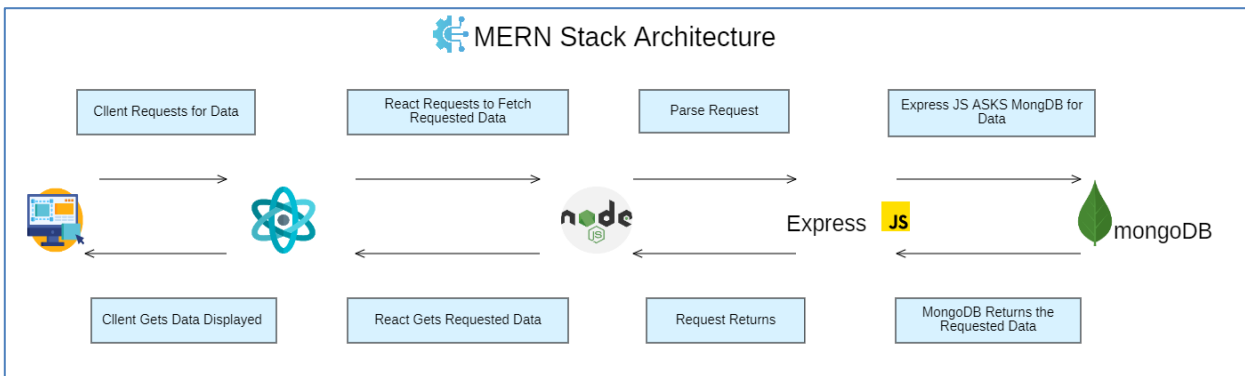


Figure 3.1: MERN Stack work flow

2.1. Frontend (React):

- This is the part that users interact with. It's like the face of the application. The frontend is built using React, and it shows everything the user needs to see, like lists of users, topics, or juries.
- When someone clicks a button (like assigning a jury or changing a teacher's level), React sends a request to the backend to make that change or to fetch some data.

2.2. Backend (Node.js + Express):

- This is the brain of the app. The backend (built with Node.js and Express) listens for those requests from React. For example, if the frontend asks for the list of juries or needs to assign a teacher to a jury, the backend processes that request and communicates with the database to get or update the right data.
- It makes sure the rules are followed (like ensuring the right mix of teachers for each jury based on gender and academic level).

2.3. Database (MongoDB):

- All of our data (teachers, juries, topics...) is stored in a MongoDB database. When the backend needs to check something, like which teachers are assigned to which juries, it talks to the database and gets the data.
- It helps store things like whether a teacher has already been assigned to a jury, what the current topics are, and what juries are "chosen" or "archived".

2.4. How it works:

- When the user interacts with the frontend (like clicking a button to validate or reject a topic), React sends an HTTP request to the backend.
- The backend then checks the database to retrieve the topic details and ensures the user has the correct permissions (e.g., the user's role should be a super-admin or admin).

- If everything is valid, the backend updates the topic's status in the database (either changing it to 'Validated' or 'Rejected') and sends a response back to the frontend with the updated status.
- React then updates the UI to show the new status of the topic (like showing "Validated" or "Rejected") or an error message if something went wrong.

In simple terms, the user sees and interacts with the frontend (React), which then talks to the backend (Node.js/Express) to fetch or update data from the database (MongoDB). This process helps manage the juries and ensures all the rules for assigning teachers are followed.

3.Database Schemas:

These are some of the most important database schemas in MongoDB:

User Schema:

```
const userSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  role: { type: String, enum: ['Teacher', 'Admin', 'Super-admin'], required:
true },
  level: { type: String, enum: ['Assistant Professor', 'Professor', 'Lecturer']
},
  gender: { type: String, enum: ['Male', 'Female'] },
  group: { type: Number, enum: [1, 2, 3] }
});
```

Topic Schema:

```
const topicSchema = new mongoose.Schema({
  title: { type: String, required: true },
  description: { type: String, required: true },
  goals: { type: String, required: true },
  academicLevel: { type: String, enum: ['Licence', 'Master'],
  required: true },
  specialty: { type: String, required: true },
  teacher: { type: mongoose.Schema.Types.ObjectId,
  ref: 'User', required: true },
  status: { type: String, enum: ['pending', 'validated', 'rejected',
  'completed'],
  default: 'pending' },
  submissionDate: { type: Date, default: Date.now }
});
```

Jury Schema:

```
const jurySchema = new mongoose.Schema({
  topic: { type: mongoose.Schema.Types.ObjectId, ref: 'Topic', required:
true }, members: [{ role: { type: String, enum: ['Chair', 'Supervisor',
'Examiner'],
  required: true } },
  teacher: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required:
true } }], diversityScore: { gender: { type: Boolean, default: false } },
  level: { type: Boolean, default: false },
  group: { type: Boolean, default: false } }
});
```

Evaluation Schema:

```
const evaluationSchema = new mongoose.Schema({
  studentNames: { type: String, required: true },
  projectLevel: { type: String, enum: ['Licence', 'Master'], required: true },
  projectTitle: { type: String, required: true },
  supervisorName: { type: String, required: true },
  evaluationDate: { type: Date, required: true },
  criteria: [{ name: { type: String, required: true } },
  score: { type: Number, min: 0, max: 10, required: true },
  comment: { type: String } }],
  totalScore: { type: Number, required: true },
  generalComments: { type: String },
  teacher: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required:
true }
});
```

These schemas provide structured data storage with validation rules, relationships, and defaults to maintain data integrity throughout the application.

4.The Main Interfaces of the Application:

(Dark/Light mode) Shared interfaces:

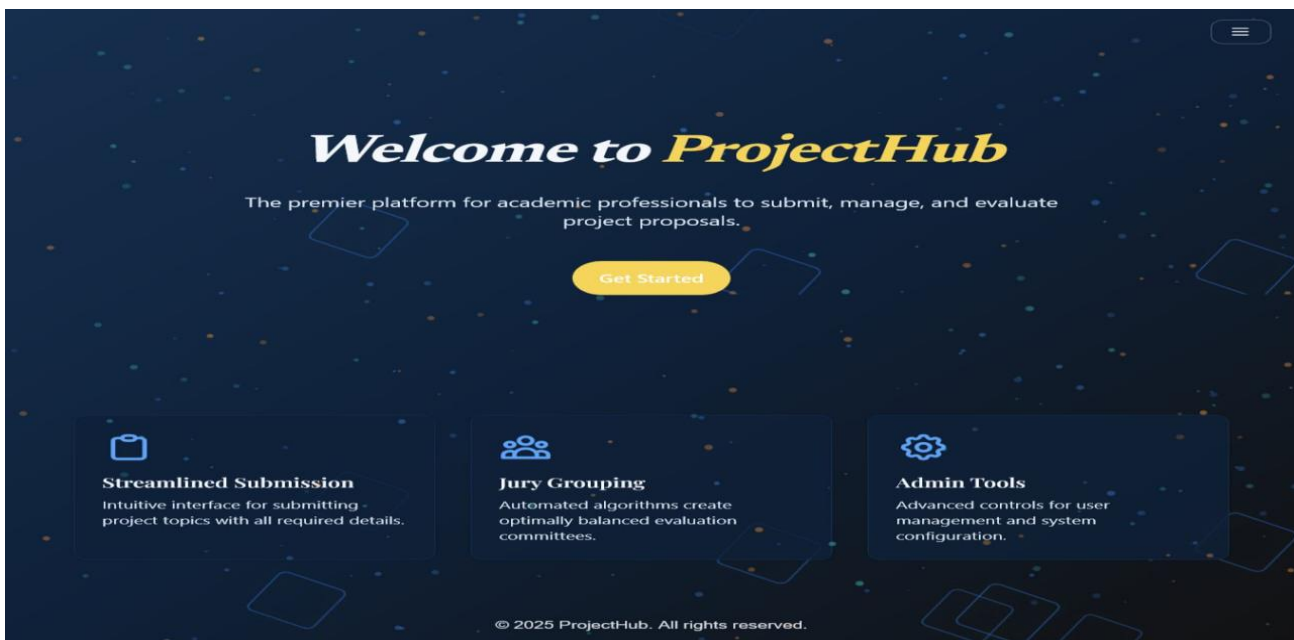
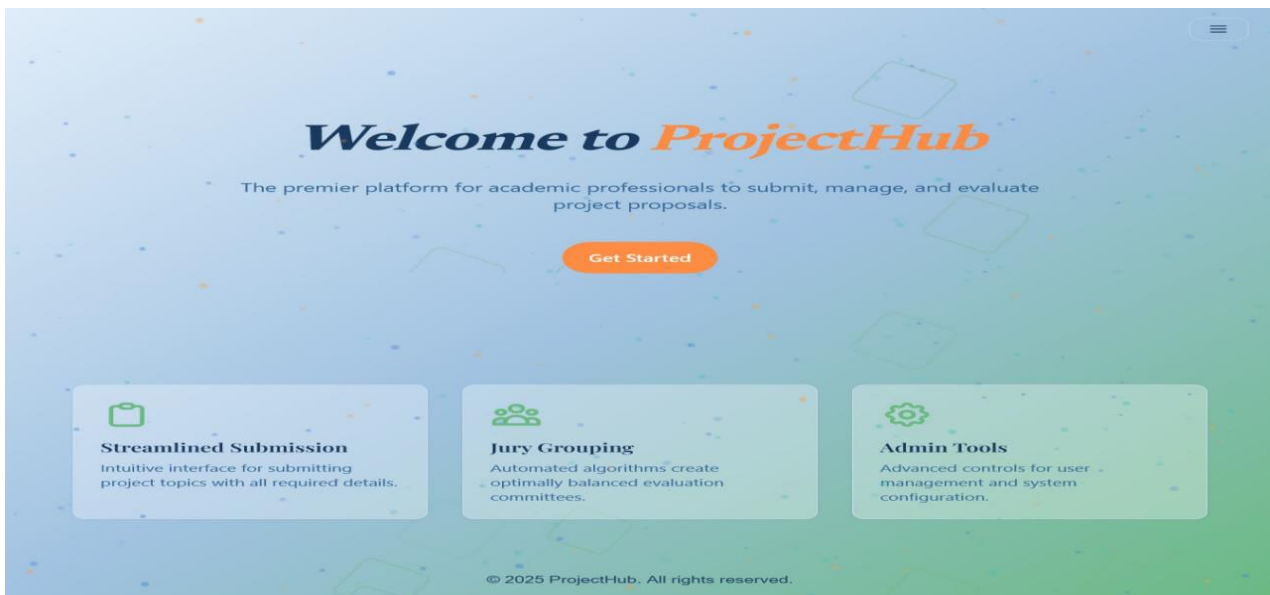


