

Comp3900 Computer Science Project

Final report

Written by team 3900F13Aikun

Project: P21 Student Industry Project Management System

Project submission data: 17 November 2023 @ 9.00pm

Team member:

- Jiayang Jiang (z5319476), role: **scrum master**, email: z5319476@ad.unsw.edu.au
- Jian Yao (z5319493), role: **developer**, email: z5319493@ad.unsw.edu.au
- Chenkai Shen (z5324155), role: **developer**, email: z5324155@ad.unsw.edu.au
- Kang Liu (z5346398), role: **developer**, email: z5346398@ad.unsw.edu.au

Table of contents

I. Overview-----	3
System architecture-----	3
User Interface Design-----	7
II. System functionalities -----	12
Functionality description and how to use them-----	12
Functionality 1:User Authentication-----	13
Functionality 2: User Profile-----	16
Functionality 3: Recommendation-----	18
Functionality 4: Student Group-----	21
Functionality 5: Opportunity Hunt-----	24
Functionality 6: Project-----	27
III. Third-party functionalities-----	37
Flask -----	37
Flask-JWT-Extended -----	37
Flask-Mail -----	37
Flask-SQLAlchemy -----	38
Flask-Marshmallow -----	38
React-router-dom -----	38
Material UI -----	38
Data Structure -----	38
Route -----	42
IV. Implementation challenge-----	45
V. Installation/Userdocument/manual:-----	46
Environment-----	46
Run server and source code-----	46

Overview

The Ikun Student Industry Project Management System stands out as a user-friendly web application strategically crafted to serve as a collaborative hub meeting the diverse but overlapping needs of industry partners, students, and academic supervisors in the following aspects:

- Industry partners: Ikun provides an efficient avenue for accessing support from a diverse pool of students for particular projects. This enables industry partners to take advantage of the innovative solutions and capabilities that students bring to the table, transforming them into mature products that can effectively meet market demands.
- Students: Ikun serves as a crucial pathway for the accumulation of practical experience. By actively engaging in projects, students not only make meaningful contributions to real-world tasks but also strengthen their resumes with concrete examples of hands-on involvement and the application of multiple skills required for the projects.
- Academic supervisors: Ikun is a valuable tool that enables academic supervisors to steer students on the right path when delivering their products. Throughout this process, academic supervisors can also showcase their skills and capabilities to industry partners, thereby strengthening and expanding their professional networks and connections to prepare for potential cooperation in the future.

System Architecture

In the dynamic landscape of academic-industry collaboration, Ikun is committed to successfully delivering a system that meets the diverse needs of three key stakeholders over a ten-week product delivery cycle. To realize this objective, Ikun has embraced a robust three-layered architecture, encompassing the Interface Layer, Business Layer, and Database Layer. This architectural and bottom-up approach strategically divides the application into more manageable units. This not only enhances efficiency but also facilitates implementations across various aspects of the system. Further details are illustrated in Diagram 1,

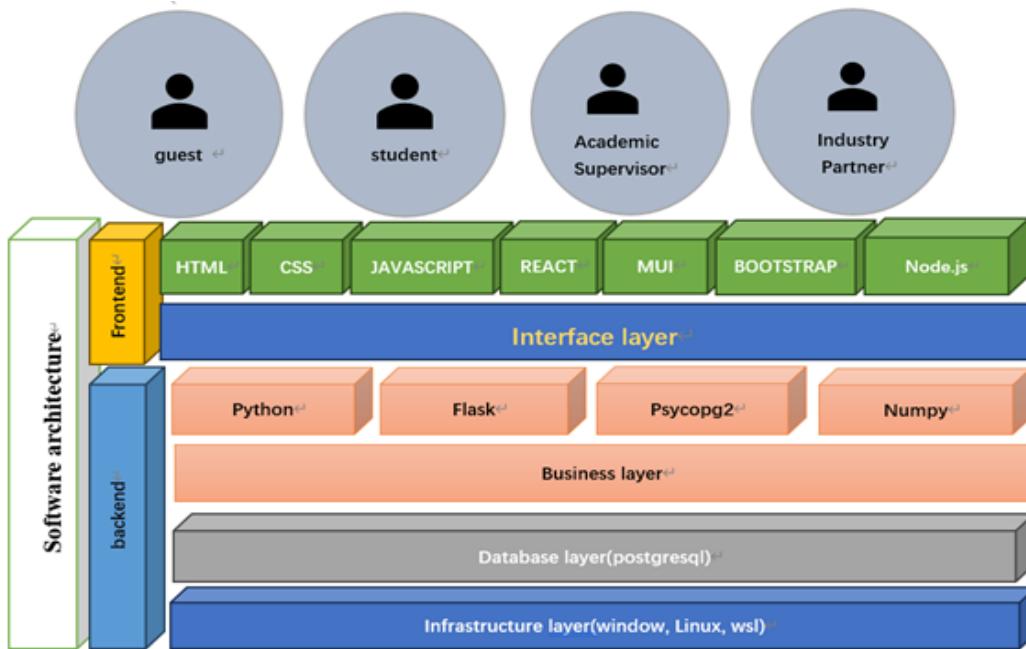


Diagram 1: System Architecture

Interface Layer

The Interface Layer takes center stage in our architecture, as it connects the system and users as a forefront layer. It serves as the interface for the Ikun system to gather input from users and present stored information, acting as a direct channel for interaction with all four potential external actors: industry partners, students, academic supervisors, and guests (users who haven't logged into the application). Within this layer, three primary user types (except guests) can access most of the functionalities, with exceptions designed for specific user types, such as only students having the exclusive ability to join or leave a team.

The Interface Layer of the Ikun system is intricately shaped by a synergistic blend of technologies to comprehensively improve the user experience, each of them playing a pivotal role in delivering a seamless and engaging user experience.

- **HTML**
 - HTML stands as the cornerstone of the presented contents on the Ikun system of the application, and it plays a fundamental role in forming the layout, structure, and content of the user interface for various web elements, such as text, images, forms, and other interactive elements.
- **CSS**
 - CSS functions as a robust tool, enhancing the visual Interface of the HTML elements we applied. It empowers our team to control the layout, color, font, and overall aesthetics of the user interface. This ensures a uniform and visually appealing design that remains consistent across different browsers.
- **JavaScript**

- JavaScript serves as the driving force behind interactivity and responsiveness on web pages, providing the essential capability for the system to engage with users. Within the Interface Layer, JavaScript helps us generate dynamic content, manage user inputs, and implement various client-side functionalities, which enhances the overall user experience.
- **React**
 - React is a JavaScript library dedicated to building user interfaces. It empowers our team to create reusable UI components. React facilitates the development of complex user interfaces through a modular and structured approach, enabling the creation of interactive and dynamic web views.
- **MUI**
 - MUI acts as a mature React UI framework, providing a collection of pre-designed and customizable React components within the Interface Layer. These components follow a cohesive design language, offering a streamlined development process by supplying ready-to-use UI elements and reducing our workload.
- **Bootstrap**
 - Bootstrap, as a front-end framework, simplifies our web development process by offering a curated selection of pre-designed components and styles. These include elements like the grid system and navigation components, seamlessly interacting with the technologies mentioned above (HTML, CSS, and JavaScript). Its contribution effectively enhances the visual appeal and responsiveness of the user interface.

Throughout this channel, the Ikun system can offer a seamless and user-friendly experience, facilitating effortless registration, login processes, and engagement with the platform's diverse functionalities.

Business Layer

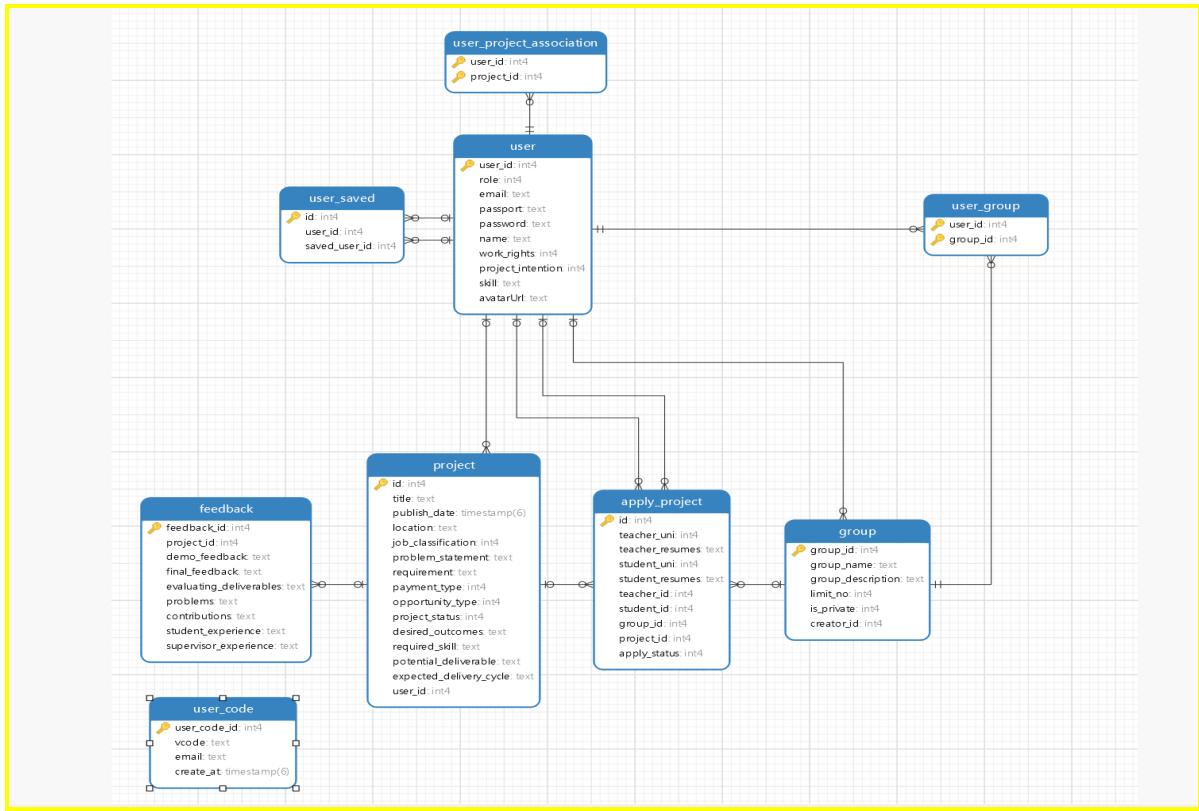
The central hub of the system architecture is the crucial Business Layer, strategically positioned between the Interface and Database Layers. Functioning as the core engine, this layer smoothly converts incoming requests or raw data from the Interface Layer into meaningful backend data. It processes these inputs, along with potential data from the Database Layer, following the system's logic and functionalities, thereby enabling the Ikun system to provide meticulously designed features and keep critical data in the database. Serving as the key to the system architecture, the business layer facilitates seamless interaction among all three layers. It is essential for establishing a cohesive and functional application that effectively caters to user needs. Given the importance of the Business Layer, the Ikun system is dynamically powered by the Python programming language and the Flask web framework.