Figure 3.2: Home page (before signing in)

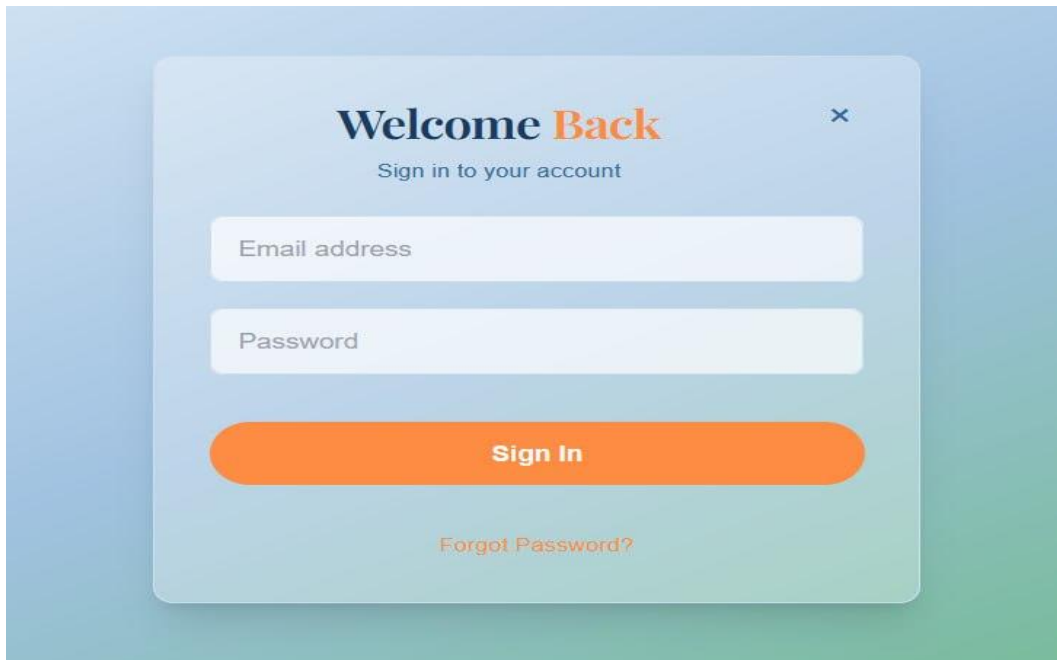


Figure 3.3: Sign-in pop up

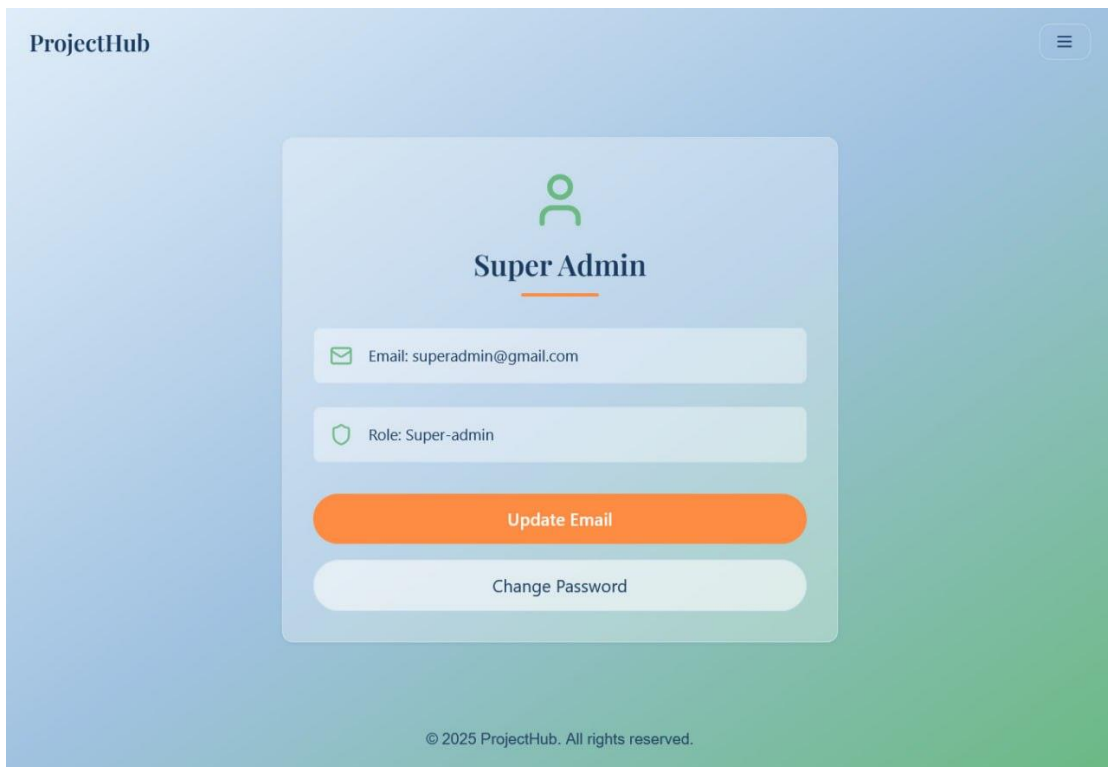


Figure 3.4: Profile Page

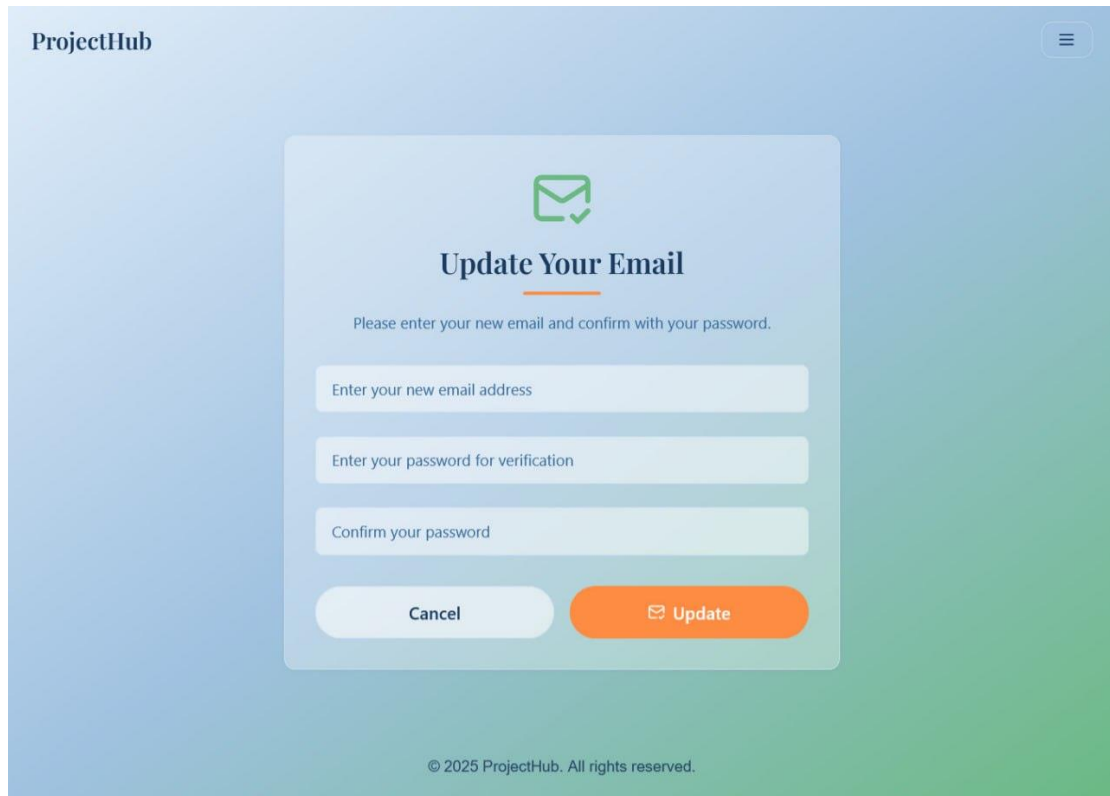


Figure 3.5: Update Email Page

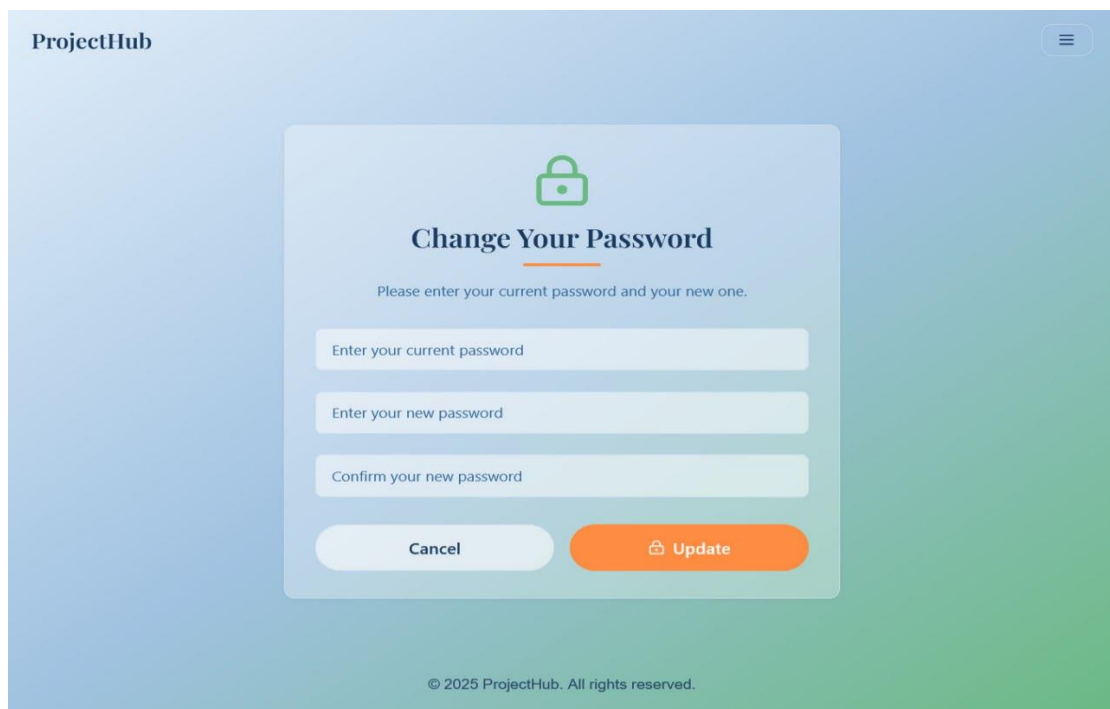
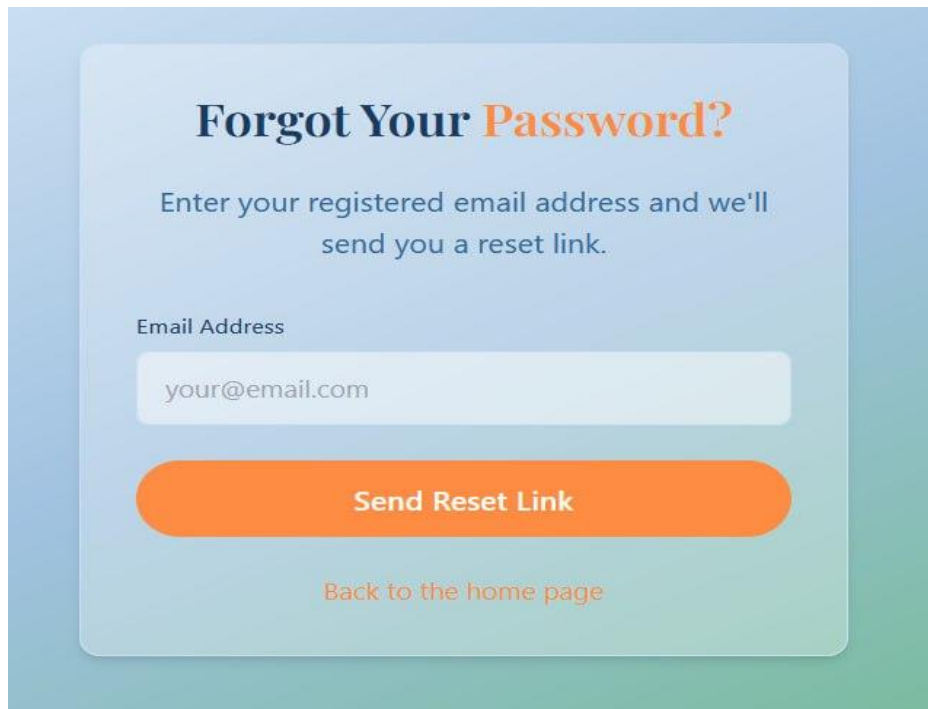


Figure 3.6: Change Password Page



Forgot Your Password?

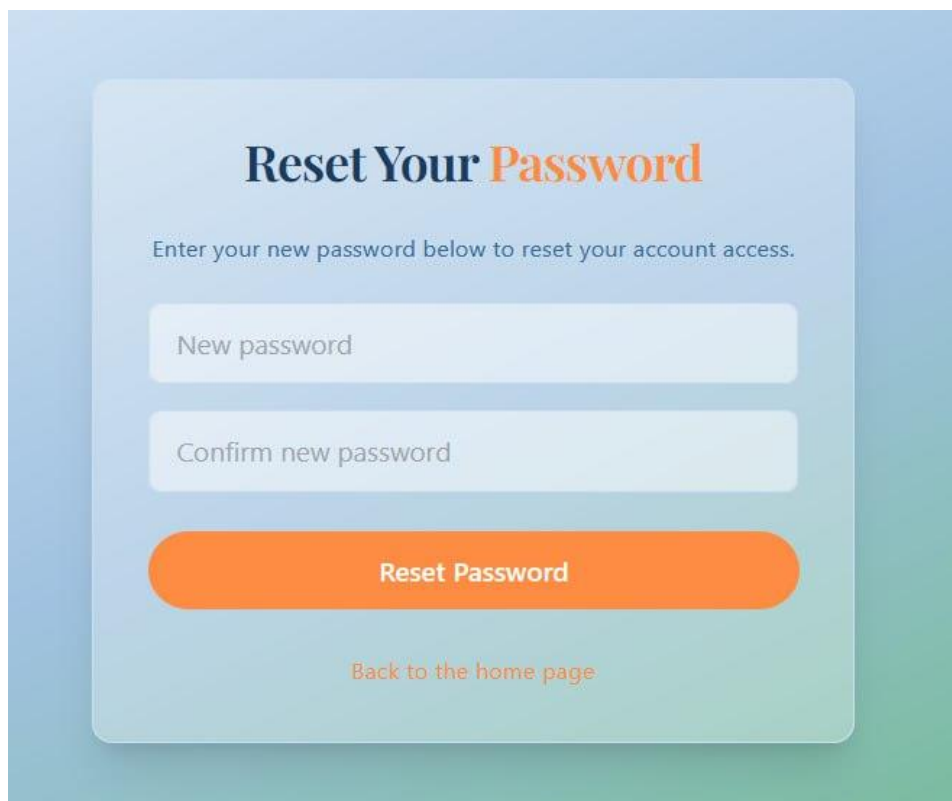