Python serves as the primary language for constructing the essential functionalities and logic that propel the system's backend operations. Its versatility makes it an optimal choice for implementing data processing and facilitating smooth interactions between the Interface Layer and the Data Layer. In conjunction with Python, Flask, a modular web framework, assumes a pivotal role in organizing the backend of the application. Flask streamlines the development of robust web applications by furnishing a suite of tools and utilities for routing, managing requests from the frontend (i.e. Interface Layer), responding with outputs generated by the Business Layer functionalities in the backend, and overseeing sessions. With its minimalistic design and flexibility, Flask is well-suited for constructing the Business Layer. Together, Python and Flask form a dynamic duo within the Business Layer, ensuring the effective execution of business logic, handling data transactions, and facilitating the seamless flow of information between the frontend and backend components. This synergistic combination empowers the Ikun system to provide a responsive, scalable, and high-performance application experience for users, fostering a dynamic and user-friendly environment.

Database Layer

The Database Layer in the Ikun system serves as a foundational component, intricately integrated into the architecture to handle data storage and retrieval. Positioned alongside the Business Layer, it plays a crucial role in storing and managing critical information for the system's functionalities. As a repository for data, the Database Layer facilitates seamless communication between the frontend and backend, ensuring that the Ikun system operates with efficiency and accuracy.

To guarantee the proper functioning of the system, we designed a database in accordance with the ER diagram depicted in diagram 2. Its integration with the Business Layer enables the processing of raw data into meaningful insights, facilitating the delivery of meticulously designed features while maintaining the integrity of vital data.



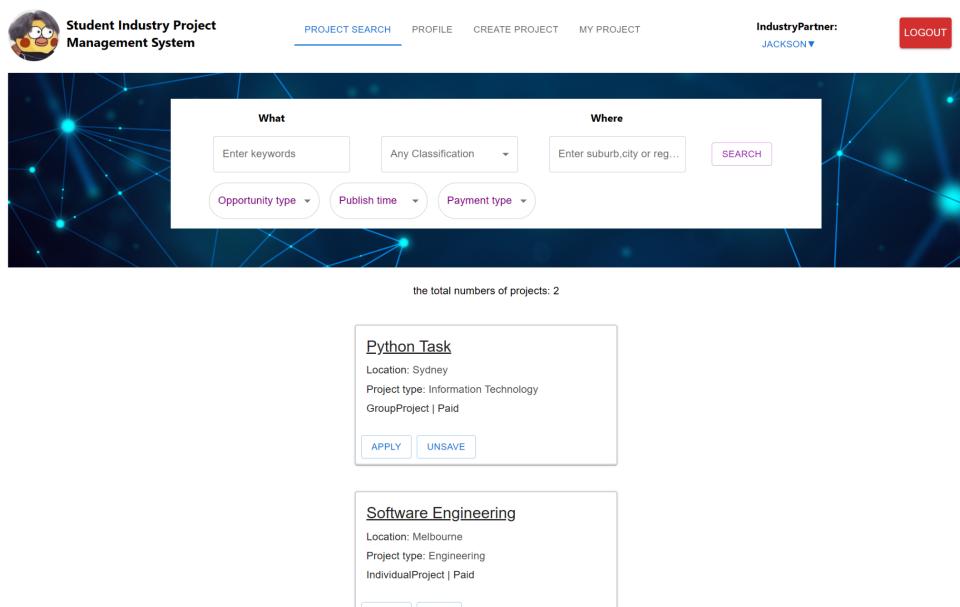
As a result, to meet the Ikun system's data type requirements and support a high volume of queries across various relationship tables, PostgreSQL is employed. As an object-relational database, it allows for the efficient storage and organization of the structural data needed by the system functions, fostering seamless interactions with the Business Layer. With its support for complex queries and transactions, powered by a built-in query optimizer and just-in-time compilation, PostgreSQL enhances the system's capacity to process and manage data. Furthermore, its adherence to ACID principles ensures data integrity and reliability, making it well-suited for critical applications. In summary, the integrated Data Layer and PostgreSQL serve as a reliable reservoir for structured data, significantly enhancing the overall functionality and responsiveness of the Ikun system.

User Interface Design

User interface (UI) design is an indispensable and user-centric component that resides in the Interface Layer and focuses on creating a visually appealing, intuitive, and user-friendly graphical user interface. It plays a significant role in shaping the overall user experience, ensuring that students, academic supervisors, and industry partners can effortlessly navigate through the system information, therefore allowing them to utilize diverse functionalities of the system smoothly.

- **Concise and Intuitive Web Page Structure**

At the heart of our UI design is a concise and intuitive web page structure. This design approach avoids unnecessary visual distractions, ensuring users can efficiently navigate the system, especially when dealing with substantial information, such as browsing projects. We prioritize supplying only essential elements in each webpage section, thus also reducing the user's burden. For instance, in the project browse section (as illustrated in screenshot 1). Since this structure supports effective communication between the system and users, it can potentially lead to users engaging with the system more actively with higher user satisfaction.



Screenshot 1: Project Browse

- **Perceivable and Understandable Design:**

A user-friendly UI design embraces a perceivable and understandable design, which is vital for enhancing user experience.

We prioritize users' concerns by implementing a consistent design style across all pages, eliminating the need for them to adapt to different layouts. For instance, Screenshots 2 and 3, presented, showcase different web pages within our application. Despite their distinct purposes, both exemplify a unified design concept, reflecting the consistent and cohesive visual identity we maintain across various sections of our application.

The screenshot shows the User Dashboard of the Student Industry Project Management System. At the top, there is a navigation bar with links for PROJECT SEARCH, PROFILE, MY GROUP, and MY PROJECT. On the right, there is a user profile icon labeled "Student: ALLEN" and a red "LOGOUT" button. Below the navigation bar is a search interface with fields for "What" (Enter keywords, Any Classification, Enter suburb/city or reg...), "Where" (Opportunity type, Publish time, Payment type), and a "SEARCH" button. The background features a dark blue network-like pattern. Below the search bar, there are three sections: "Recommended Projects" (listing "Python Task" and "Software Engineering"), "Saved Projects" (listing "Python Task"), and "Recommended Academic Supervisors" (listing "Allen"). Each project and supervisor entry includes a "VIEW ALL" button.

Screenshot 2: User Dashboard

The screenshot shows the Project Recommendation section of the system. At the top, there is a navigation bar with links for PROJECT SEARCH, PROFILE, CREATE PROJECT, and MY PROJECT. On the right, there is a user profile icon labeled "IndustryPartner: JACKSON" and a red "LOGOUT" button. Below the navigation bar is a header "Recommend Lists". Underneath, there are two tabs: "RECOMMENDED PROJECTS" (selected) and "RECOMMENDED ACADEMIC SUPERVISORS". A message states "The total numbers of recommend projects: 2". Below the tabs, there are two project cards: "Python Task" and "Software Engineering". Each card displays the project name, location, project type, and status (e.g., GroupProject | Paid). There are also "UNSAVE" and "SAVE" buttons next to each card.

Screenshot 3: Project Recommendation

Furthermore, our navigation bars and web buttons are meticulously crafted with predictability in focus, enabling users to navigate effortlessly without any prerequisite knowledge. As an illustration, the button positioned next to the project search bar is exclusively dedicated to assisting users in finding projects that align with their specified criteria. This intentional design ensures a user-friendly and intuitive experience throughout the platform.

To improve clarity, we carefully adjusted letter expression, size, color, and contrast. This distinction helps users easily navigate the various functional areas and hierarchies within the system. Employing this design philosophy extends to “create

project” button creation at screenshot 4, emphasizing its significance through all-caps letters and a dark color scheme with high contrast. Despite its smaller size relative to the primary content on the web page, this intentional design choice directs users' attention to the key focal points of the page.

The screenshot shows a web-based project management system. At the top, there's a navigation bar with links for 'PROJECT SEARCH', 'PROFILE', 'CREATE PROJECT' (which is underlined in blue), and 'MY PROJECT'. On the right side of the header, it says 'IndustryPartner: JACKSON' and has a 'LOGOUT' button. Below the header, a dark blue banner reads 'My Create Projects'. Underneath, there's a message stating 'The total numbers of projects: 2'. Two project cards are listed: 'Python Task' (Location: Sydney, Project type: Information Technology, GroupProject | Paid) and 'Software Engineering' (Location: Melbourne, Project type: Engineering, IndividualProject | Paid). Each card has 'EDIT' and 'DELETE' buttons at the bottom.

Screenshot 4: Project Management

Furthermore, we extensively utilize guiding text within interactive features to ensure users readily comprehend system requirements and potential responses. As depicted in Screenshot 5, the system provides clear prompts, guiding users on fulfilling the system's requirements for password creation and preventing them from confusion.

A modal dialog box titled 'localhost:3000 显示' (localhost:3000 Display). It contains text: 'The password must meet the following criteria:' followed by 'At least 8-digit long', 'At Least 1 number', and 'At Least 1 letter'. In the bottom right corner of the dialog, there is a blue button labeled '确定' (Confirm).

Screenshot 5: Password Creation Hint

Through the full combination of the above fields, our system can effectively improve users' efficiency and bring them a convenient experience.

- **Role-based Tailored Interface:**

To enhance the overall customer experience within the system, we have prioritized a role-based customized approach in our UI design. Recognizing the diverse demands of different user types, we understand that a one-size-fits-all design, while efficient in reducing workload, may compromise user efficiency. Tailoring the design to specific

roles eliminates unnecessary clutter, allowing users to navigate directly to content and features relevant to their tasks. This focused approach streamlines user interactions, reducing the time and effort required for task completion. Simultaneously, users are spared from unnecessary information, alleviating cognitive load and contributing to a more user-friendly experience.

As illustrated in screenshots 6 to 9, we have tailored distinct user navigation bars for various group types. For instance, industry partners (screenshot 6) with the ability to post projects can conveniently locate the interface in the navigation bar dedicated to project creation. Similarly, students (screenshot 7) have easy access to group functionalities, preparing them for group project applications. In contrast, academic supervisors (screenshot 8), who neither upload project opportunities nor supervise projects in a team, have the most concise navigation bars among the three parties, exactly the same as the design for guests (screenshot 9).



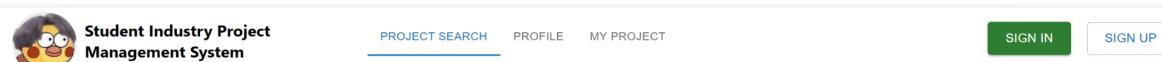
Screenshot 6: Industry Partner Navigation Bar



Screenshot 7: Student Navigation Bar



Screenshot 8: Academic Supervisor Navigation Bar



Screenshot 9: Guest Navigation Bar

By incorporating principles of consistency, clarity, and accessibility, our UI design strives to create an interface that not only meets the specific needs of each user group but also aligns with the overall objectives of the collaborative ecosystem.

System Functionalities

In order to ensure the system aligns with the customer's project requirements outlined in the specification, we devised 5 major functionality components on the Ikon System. These components seamlessly interact, providing users with an operational and comprehensive application that facilitates the fulfillment of their demands.

To streamline the presentation of requirements and avoid redundancy, the following section will adhere to the mapping table 1 presented.

Requirement Serial Number	Requirement
1	“Students, Academic Supervisors, Industry Partners to register and login using username and password credentials.”
2	“Industry Partners to post real-world problems and project opportunities directly on the system. Each problem should include details such as the problem statement, desired outcomes, required skills, and potential deliverables.”
3	“Students to browse through the posted industry problems and express their interest in specific projects.”
4	“Students to submit their resumes to demonstrate their suitability for the projects they are interested in.”
5	“Pairing Students and Industry Partners based on the analysis of students' profiles, skills, and the project intention.”
6	“The recommendation of suitable projects to all users.”
7	“The recommendation of suitable Academic Supervisors to Students and Industry”
8	“All users to save and unsave the posted projects, Students and Industry partners to save and unsave recommended supervisors”
9	“Academic Supervisors to view the projects and select the ones they are best equipped to supervise.”
10	“Academic Supervisors to provide feedback to improve the alignment between projects and students' learning goals.”
11	“Academic Supervisors to assess students' performance during the project period. This could include evaluating deliverables, problem-solving skills, and overall contributions to the project.”
12	“Industry Partners to rate their experience working with Students and Academic Supervisors.”

Table 1: Specification Requirement Mapping Table
Functionality description and how to use them

Functionality 1: User Authentication

- **Description**

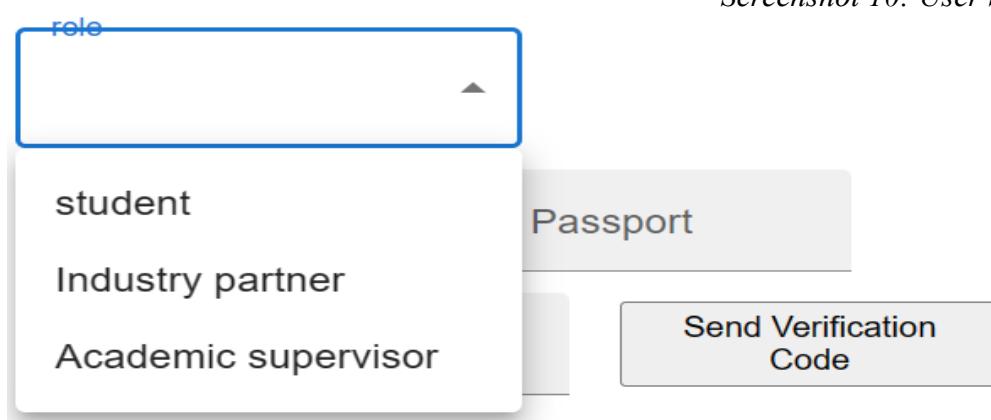
- In our system, the User Authentication functionality plays a crucial role in providing a secure and personalized experience for all stakeholders. These

functionalities also serve as preconditions for applying most of the system's features. This component is collectively delivered through two main functions: User Registration and User Login, accompanied by two auxiliary functions for User Login - Reset Password and Switch Account."

- **User Registration:** In the user registration section, as depicted in screenshot 10, users are required to enter their email, password, and name to create a unique 'digital ID.' Meanwhile, users must choose their desired role (industry partner, student, or academic supervisor) during registration, which shows at screenshot 11. Upon completing registration and logging into the system, the chosen role will grant them specific functionalities and permissions. To enhance security and prevent misuse of personal information by malicious users, passport verification requirements have been implemented in the user registration interface. Furthermore, to validate the accuracy and legitimacy of the entered email, an email verification code function has been introduced, ensuring the provided email is accurate and valid.

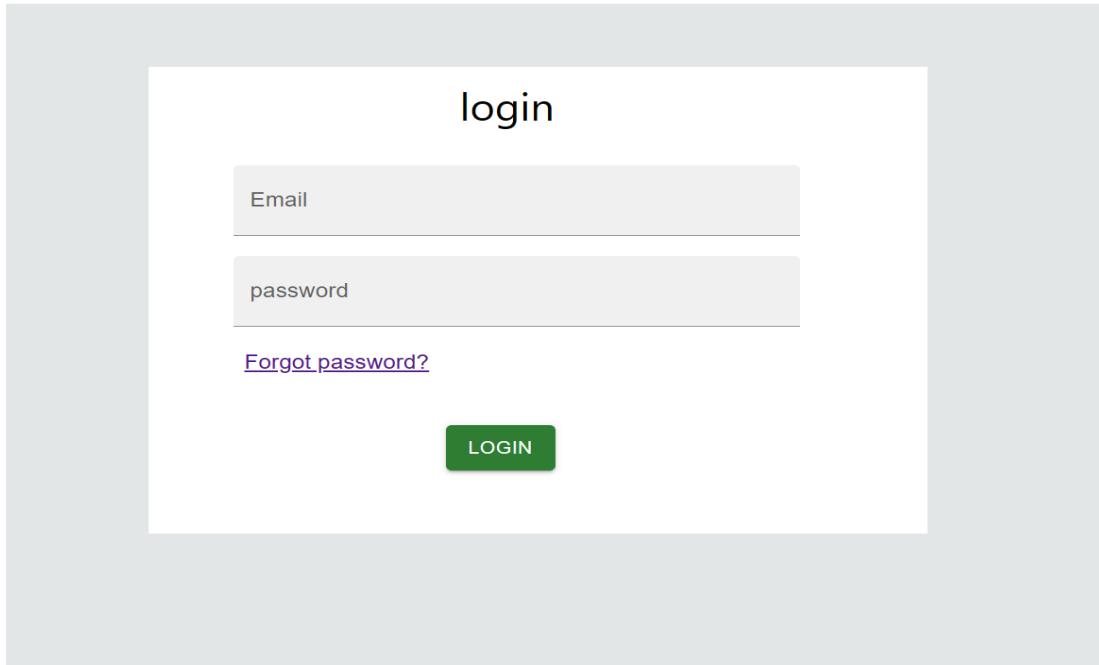
The screenshot shows the 'Sign up' page of the 'Student Industry Project Management System'. At the top right is a 'RETURN HOME' button. The main form has fields for Email, Password, Verify Password, name, and role (a dropdown menu). Below these are two buttons: one for entering a passport and another for sending a verification code. At the bottom is a green 'REGISTER' button.

Screenshot 10: User Sign Up



Screenshot 11: User Role Options in User Sign Up

- **User Login:** The user login function serves as the key for users to access the Ikun system, unlocking the door to various functionalities. Users are required to input a valid email and the associated password set during registration, as illustrated in screenshot 12.



Screenshot 12: User Login

- **Reset Password:** Recognizing that users may forget their registered password, the system offers a 'Forgot password' hyperlink on the user login interface. If a user forgets their password, clicking on the link redirects them to a password reset section. To reset their password, users need to follow 3 steps.
 - i. enter the email address associated with the account (screenshot 13),
 - ii. input the correct password reset verification code sent to their email (screenshot 14) ,
 - iii. establish a new password for accessing their account (screenshot 15), therefore allowing them to maintain access to their accounts.

This screenshot shows the first step of the password reset process. At the top, there is a horizontal navigation bar with three items: "1 Send request code", "2 Verify code", and "3 Reset password". Below this, there is a form with a single input field labeled "Email" and a green "SEND" button below it.

Screenshot 13: Reset Password Step 1

We have already sent the code to your email.

Code

VERIFY

Screenshot 14: Reset Password Step 2

Now you can reset your password.