Enter your registered email address and we'll send you a reset link.

Email Address

Send Reset Link

[Back to the home page](#)

Figure 3.7: Forgot-Password Page



Reset Your Password

Enter your new password below to reset your account access.

Reset Password

[Back to the home page](#)

Figure 3.8: Reset Password Page

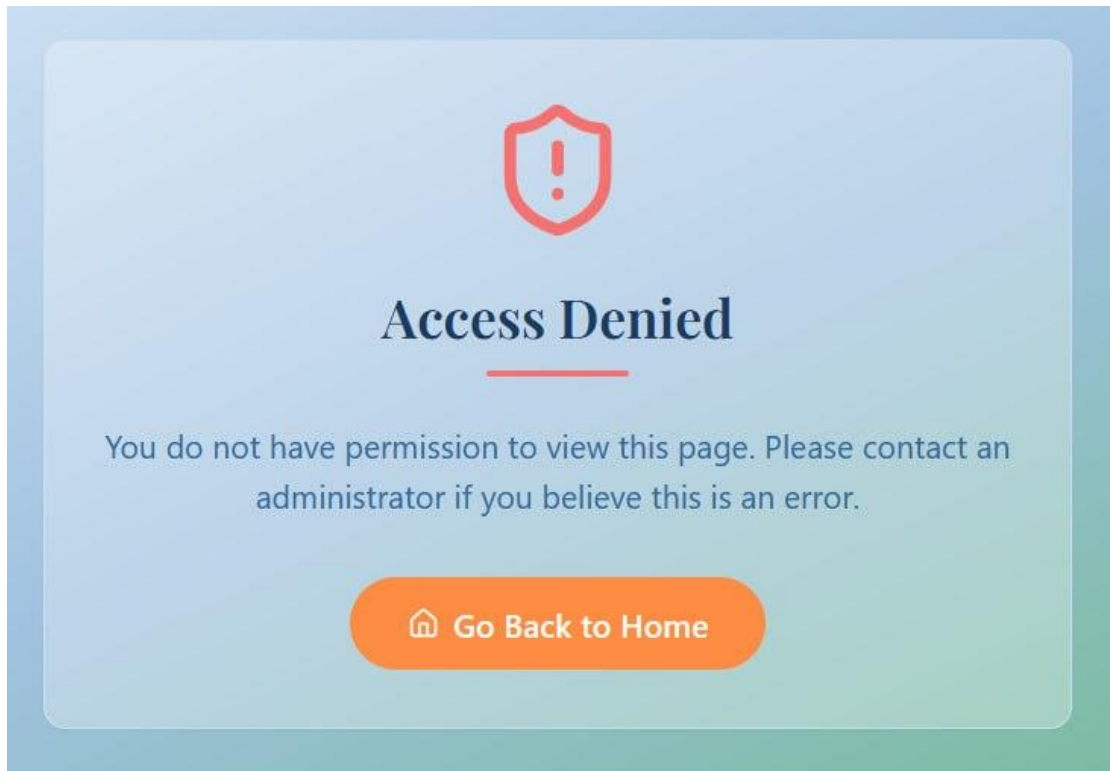


Figure 3.9: Access Denied Page

Super-admin's Interfaces:

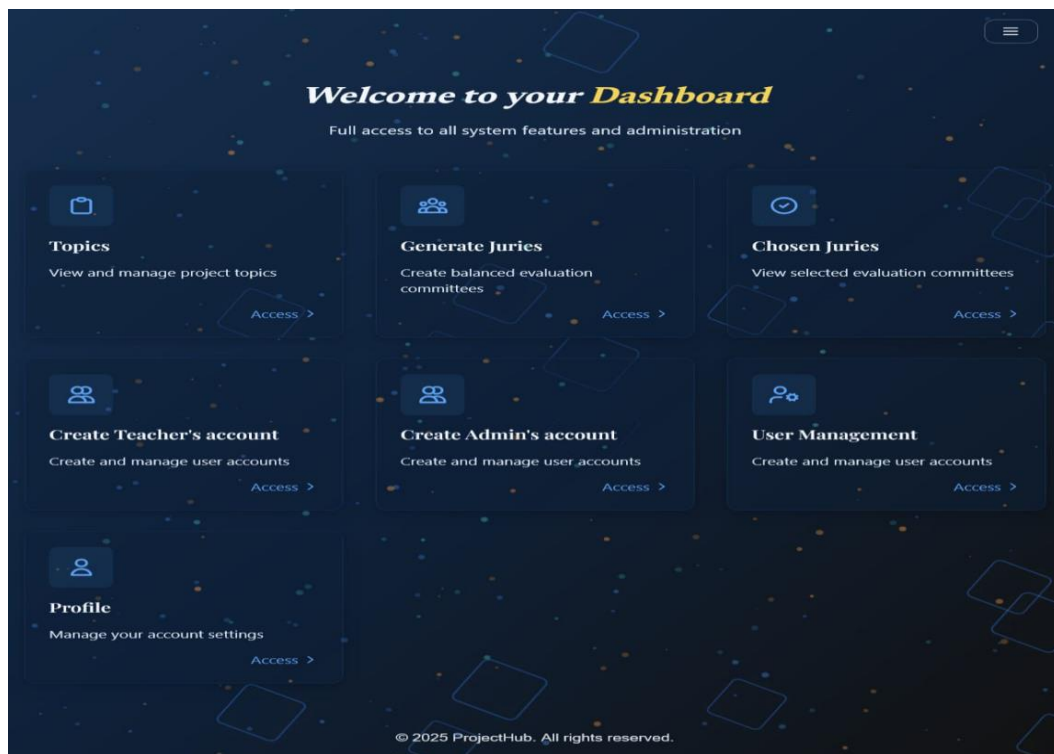
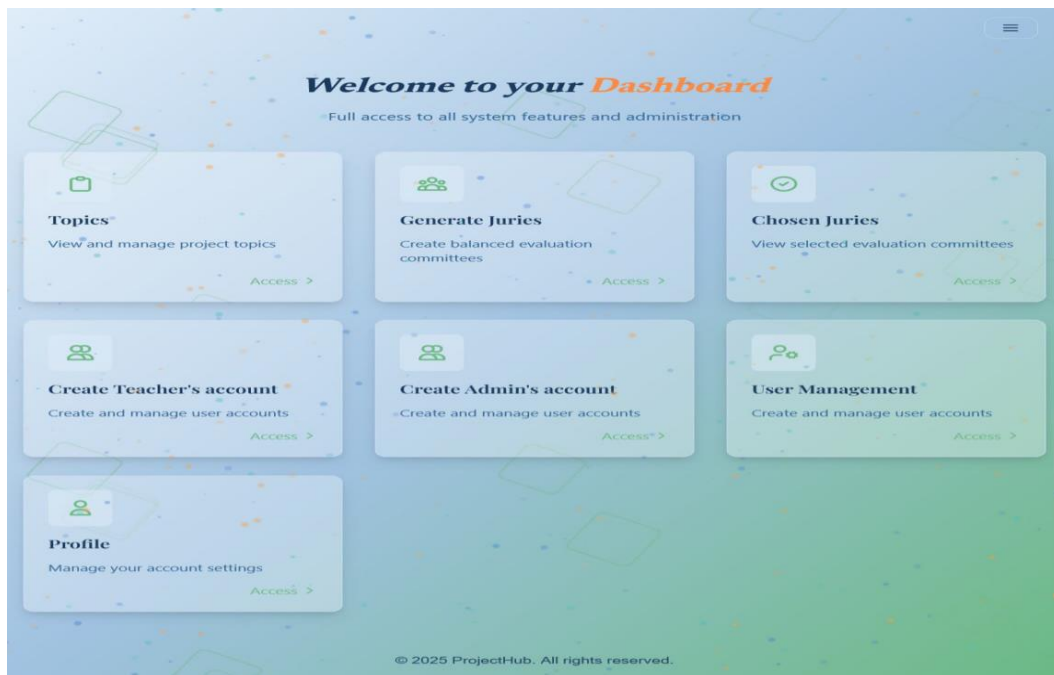
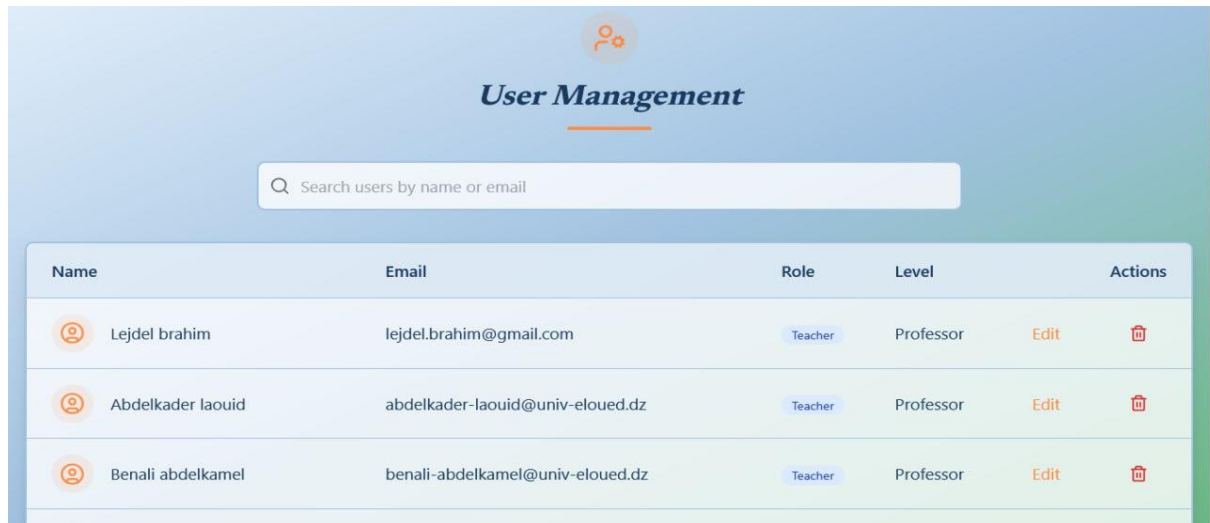


Figure 3.10: Super-admin's Dashboard (Home page after signing in)









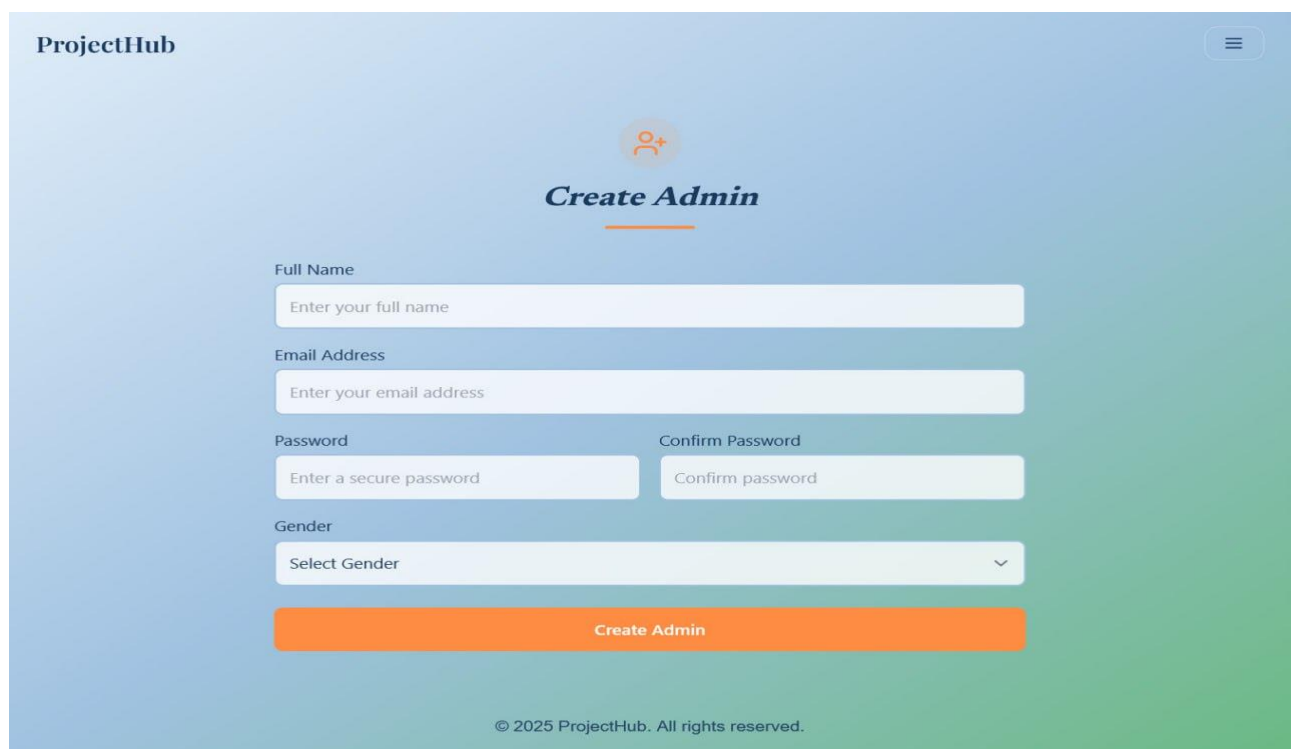
Name	Email	Role	Level	Actions
 Lejdel brahim	lejdel.brahim@gmail.com	Teacher	Professor	Edit 
 Abdelkader laouid	abdelkader-laouid@univ-eloued.dz	Teacher	Professor	Edit 
 Benali abdelkamel	benali-abdelkamel@univ-eloued.dz	Teacher	Professor	Edit 

Figure 3.11: Super-admin User management



ProjectHub

Create Admin

Full Name
Enter your full name

Email Address
Enter your email address

Password
Enter a secure password

Confirm Password
Confirm password

Gender
Select Gender

Create Admin

© 2025 ProjectHub. All rights reserved.