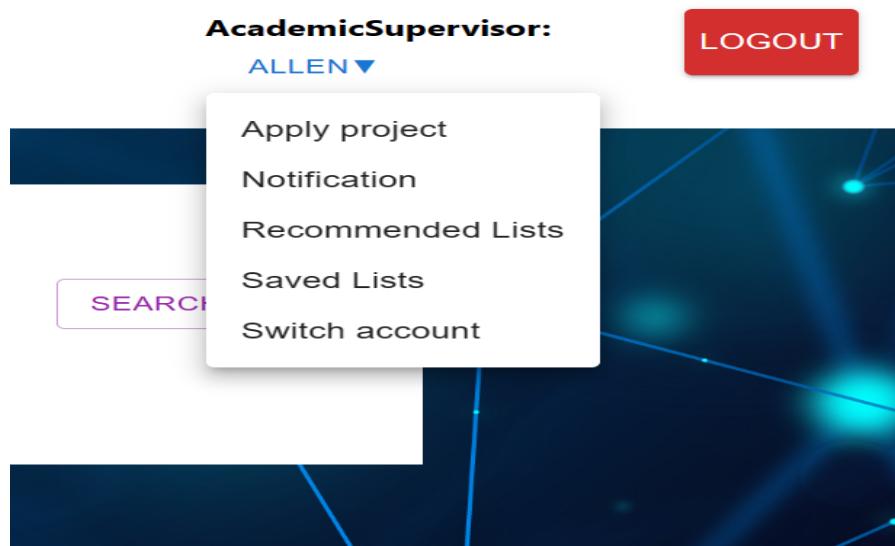
Password

Confirm Password

RESET

Screenshot 15: Reset Password Step 3

- **Switch Account:** As users may hold multiple roles within the system corresponding to different aspects of their lives (e.g., an academic supervisor who also runs their own company), the 'Switch Account' function available in the navigation bars and demonstrated in screenshot 16, which enables them to effortlessly transition between accounts with different roles. This feature eliminates the need for repetitive logouts by directly redirecting users to the login page which is depicted in screenshot 12.



Screenshot 16: Switch Account Function

- **Mapped Project Objective (1 and All)**

- While the user authentication functionality is explicitly outlined as requirement 1, its influence is also pervasive across all functions. The user's authenticated account serves as a reference point, enabling the system to identify the specific user and retrieve previous requests for processing. This ensures the seamless execution of specified operations in the specifications. Additionally, user authentication informs the system about the appropriate user interface (UI) to display and the corresponding permissions to grant, tailored to the user's role. This customization ensures that each user type can effectively meet their specific needs within the system.

Functionality 2: User Profile

- **Description**

- While the User Profile functionality may not directly address specific user demands, it serves a pivotal role as a tool for leveraging information within the system. Beyond traditional introductions among users, this functionality enables the system to comprehend individual users, forming the foundation for delivering personalized and effective information recommendations tailored to each user's unique needs and preferences. These capabilities are facilitated through the Profile Edit and Profile Browse functions.
- Profile Edit:** After the user completes the registration, the system automatically generates a user profile containing elements such as email, name, availability (Workright), project interest type (Project Intention), and skills, where email and name are synchronized automatically when the account is created, as depicted in screenshot 17. Besides, Users have the flexibility to edit Workright, Project Intention, and Skills at any time after clicking the edit button in the Profile interface, as illustrated in screenshot 18. This allows users to showcase their recent situation, promoting better understanding for potential viewers. Additionally, Project Intention and Skills serve as the foundation for the recommendation function. Users can receive different tailored recommendations by adjusting these elements.

The screenshot shows a user profile interface titled "Profile Details". The interface displays the following information in a table:

Email:	2780327997@qq.com
Name:	Jackson
Workright:	Wednesday, Thursday
Project Intention:	Information Technology, Engineering
Skills	asdsdd

At the bottom of the interface is a blue "EDIT" button.

Screenshot 17: User Profile Interface

The screenshot shows a user profile edit form. At the top, there's a header with a back arrow and the word "Sunday". Below it, the "Project Intention" section contains a list of checkboxes. "Information Technology" and "Engineering" are checked, while others like Accounting, Banking, Sport, Business, and Media are unchecked. The "Skills:" section contains a text input field with the value "asdsdd". At the bottom right is a green "SAVE" button.

Screenshot 18: User Profile Edit

- **Profile Browse:** To promote the system's practicality, user profile browsing proves indispensable in multiple human interaction scenarios, including
 - industry partners assessing candidates applying for the program,
 - students searching for suitable groups (reviewing group members needed) and
 - both industry partners and students examining recommended academic supervisors.

In these situations, efficient access to the other party's user profile, as depicted in screenshot 19, facilitates a more effective evaluation of suitability and therefore improves user experiences.

The screenshot shows a user profile browse form. It displays the following information in a table-like structure:

Email:	2780327997@qq.com
Name:	Jackson
Workright:	Wednesday, Thursday
Project Intention:	Information Technology, Engineering
Skills	asdsdd

Screenshot 19: User Profile Browse

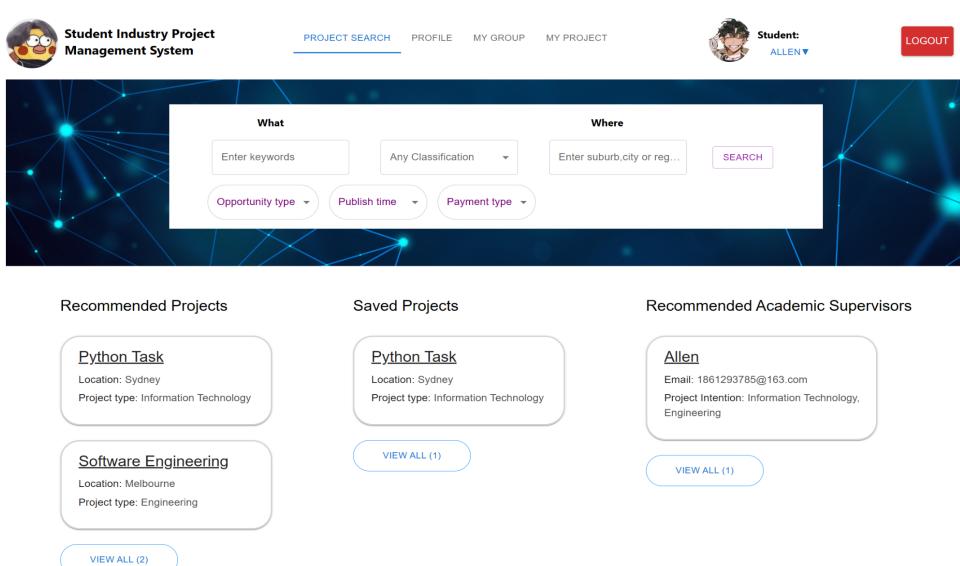
- **Mapped Project Objective (5, 6 and 7)**
 - As previously mentioned in the description, while the User Profile functionality may not have an immediate impact on the user's experience, its underlying concept plays the most critical role in shaping the system's understanding of user needs. This understanding becomes instrumental in

driving the subsequent Recommendation functionality, as the accuracy of recommendations solely relies on the Project Intention and Skills components of the user profile; thus, it maps requirements 5, 6 and 7.

Functionality 3: Recommendation

- **Description**

- The recommendation functionality serves as a pivotal element in optimizing system efficiency and user experience by thoroughly analyzing user profiles. This analysis enables the system to tailor customized recommendations for industry partners and students, ensuring they can fully leverage the information available within the system. By minimizing the user's search burden in a comprehensive context, the recommendation feature enhances the overall usability and effectiveness of the system at the same time, offering users more relevant suggestions on partnerships or project opportunities. Currently, the Ikun system employs this concept to provide tailored recommendations on projects and academic advisors.
- **Project Recommendation (for students and academic supervisors only):** Recognizing the abundance of opportunities within the Ikun system, potentially requiring considerable time for students and academic supervisors to find suitable projects, we aim to utilize a project recommendation system to streamline the process. The dashboard interface now features a dedicated "Recommended Projects" section, as indicated in screenshot 20 and 21, directly presenting projects tailored to individual profiles and project intentions. This functionality is complemented by the project intention option in the profile, aligning with the system's recommendation mechanism. Users can explore more recommended projects (On the initial page, the system only shows 3) that meet the criteria through the "view all" button within the Recommended Projects area or by accessing Recommended Projects section (screenshot 22) after clicking the "Recommended List" option in the navigation bars (screenshot 23).
-



Screenshot 20: Recommendations for Students

What

Where

Opportunity type

Publish time

Payment type

SEARCH

Recommended Projects

Saved Projects

[Python Task](#)
Location: Sydney
Project type: Information Technology

[Software Engineering](#)
Location: Melbourne
Project type: Engineering

[VIEW ALL \(1\)](#)

[VIEW ALL \(2\)](#)

Screenshot 21: Recommendations for Academic Supervisors

RECOMMENDED PROJECTS

RECOMMENDED ACADEMIC SUPERVISORS

The total numbers of recommend projects: 2

[Python Task](#)
Location: Sydney
Project type: Information Technology
GroupProject | Paid
[UNSAVE](#)

[Software Engineering](#)
Location: Melbourne
Project type: Engineering
IndividualProject | Paid
[SAVE](#)

Screenshot 22: All Recommended Projects

AcademicSupervisor:
ALLEN▼

LOGOUT

SEARCH

- Apply project
- Notification
- Recommended Lists
- Saved Lists
- Switch account

Screenshot 23: Recommended Lists Option

- **Academic Supervisor Recommendation (for industry partners and students only):** With a focus on reducing the search burden for industry partners and students seeking suitable academic guidance, we have implemented a parallel recommendation system design align to Project Recommendation. This assists users in swiftly identifying the partnerships that best align with their project needs. The design, illustrated in screenshots 20 and 25, mirrors the approach taken for project recommendations. The only distinction lies in accessing all recommended academic supervisors, which can be done through the "Recommended List" option in the navigation bars by clicking on the "Recommended Academic Supervisor" option, like screenshot 26 shows.

Student Industry Project Management System

PROJECT SEARCH PROFILE CREATE PROJECT MY PROJECT

IndustryPartner: JACKSON ▾ LOGOUT

What

Enter keywords Any Classification Enter suburb, city or reg... SEARCH

Opportunity type Publish time Payment type

Where

Recommended Projects

Python Task
Location: Sydney
Project type: Information Technology

Software Engineering
Location: Melbourne
Project type: Engineering

Saved Projects

Python Task
Location: Sydney
Project type: Information Technology

Recommended Academic Supervisors

Allen
Email: 1861293785@163.com
Project Intention: Information Technology, Engineering

VIEW ALL (2) **VIEW ALL (1)**

Screenshot 24: Recommendations for Industry Partner

Student Industry Project Management System

PROJECT SEARCH PROFILE CREATE PROJECT MY PROJECT

IndustryPartner: JACKSON ▾ LOGOUT

Recommend Lists

RECOMMENDED PROJECTS RECOMMENDED ACADEMIC SUPERVISORS

The total numbers of recommend academic supervisors: 1

Allen
Email: 1861293785@163.com
WorkRight: Friday, Tuesday
Project Intention: Information Technology, Engineering

UNSAVE

Screenshot 25: All Recommended Academic Supervisors

- **Mapped Project Objective (5, 6 and 7)**

- Essentially, the recommendation functionality serves as a strategic facilitator, streamlining the process of identifying ideal partnerships and project engagements for students, industry partners, and academic supervisors. This is achieved through tailored project and academic supervisor recommendations for specific user groups, effectively meeting the requirements outlined in 5, 6, and 7.

Functionality 4: Student Group

- **Description**

- As detailed in the specifications, industry partners possess the capability to publish group projects and enlist student groups for project delivery. In response, we have integrated the student group functionality at "My Group" interface to facilitate students' seamless application for their preferred projects. This functionality encompasses three critical groups of functions:
 - Group Information Browse,
 - Group Creation and Manage (from the group creator's perspective),
 - Group Join or Leave.
- **Group Information Browse:** To optimize students' utilization of group functions, we've devised a comprehensive service hub for groups. The "My Group" interface, shown in screenshot 25, empowers students to conveniently look through the groups they've joined (Groups you are in) and explore additional groups available for participation (Other joinable groups). Moreover, a simple click on any group (underlined) unveils its intricacies in screenshot 26, showcasing essential details such as the group creator (distinguished with a star next to their name), group name, and description. Simultaneously, students can peruse the profiles of group members by clicking on their names. This detailed group content presentation serves to enhance students' ability to identify and engage with suitable management groups.

The screenshot displays the 'My Group' interface for a student. At the top, a dark blue header bar contains the text 'My Group'. Below this, the main content area is divided into two sections: 'Groups you are in' and 'Other joinable groups'.

Groups you are in:

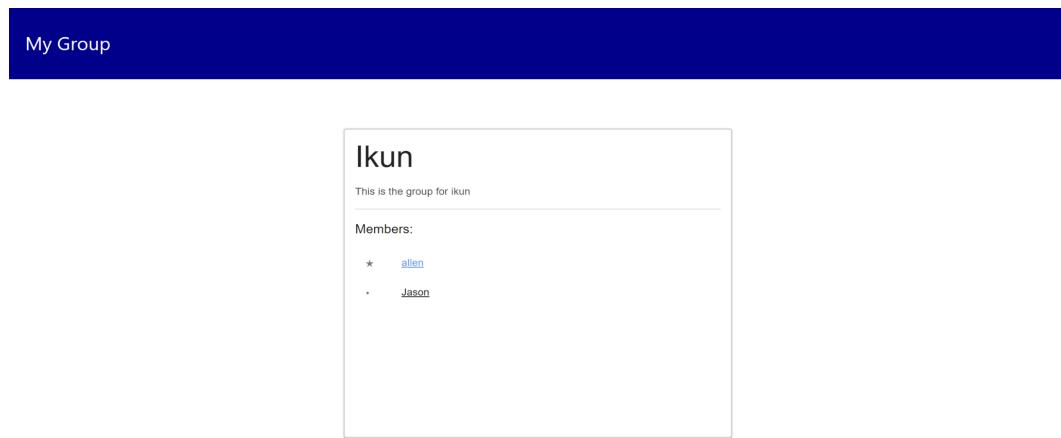
Group name	Description	Members	Actions
abcd	This is the group for abcd	1/3	EDIT DELETE

Other joinable groups:

Group name	Description	Members	Actions
ikun	This is the group for ikun	1/5	JOIN

A green 'CREATE' button is located in the top right corner of the main content area.

Screenshot 25: Student My Group Interface



Screenshot 26: Group Information

- **Group creation and Manage:** To initiate the use of group functionalities, students must first utilize the group creation function. As depicted in screenshot 27, the Ikun system prompts students to input a group name and description, facilitating clear communication of the group's purpose to attract like-minded individuals. Additionally, students are prompted to specify the group size and visibility, allowing the creator to manage the group effectively based on its intended scope. Recognizing the dynamic nature of projects and group dynamics, the group creator has the authority to update the group name and description, adapting to current situations as illustrated in screenshots 28 and 29. Moreover, as we can see from screenshot 28 that users have the option to delete a group if it no longer serves a purpose, providing a streamlined interface for the group page. This comprehensive suite of functionalities enhances the flexibility and management of student groups within the Ikun system.

A screenshot of a "Create Group" form. It includes fields for "Group Name", "Group Description", and "Limit members". Below these, there is a "Group status:" section with radio buttons for "public" and "Private", and a "PUBLISH" button at the bottom.

Screenshot 27: Group Creation

Groups you are in

Group name	Description	Members		
<u>Ikun</u>	This is the group for ikun	2/5	EDIT	LEAVE
<u>abcd</u>	This is the group for abcd	1/3	EDIT	DELETE

Screenshot 28 & 29: Group Edit

Other joinable groups

Group name	Description	Members	
<u>Ikun</u>	This is the group for ikun	1/5	JOIN

Screenshot 30: Group Join

Groups you are in				CREATE
Group name	Description	Members		
<u>Ikun</u>	This is the group for ikun	2/5	EDIT	LEAVE
<u>abcd</u>	This is the group for abcd	1/3	EDIT	DELETE

Screenshot 31: Group Leave

- **Mapped Project Objective (4)**

- In a nutshell, the incorporation of the student group functionality in the Ikun system harmoniously adheres to requirement 4, catering to the requirements of industry partners and streamlining students' involvement in group projects. This enhancement fosters a more collaborative and adaptable environment for student groups within the Ikun system, ultimately contributing to improved project delivery.

Functionality 5: Opportunity Hunt

- **Description**

- Utilizing the powerful Opportunities Hunt functionality, the Ikun system offers users an efficient and all-encompassing method for swiftly identifying and participating in relevant opportunities. These carefully crafted functionalities are adeptly designed to meet the varied needs of users, seamlessly integrating components: Recommendation, Keyword Search, Search Filter and Result Rank, and Save Project. This strategic approach ensures a dynamic and

time-effective exploration of opportunities, providing users with a tailored and streamlined experience within the Ikun system.

- **Recommendation:** As outlined in the User Profile functionality, the Ikun system relies on the Project Intention section within user profiles to deliver personalized services. This section serves as the basis for customized project recommendations to all logged-in users, as shown as screenshot 32, enhancing the efficiency of project discovery for students and academic supervisors. Additionally, industry partners can utilize the recommended projects as references to help them better improve their project specification. Furthermore, as depicted in screenshot 33, a dedicated feature is available for students and industry partners, offering recommendations on academic supervisors with similar project intentions. It aims to streamline the process of finding academic supervisors aligned with shared goals, saving users valuable time and enhancing their overall experience on the platform.

Recommended Projects

The screenshot shows a list of recommended projects. Each project is presented in a rounded rectangular box. The first project is titled "Python Task" and includes details: Location: Sydney, Project type: Information Technology. The second project is titled "Software Engineering" and includes details: Location: Melbourne, Project type: Engineering. Below the boxes is a blue button labeled "VIEW ALL (2)".

Python Task
Location: Sydney
Project type: Information Technology

Software Engineering
Location: Melbourne
Project type: Engineering

[VIEW ALL \(2\)](#)

Screenshot 32: Project Recommendation

Recommended Academic Supervisors

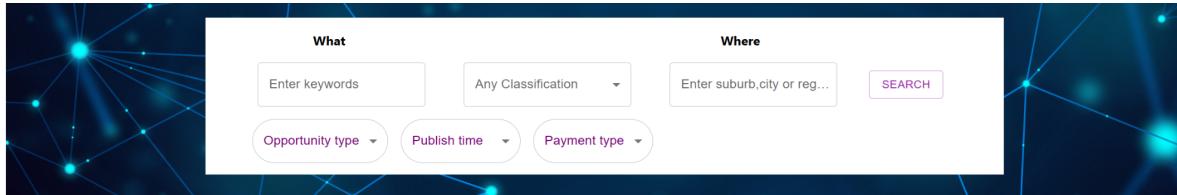
The screenshot shows a list of recommended academic supervisors. Each supervisor is presented in a rounded rectangular box. The first supervisor is named "Allen" and includes details: Email: 1861293785@163.com, Project Intention: Information Technology, Engineering. Below the box is a blue button labeled "VIEW ALL (1)".

Allen
Email: 1861293785@163.com
Project Intention: Information Technology,
Engineering

[VIEW ALL \(1\)](#)

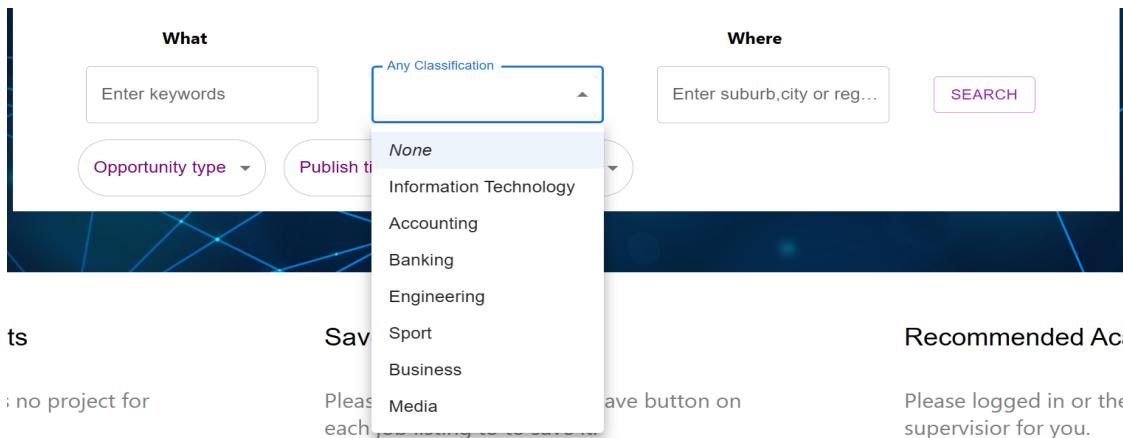
Screenshot 33: Academic Recommendation

- **Keyword Search:** In addition to passively obtaining project information, users can also take an active approach to finding related projects through keyword searches on “What” and “Where” sections on the dashboard, as depicted in screenshot 34. Users can effectively refine their search, identifying opportunities that match specific criteria. This active search approach lets users quickly assess whether the project aligns with their job content and workplace preferences, providing a more targeted and efficient project discovery experience.