Figure 3.12: Create Admin Page

Admin’s Dashboard and shared interfaces with the super-admin:

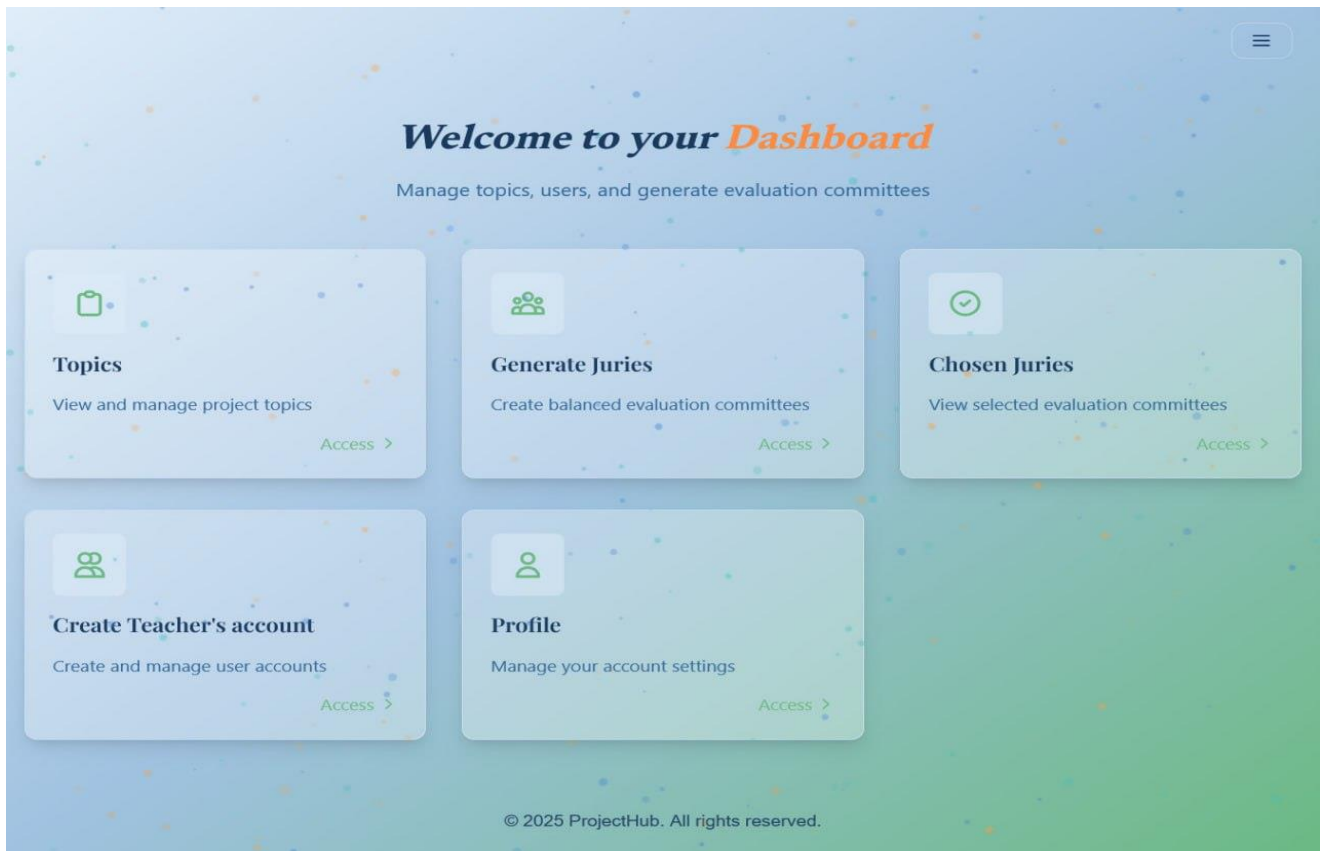


Figure 3.13: Admin’s Dashboard

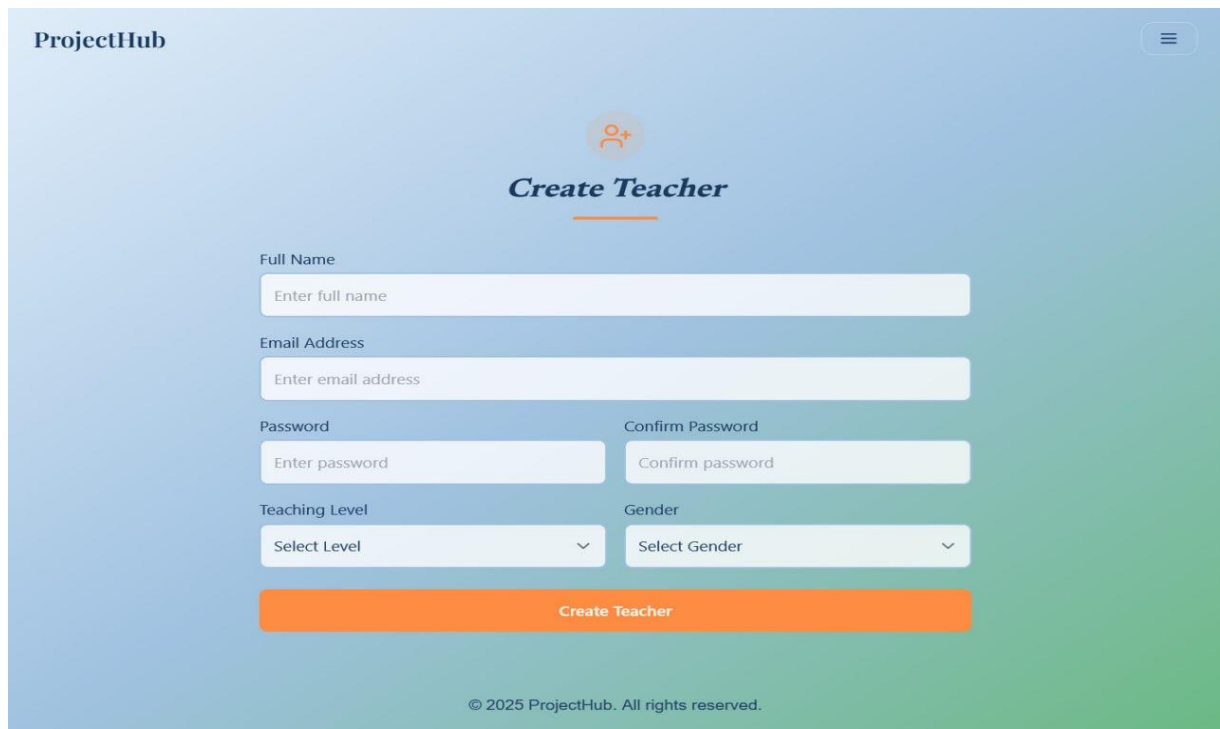


Figure 3.14: Create Teacher Page

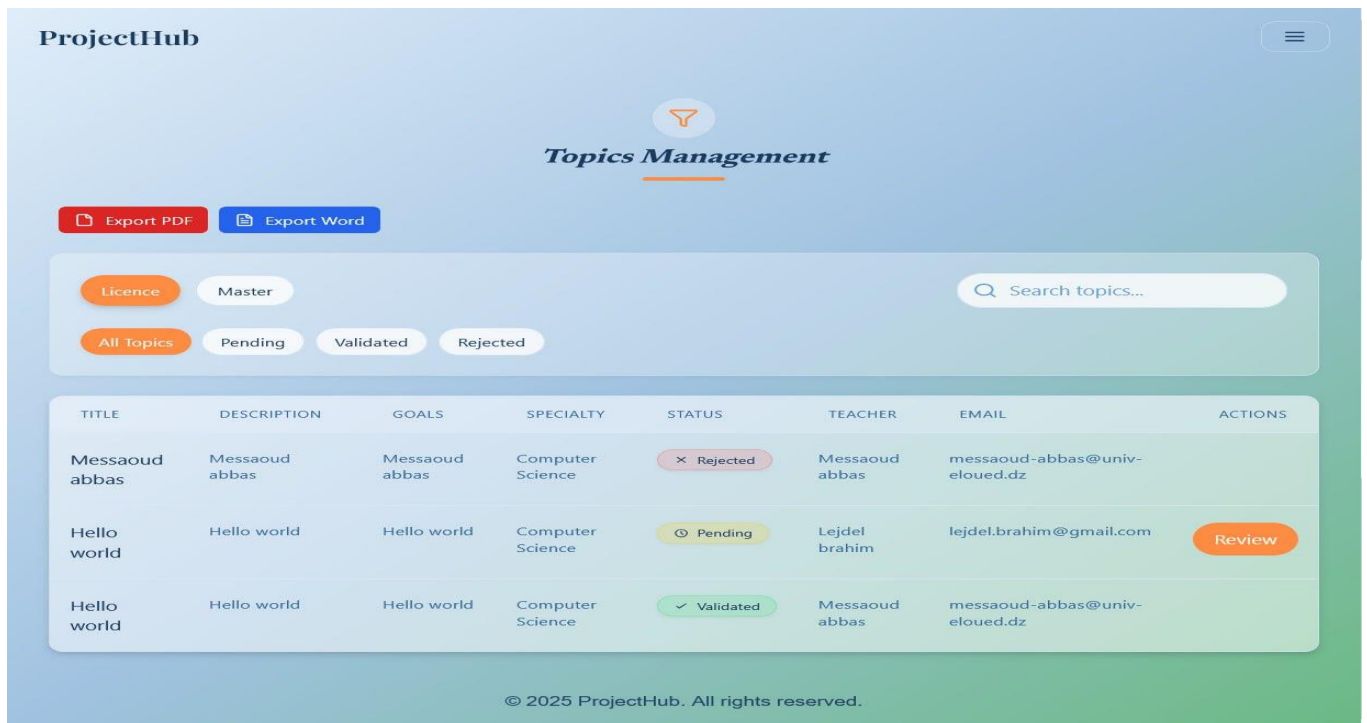


Figure 3.15: Topics Management Page

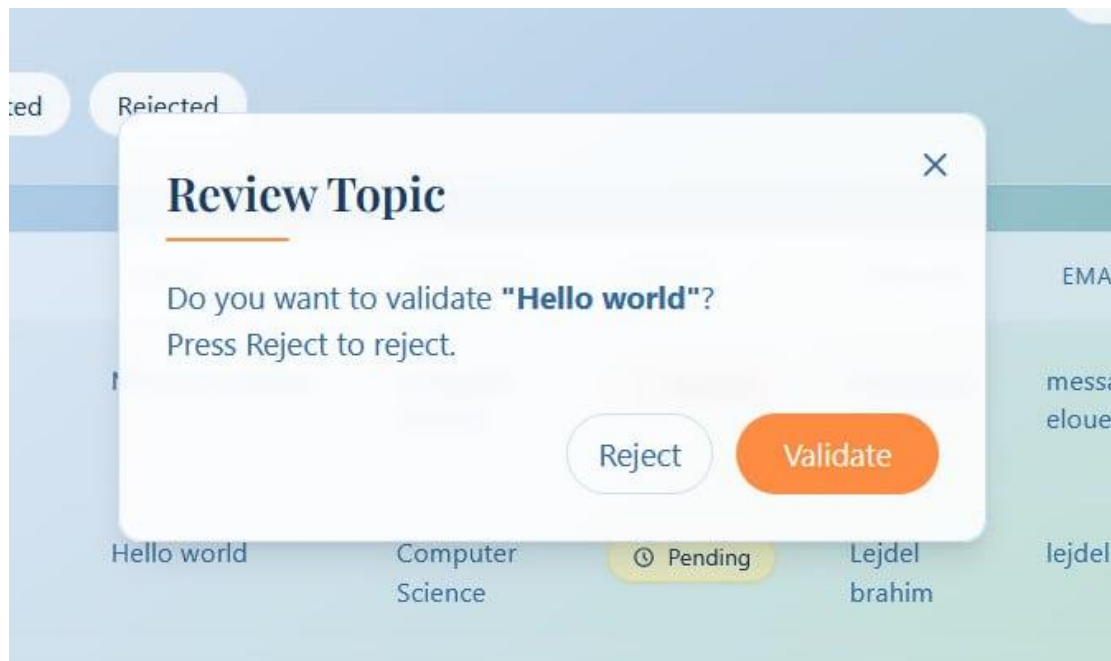


Figure 3.16: Topics Status Window

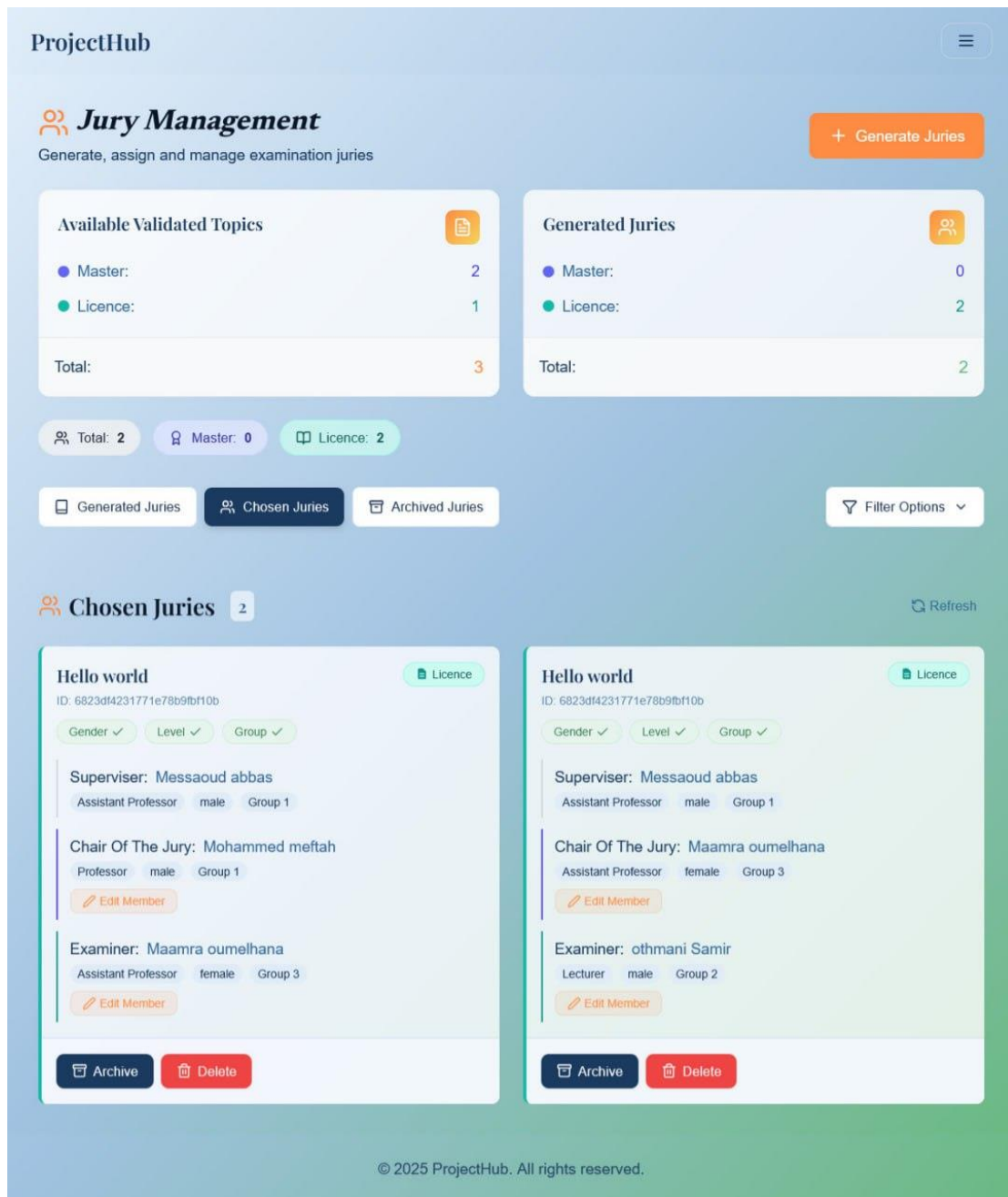









Figure 3.17: Jury Management Page

ProjectHub ☰


Chosen Juries Schedule

[Export PDF](#) [Export Word](#)

Topic	Level	Jury Members	Schedule	Actions
Chafik berdjouh	licence	supervisor: Chafik berdjouh (Assistant Professor) chair of the jury: Maamra oumelhana (Assistant Professor) examiner: Gherbi kaddour (Lecturer)	<input type="text" value="mm / dd / yyyy"/>  <input type="text" value="-- :-- --"/>  <input type="text" value="Enter location"/> 	Save
khelaifa abdennacer	licence	supervisor: khelaifa abdennacer (Assistant Professor) chair of the jury: Mohammed meftah (Professor) examiner: Settou tarablesse (Assistant Professor)	<input type="text" value="05 / 26 / 2025"/>  <input type="text" value="10 : 10 AM"/>  <input type="text" value="9"/> 	Save

© 2025 ProjectHub. All rights reserved.

Figure 3.18: Chosen Juries Schedule Page

Teacher's interfaces:

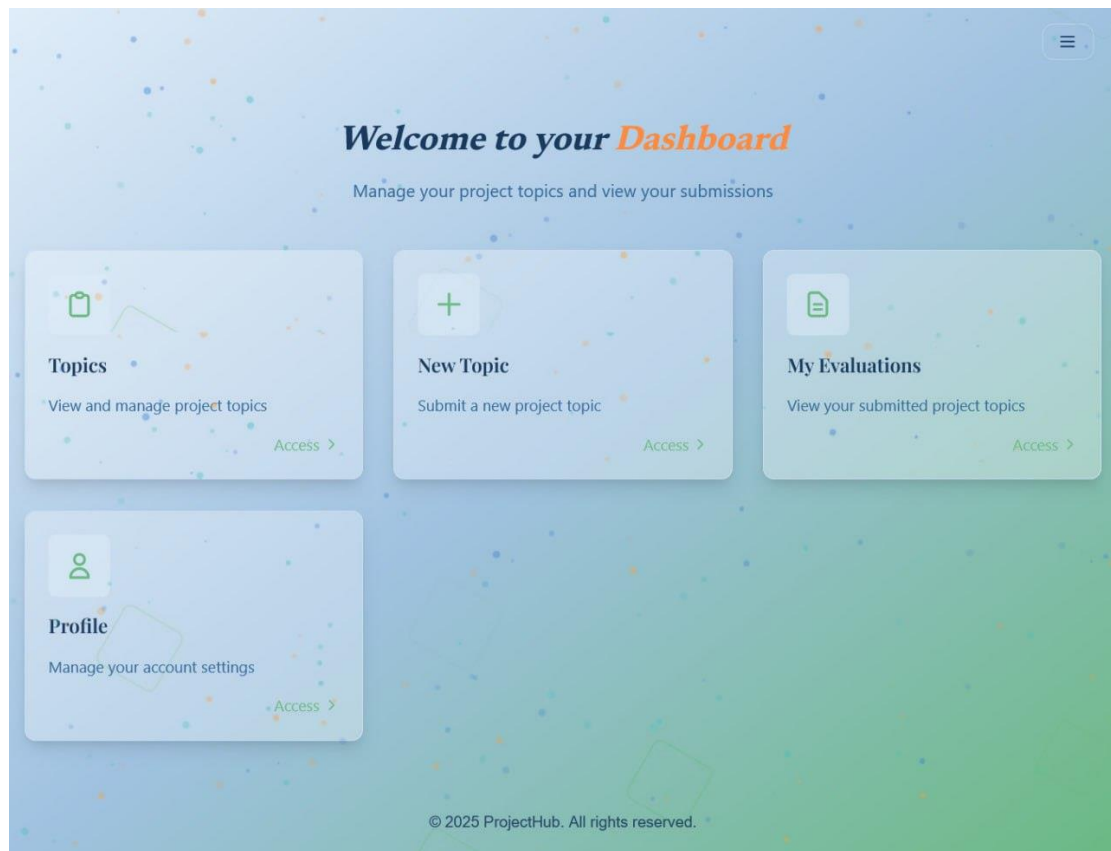


Figure 3.19: Teacher's Dashboard

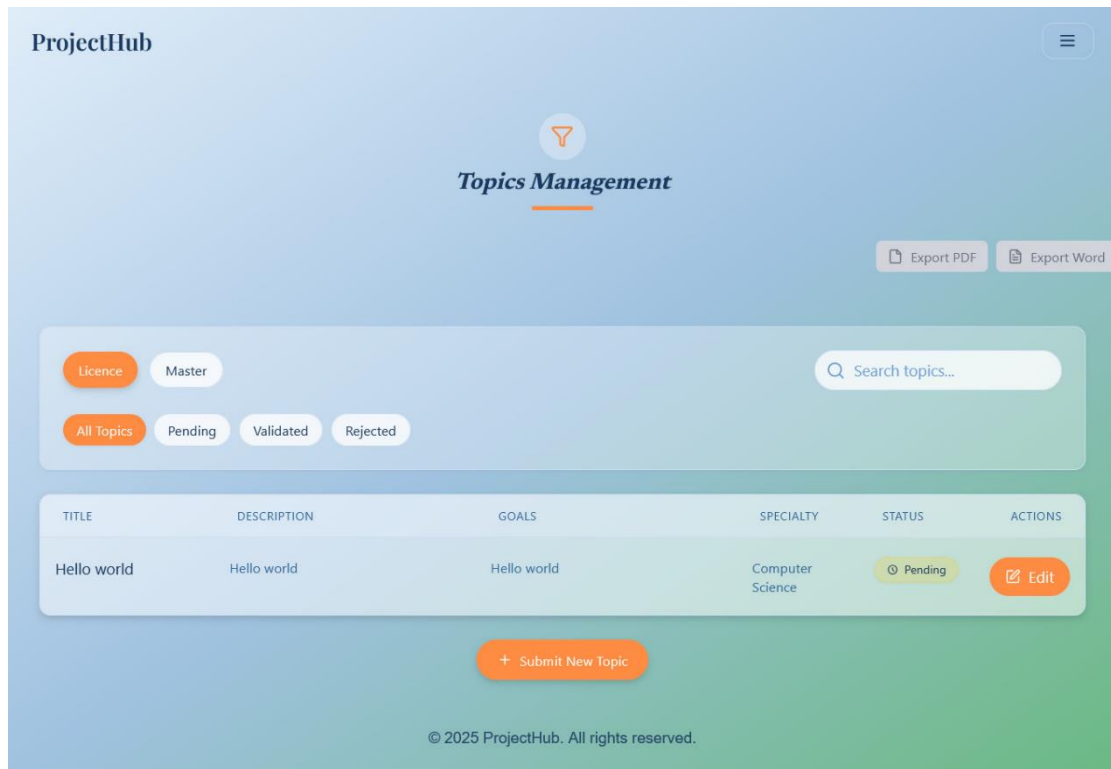
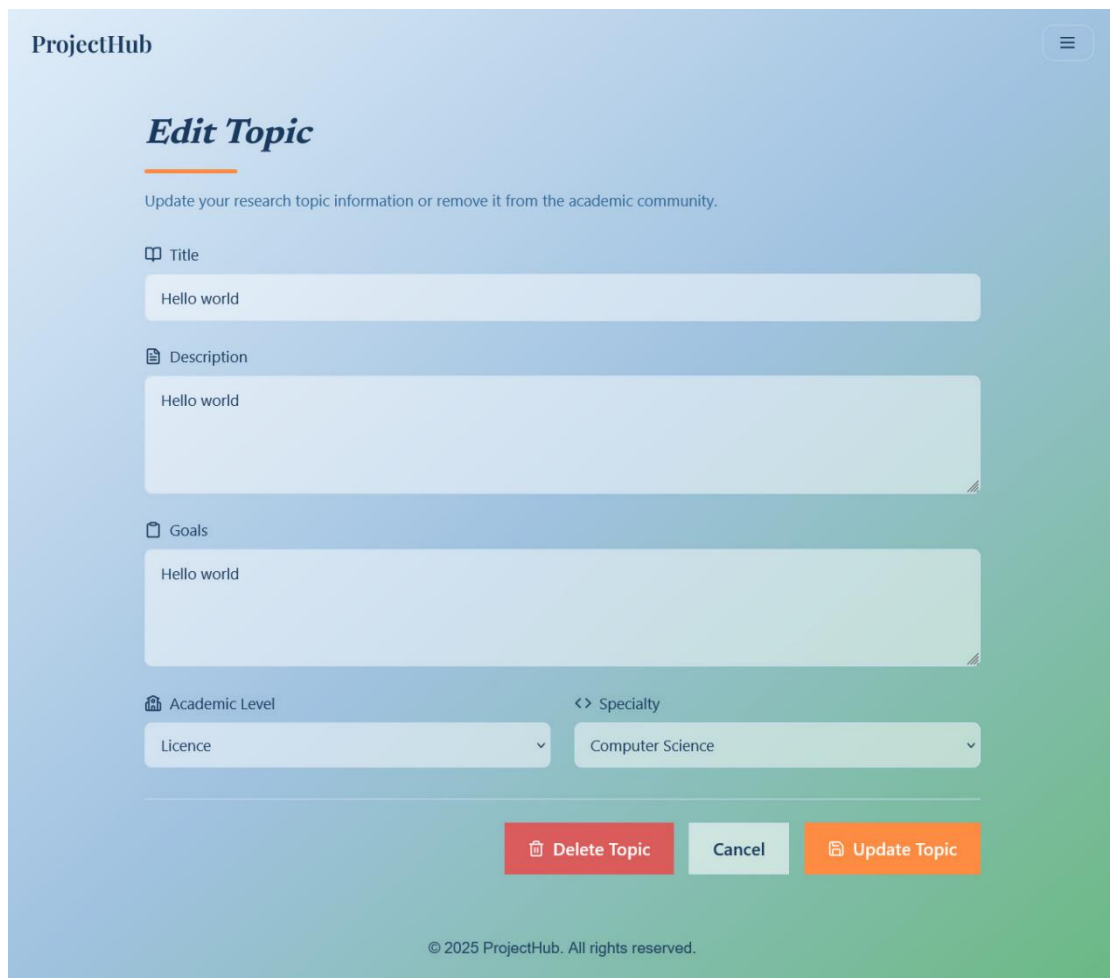


Figure 3.20: Topics Management Page For Teachers

The screenshot shows a web form titled "Submit a New Topic" on the ProjectHub platform. The form is set against a light blue background with a green gradient at the bottom. It contains the following elements:

- Title:** A text input field with the placeholder "Enter a clear, descriptive title for your topic".
- Description:** A larger text area with the placeholder "Provide a detailed description of your topic, including its significance and relevance".
- Goals:** A text area with the placeholder "List the main goals and objectives of your research topic".
- Academic Level:** A dropdown menu with the placeholder "Select Level".
- Specialty:** A dropdown menu with the placeholder "Select Specialty".
- Buttons:** A "Cancel" button and a "Submit Topic" button (highlighted in orange).
- Footer:** A copyright notice: "© 2025 ProjectHub. All rights reserved."

Figure 3.21: Submit New Topic Page



ProjectHub

Edit Topic

Update your research topic information or remove it from the academic community.

Title
Hello world

Description
Hello world

Goals
Hello world

Academic Level
Licence

Specialty
Computer Science

Delete Topic Cancel Update Topic

© 2025 ProjectHub. All rights reserved.

Figure 3.22: Edit / Delete Topic Page

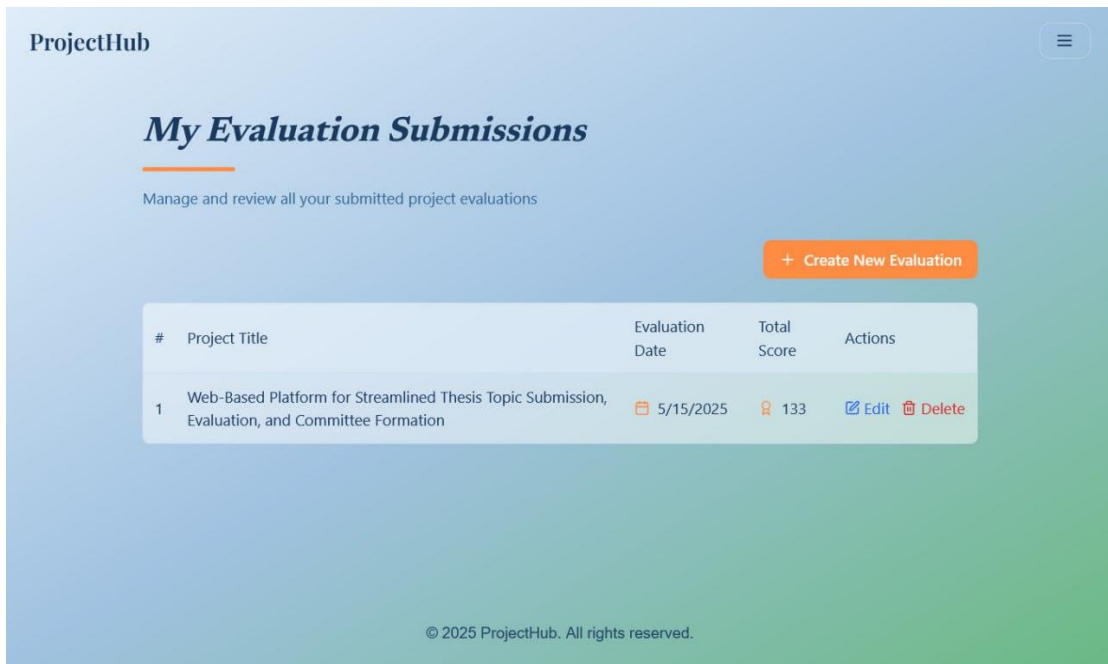


Figure 3.23: Teacher's Evaluations Forms

ProjectHub

Project Evaluation Form

Submit your evaluation for student projects. Provide detailed feedback to help students understand their strengths and areas for improvement.