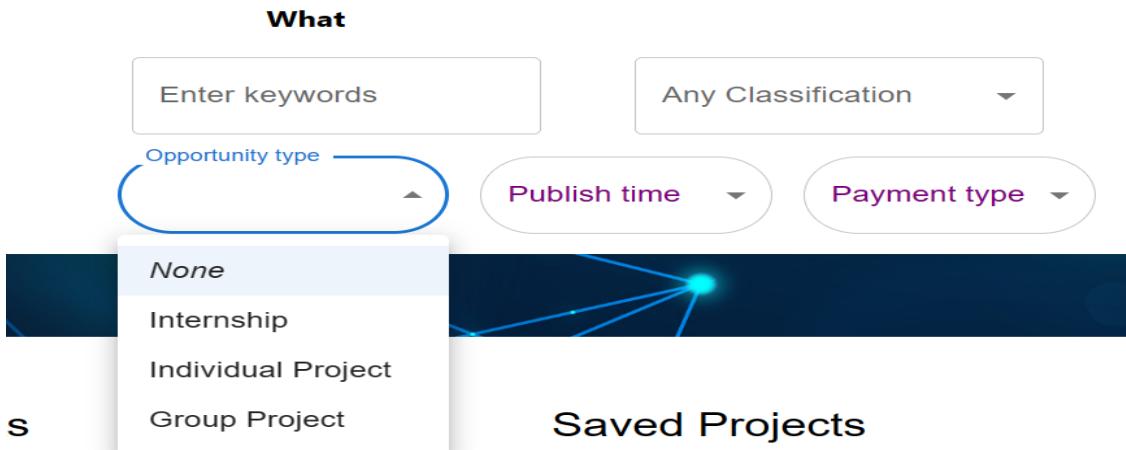


Screenshot 34: Search Bar

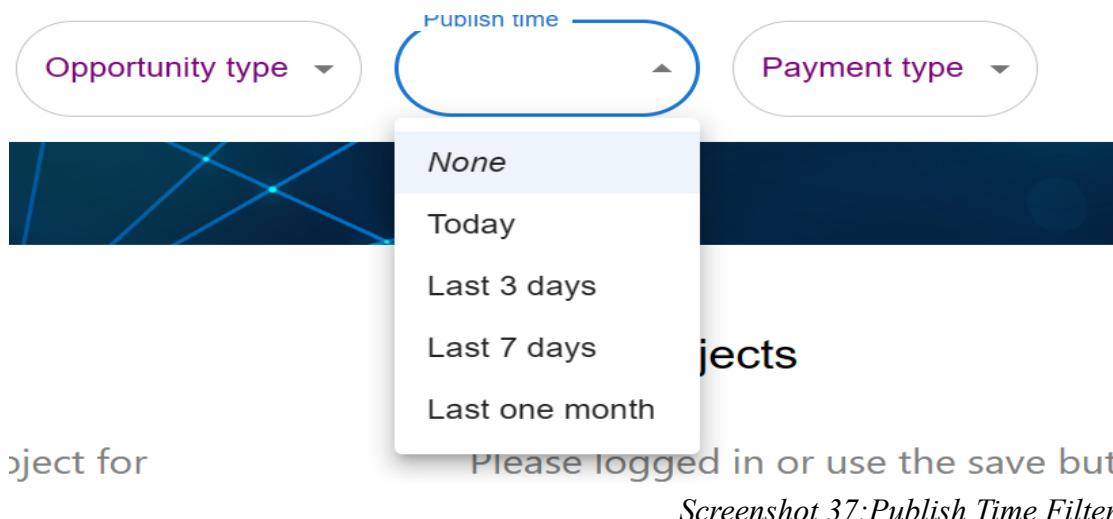
- **Search Filter and Result Rank:** At the same time, the Ikun system integrates an advanced search filter mechanism to enhance the active search experience. This feature allows users to search their opportunity results by specifying criteria: industry classification, opportunity type, project publication time, and project payment type, shown across screenshots 35 to 38. By providing these detailed filters, the system empowers users to further narrow down their search scope, ensuring a more precise match with their individual preferences and requirements.



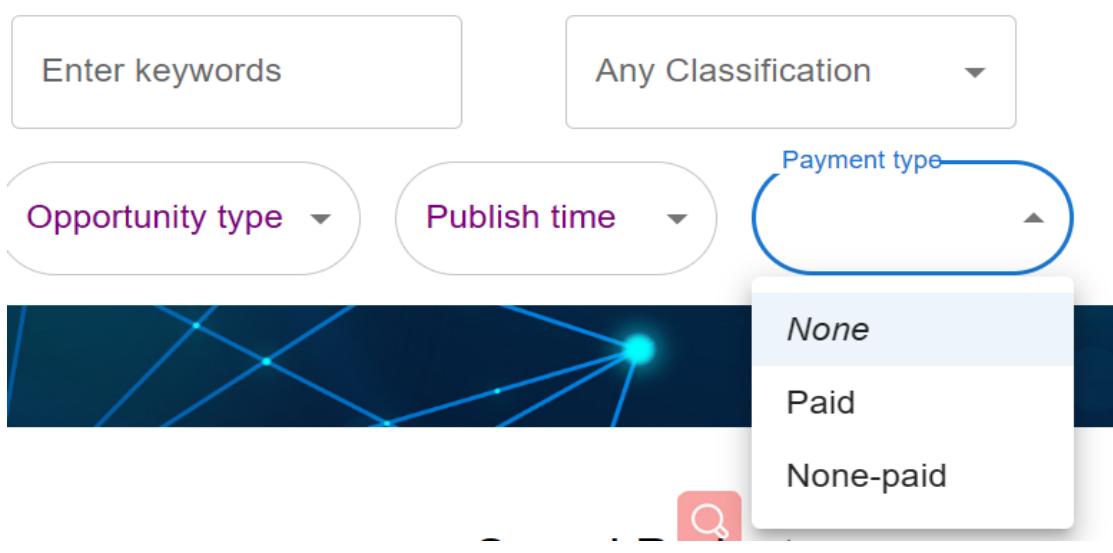
Screenshot 35: Industry Classification Filter



Screenshot 36: Opportunity Type Filter



Screenshot 37: Publish Time Filter



Screenshot 38: Payment Type Filter

- **Save/Unsave Project:** Given the seriousness of working on a project, students and academic advisors often exercise great caution when finalizing the projects they apply for. The Ikun system offers a valuable feature enabling them to save the projects they find intriguing. This functionality not only allows for easy access to these saved projects in the future but also enhances overall convenience for them to make application decisions better. Similarly, industry partners can utilize this feature to bookmark projects of interest, serving as a reference for future considerations. Additionally, users have the flexibility to unsave projects that no longer capture their interest, contributing to effective management.
- **Mapped Project Objective (3, 5, 6, 7, 8)**
 - Within the Recommendation section, students and academic supervisors effortlessly access projects matching their project intentions, establishing connections with industry partners and fulfilling requirements 5, 6, 7, and 8. Simultaneously, users can utilize the active search feature to explore all currently uploaded projects, meeting the criteria of requirement 3 (browsing) and requirement 8 concurrently. The integration of these components and others contributes to a smoother browsing and search experience, ultimately generating a user-friendly environment.

Functionality 6: Project

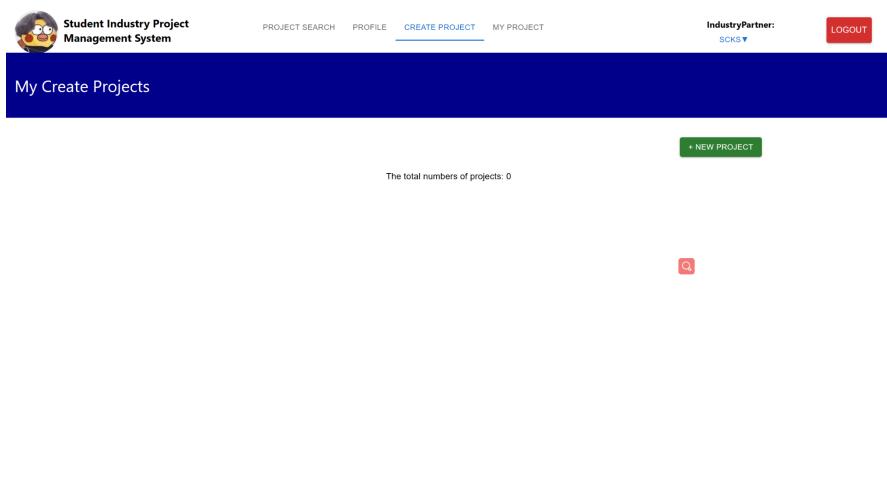
- **Description**
 - The project functionality serves as the cornerstone of the Ikun system, forming the foundation for a project management system. Its primary duty lies in facilitating the successful delivery of all projects within the system while fostering a collaborative environment on the platform. This integral component ensures seamless project management, providing users with features: Project Creation, Edit and Delete, Save/Unsave User, Application Respond and Feedback necessary for efficient participation, successful completion of diverse projects.
 - **Project Creation, Edit and Delete:** The project creation feature serves as a distinctive authorization that sets industry partners apart from other user types. Industry partners have the exclusive ability to upload projects onto the Ikun system, recruiting students and academic supervisors to collaboratively develop and implement project proposals.

Industry partners can click the "New project" button in screenshot 39 to publish their own projects in the CREATE PROJECT page in the navigation. To utilize this unique capability, industry partners are required to input essential project information, as illustrated in screenshot 40, which includes Opportunity Title, Project Classification, Location, Opportunity Type, Problem Statement, Availability Requirements, and Payment Type. This comprehensive information ensures that students and academic supervisors have a thorough understanding of the project. Additionally, industry partners have the option to provide additional details such as Project Desired Outcomes, Required Skills, Potential Deliverables, and Expected Delivery

Cycle, as depicted in screenshot 41. Once published, the project becomes visible to all users in search page.

Once the project is created, the original screenshot 39 will show the created projects (each industry partner can only see their own published projects on this page). The updated page is displayed in screenshot 42. Industry partners can edit their created projects by clicking the "edit" button and go back to the "Opportunity Post" page (screenshot 40&41) to make changes.

Furthermore, Industry partners can also click the "Delete" button to delete an already created project (screenshot 42).



Screenshot 39:Created Projects

A screenshot of the 'Opportunity POST' page. It features input fields for 'Title' (a grey placeholder box), 'Project Classification' (radio buttons for Information Technology, Accounting, Banking, Engineering, Sport, Business, and Media), 'Location' (a grey placeholder box), and 'Opportunity type' (radio buttons for Internship, Individual Project, and Group Project).

Screenshot 40:Project Creation

Problem statement

Availability requirement

Payment type

Paid

Non-paid

Desired outcomes(optional)

Required skills(optional)

Potential deliverables(optional)

Expected delivery cycle(optional)

PUBLISH

Screenshot 41:Project Creation

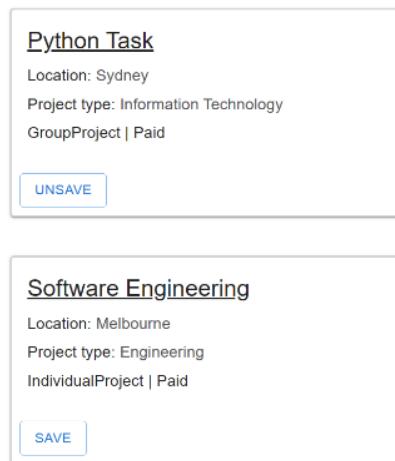
The screenshot shows the 'My Create Projects' section of the system. At the top, there's a navigation bar with links for PROJECT SEARCH, PROFILE, CREATE PROJECT (which is underlined), and MY PROJECT. On the right, it shows 'IndustryPartner: JACKSON▼' and a LOGOUT button. Below the navigation, a message says 'The total numbers of projects: 2'. Two project cards are listed:

- Python Task**
Location: Sydney
Project type: Information Technology
GroupProject | Paid
EDIT **DELETE**
- Software Engineering**
Location: Melbourne
Project type: Engineering
IndividualProject | Paid
EDIT **DELETE**

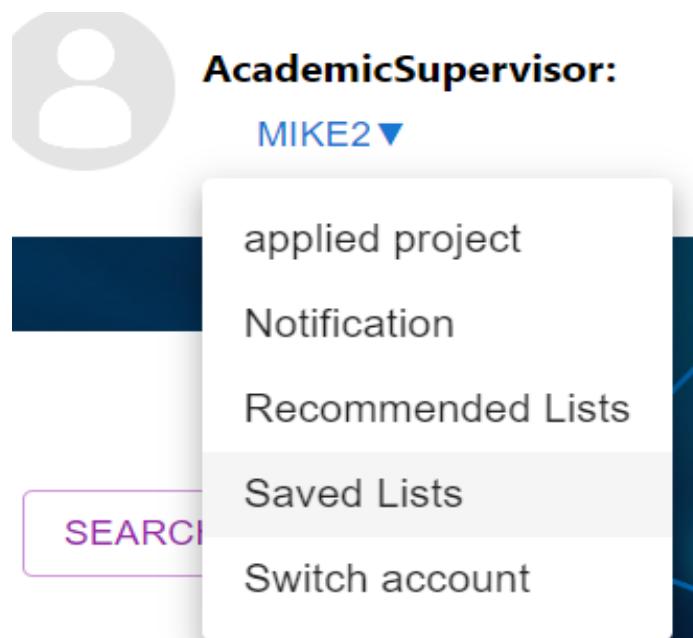
Screenshot 42:Created Projects

- **Project save and unsave:**

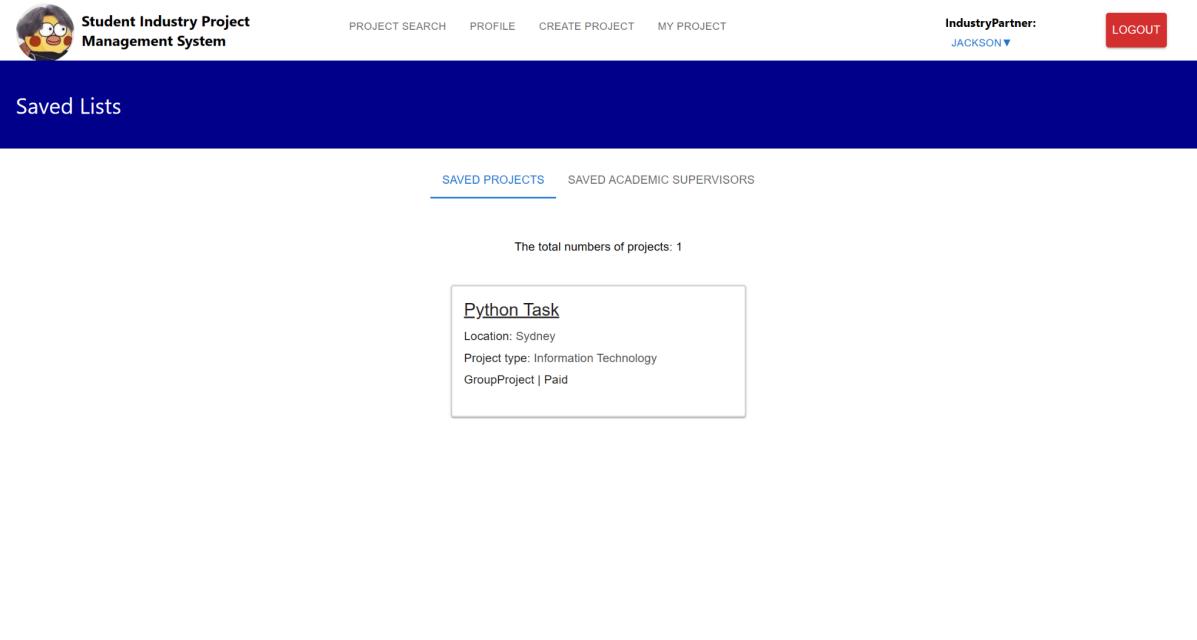
Either the item card in the recommendation system, the project card in the search, or the project can be save or Unsave in the Project Details. When you click save, the saved items will appear in save lists and the button state will change to Unsave. When you click Unsave, the item is removed from the save list and the button status switches back to Save. Academic supervision can be saved or unsaved in the same way.



Screenshot 43:save project

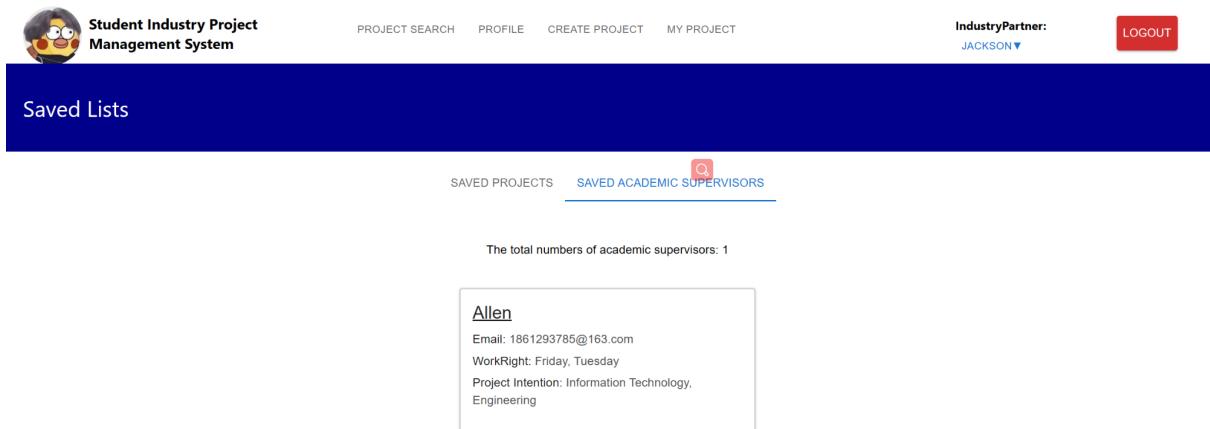


Screenshot 44:saved lists



The screenshot shows a user's saved project list. At the top, there is a navigation bar with links for 'PROJECT SEARCH', 'PROFILE', 'CREATE PROJECT', 'MY PROJECT', 'IndustryPartner: JACKSON', and a 'LOGOUT' button. Below the navigation bar, the title 'Saved Lists' is displayed. Underneath, there are two tabs: 'SAVED PROJECTS' (which is currently selected) and 'SAVED ACADEMIC SUPERVISORS'. A message indicates 'The total numbers of projects: 1'. A box contains the details for a project named 'Python Task', located in Sydney, categorized as Information Technology, and marked as a GroupProject | Paid.

Screenshot 45:saved Lists



The screenshot shows a user's saved academic supervisor list. The interface is similar to Screenshot 45, with a 'SAVED PROJECTS' tab selected. A message says 'The total numbers of academic supervisors: 1'. A box displays the details for an academic supervisor named 'Allen', including their email (1861293785@163.com), work schedule (WorkRight: Friday, Tuesday), and project intention (Information Technology, Engineering).