Student Names
Enter the names of all students involved in the project

Project Level **Evaluation Date**
Select Level

Project Title
Enter the title of the project

Supervisor Name
Enter the name of the project supervisor

Evaluation Criteria

#	Criteria	Score (0-10)	Comment
1	Seriousness and Commitment	0	Comment
2	Continuity of Work	0	Comment
3	Compatibility with Objectives	0	Comment
4	Attendance in Supervisory Meetings	0	Comment
5	Communication and Collaboration (Gro	0	Comment
6	Technical Knowledge and Application	0	Comment
7	Innovation and Creativity	0	Comment
8	Documentation Quality	0	Comment
9	Presentation and Defense to work comj	0	Comment
10	Completion of Objectives Compared to	0	Comment
11	Ability to Work Independently	0	Comment
12	Use of Tools/Technologies	0	Comment
13	Problem Solving and Critical Thinking	0	Comment
14	Research and Literature Review	0	Comment
15	Adherence to Timeline and Deadlines	0	Comment

Total Score: 0 / 150

General Comments
Provide overall feedback on the project

© 2025 ProjectHub. All rights reserved.

Figure 3.24: Submit Evaluation Form

ProjectHub

Edit Evaluation Form

Update your evaluation for student projects. Provide detailed feedback to help students understand their strengths and areas for improvement.

Bey heyam
 Zouari ahmed lina

Select Level

05 / 15 / 2025

Web-Based Platform for Streamlined Thesis Topic Submission, Evaluation, and Committee Formation

Enter the name of the project supervisor

Evaluation Criteria

#	Criteria	Score (0-10)	Comment
1	Seriousness and Commitment	10	Comment
2	Continuity of Work	10	Comment
3	Compatibility with Objectives	10	Comment
4	Attendance in Supervisory Meetings	10	Comment
5	Communication and Collaboration (Gro	9	Comment
6	Technical Knowledge and Application	9	Comment
7	Innovation and Creativity	10	Comment
8	Documentation Quality	10	Comment
9	Presentation and Defense to work comj	9	Comment
10	Completion of Objectives Compared to	10	Comment
11	Ability to Work Independently	2	Comment
12	Use of Tools/Technologies	7	Comment
13	Problem Solving and Critical Thinking	7	Comment
14	Research and Literature Review	10	Comment
15	Adherence to Timeline and Deadlines	10	Comment

Total Score: 133 / 150

Provide overall feedback on the project.

© 2025 ProjectHub. All rights reserved.

Figure 3.25: Edit Evaluation Form

Some of the pages in dark mode:

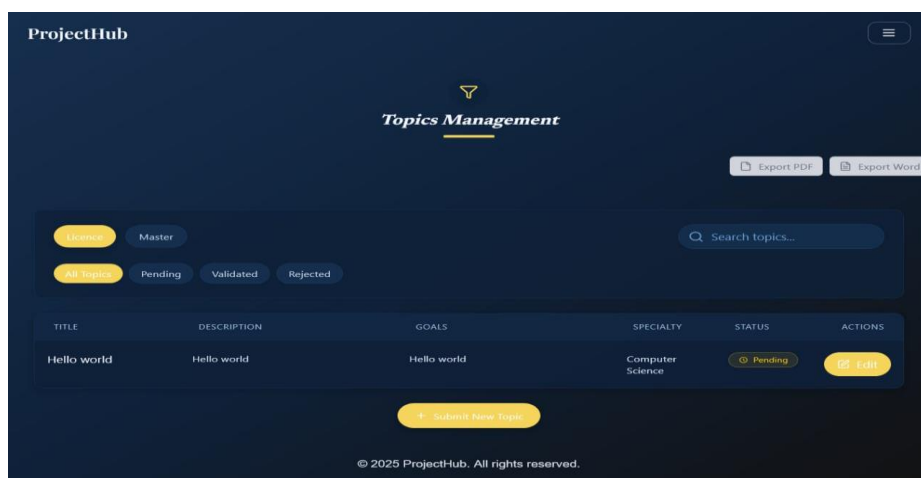
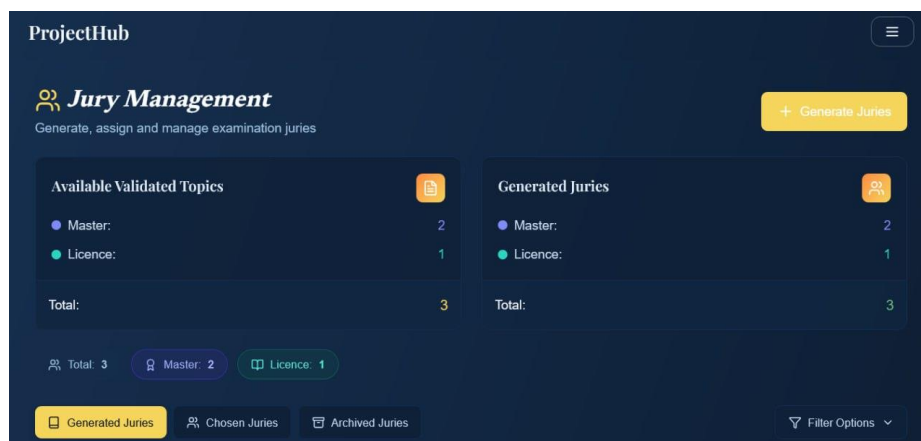
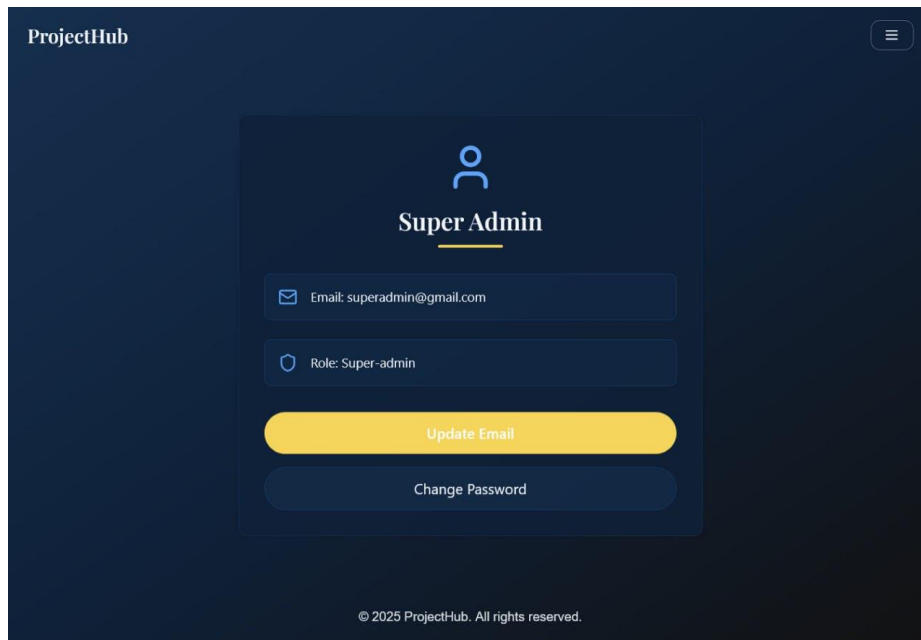


Figure 3.26: Pages In Dark Mode

Conclusion:

After building all the required and desired parts of the system, we now have a complete platform that works smoothly for managing Final Year Projects. This achievement reflects the goals of our project and turned our hard work into reality.

General Conclusion

General Conclusion:

Through this project, we aimed to modernize how Final Year Projects are managed in our academic system by replacing the traditional manual process with a more efficient web-based platform. By building this system using the MERN stack, we were able to create a solution that centralizes all the necessary tasks— project submission, evaluation, and jury assignment—into one accessible online space. The platform not only simplifies the workflow for professors and administrators but also helps ensure fairness, transparency, and time efficiency. Working on this project allowed us to apply both technical and problem-solving skills in a real-world context, and we're proud of the impact this system could have at our university.

References

References:

[1]*UML:

<https://www.omg.org/spec/UML/About-UML/>

[2]*JavaScript:

<https://www.javascript.com/>

[3]*React:

<https://react.dev/>

[4]*Node.js:

<https://nodejs.org/en>

[5]*Express.js:

<https://expressjs.com/>

[6]*MongoDB:

<https://www.mongodb.com/>

[7]*Mongoose:

<https://mongoosejs.com/>

[8]*NPM:

<https://www.npmjs.com/>

[9]*Tailwindcss:

<https://tailwindcss.com/>

[10]*Vite:

<https://vite.dev/>

[11]*Git:

<https://git-scm.com/>

[12]*Github:

<https://github.com/>

[13]*VS code:

<https://code.visualstudio.com/>

References

[14]*StarUml:

<https://staruml.io/>

[15]*ThunderClient:

<https://www.thunderclient.com/>

[16]*Resend:

<https://resend.com/>

[17]*Role Based Access control:

<https://www.ibm.com/think/topics/rbac>

[18]*MERN Stack:

<https://www.imaginarycloud.com/blog/what-is-mern-stack-and-how-does-it-work>

[19]*MVC architecture:

<https://www.techtarget.com/whatis/definition/model-view-controller-MVC>

[20]*HTML:

<https://html.com/>

[21]*CSS:

<https://html.com/css/>

References