Screenshot 46:saved Lists

User can view the projects and academic supervisors they saved in the saved Lists

- **Application Respond:** Student and academic supervisor can apply the project. All the apply buttons are in project details.
Students can only apply to the project that already has a supervisor. Therefore, for every project, the academic supervisor should first apply to supervise it.

This is the page of academic supervisor application.

Python engineer

Location: Sydney

Project type: Information Technology

GroupProject | Paid

Posted 1 day ago

SUPERVISE

UNSAVE

After clicking the supervise button, we can see the application form.

Application

Apply to this project Python engineer

Name

Email

Choose your Uni

University

Resumes

SUBMIT

After submitting it, the user goes back to the dashboard page and the user can see the project applied in the ‘applied project’ page and the user can also delete the application.

My apply projects

The screenshot shows a dark blue header with the text 'My apply projects'. Below it is a white card with a thin gray border. At the top of the card is the title 'Python engineer'. Underneath the title, there is a list of details: 'Location: Sydney', 'Project type: Information Technology', 'GroupProject | Paid', 'Univeristy: USYD', and 'resume: python'. At the bottom of the card is a red rectangular button labeled 'DELETE APPLY'.

Then, in the notification page of the corresponding industry partner, there will be a notification.

Notification

The screenshot shows a dark blue header with the word 'Notification'. Below it is a blue bar containing the text '≡ Your project "Python engineer" has been applied by the supervisor [mike2](#). [applicant's resume](#)'. To the right of this text are two buttons: a green 'ACCEPT' button and a red 'DECLINE' button.

After clicking the 'accept' button or 'decline' button, the corresponding academic supervisor will receive a message.

The screenshot shows the main interface of the 'Student Industry Project Management System'. At the top, there is a logo of a cartoon character, the text 'Student Industry Project Management System', and navigation links for 'PROJECT SEARCH', 'PROFILE', and 'MY PROJECT'. On the right side, there is a user profile icon, the text 'AcademicSupervisor: MIKE2', and a 'LOGOUT' button. Below this, a dark blue header says 'Notification'. Underneath are two blue bars with notifications: the first bar says '≡ The project "Python engineer" you applied has been accepted.' and the second bar says '≡ The project "r3wr3wr3wrw3r" you applied has been declined.'

The screenshot below of student application. The Apply button can only be activated after the teacher has successfully applied for this project.

Python engineer

Location: Sydney

Project type: Business

GroupProject | Paid

Posted 1 day ago

APPLY

SAVE

When the application button is activated, students can click apply to jump to the screenshot below. This page is different from academic supervision in that if it is a group project, it will show the optional group. If the requested project is an individual project, the selectable group will not be displayed.

Apply to this project Python engineer

Name

Email

Choose your Uni and Group

University

Group

Ikun

Resumes

SUBMIT

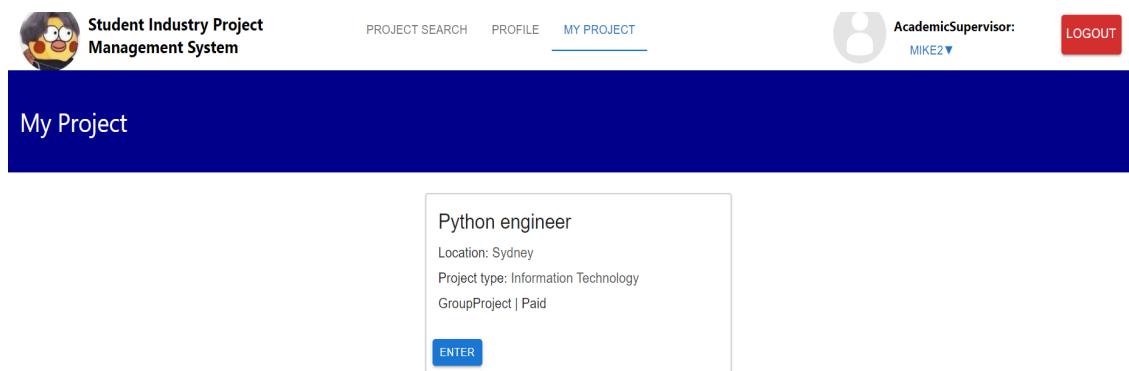
This is the notification of the page of industry partners. Industry partners can see student requests and click on the student name or group name to see the corresponding profile information and group information.



This is the notification of the page of students , which is same with academic supervisor.



After the supervision of the enterprise, the Project will appear in the "My Project" on the navigation bar after all the teachers and students are paired. Click Enter to see the specific information delivery method and feedback of the project.



The screenshot shows the "Student Industry Project Management System" interface. At the top, there is a navigation bar with icons for "PROJECT SEARCH", "PROFILE", "MY PROJECT" (which is underlined), "AcademicSupervisor: MIKE2▼", and "LOGOUT". Below this, a large blue header says "My Project". In the main content area, there is a card for a project titled "Python engineer". The card includes the location "Sydney", project type "Information Technology", and status "GroupProject | Paid". At the bottom of the card is a blue "ENTER" button.

Click on the underlined project name "Python engineer" in the project information to see the project information. Click on the underlined group name, industry partner name, and academic supervisor name in Project Members information to see the information of all team members, academic supervisor and industry partner profiles.

[Project Information:](#)

[Python engineer](#)

Location: Sydney

Project type: Information Technology

GroupProject | Paid

[Project Members Information:](#)

- Student Group Name: [werew](#)
- Industry Partner: [mike2\(8526915010@qq.com\)](#)
- Academic Supervisor: [mike2\(85269150210@qq.com\)](#)

[Project Delivery:](#)

- The industry partner will send the initial project documents to the student's (or group leader) email address.
- Students can send a demo of the project to the email address of their academic supervisor and industry partners.
- Students can send the final project results to the email address of their academic supervisor and industry partners.
- If there is an unequal contribution between students please email the academic supervisor of the project.

[Project Feedback:](#)

- [Academic Supervisor Feedback](#)
- [Academic Supervisor Assessment](#)
- [Industry Partner Assessment](#)

- **Feedback:** During the student's delivery, the Ikun system provides users with feedback and assessment pages. The academic supervision can provide demo feedback and final feedback to provide students with helpful suggestions during and after the project(screenshot 47) . The project supervisor also evaluates projects completed by students through 3 points (Evaluating deliverables, Problem-solving skills and Overall contribution) in screenshot(48). In addition, Industry Partners can rate their experience working with Students and Academic Supervisors in screenshot 50. After the academic oversight submission, students and industry partners can view the content but not change it. After industry partners have submitted evaluations, student academic oversight will only see what industry partners have written to each other to ensure the rigor of the system.

Supervisor Feedback

Demo Feedback:

Final Feedback:

SUBMIT

Screenshot 47: Supervisor Feedback

Supervisor Assessment

Evaluating deliverables:

good good good good

Problem-solving skills:

good good good good

Overall contributions:

good good good good

SUBMIT

Screenshot 48: Supervisor Assessment

Industry Assessment

Experience with students:

good good good good good

Experience with academic supervisor:

good good good good good

SUBMIT

Screenshot 49: Industry Assessment

Supervisor Feedback

Demo Feedback:

good good good good

Final Feedback:

good good good good

Screenshot 50 : Supervisor Feedback

- **Mapped Project Objective (4, 5, 7, 8, 9, 10, 11, 12)**

In the Specification Requirement Mapping Table. Applications correspond to 4,5,9. Preservation projects correspond to academic supervision 8. Feedback corresponds to 10, 11, 12.

Third-party functionalities

1. **Flask:** A lightweight Python web framework that provides basic functionalities such as routing, request handling, template engine, etc., for creating web applications.
2. **Flask-JWT-Extended:** Provides support for JSON Web Tokens (JWT) to implement user authentication, token generation, and access control.
3. **Flask-Mail:** A Flask extension for handling emails, simplifying the communication process with application users via email. Commonly used for sending registration confirmations, password reset notifications, etc.
4. **Flask-SQLAlchemy:** A Flask extension for interacting with databases, integrated with SQLAlchemy. It offers Object-Relational Mapping (ORM) functionality to execute SQL operations and simplify data access.
5. **Flask-Marshmallow:** A library for object serialization and deserialization in Flask. It facilitates the conversion of objects to JSON data (serialization) and the restoration of objects from JSON data (deserialization). This is used for handling API requests and responses, streamlining the data interaction process.
6. **React-router-dom:** React Router is a popular library for handling routing and navigation in React applications. It allows you to create single-page applications (SPAs) with multiple views, enabling you to navigate between different components or pages without causing a full page refresh. It provides a set of components and a routing configuration that makes it easy to manage the URL and render different components based on the URL. It is commonly used for building web applications with complex user interfaces and multiple views.
7. **Material UI:** Material-UI is a popular open-source React UI framework that implements Google's Material Design principles. Material Design is a design system developed by Google that provides guidelines and components for creating visually appealing and consistent user interfaces across different platforms and devices. We used it to make frontend setup easier.

Data Structure

Model Descriptions and Database Schemas

The database schema is defined using SQLAlchemy and consists of eight main models: User,UserCode, Project, Group,UserSaved,ApplyProject and Feedback.

User

This Schema contains the user's account information and profile information.

```
class User(db.Model):
    user_id = db.Column(db.Integer, primary_key=True)
    role = db.Column(db.INTEGER)
    email = db.Column(db.Text())
    passport = db.Column(db.Text())
    password = db.Column(db.Text())
    name = db.Column(db.Text())
    work_rights = db.Column(db.ARRAY(db.INTEGER))
    project_intention = db.Column(db.ARRAY(db.INTEGER))
    skill = db.Column(db.Text())
    avatarUrl = db.Column(db.Text())
    groups = db.relationship('Group', secondary='user_group', back_populates='members')
    created_groups = db.relationship('Group', back_populates='creator')
    saved_projects = db.relationship('Project', secondary=user_saved_project)
```

UserCode

This Schema stores the user's verification code information. The codes sent when registering an account or forgetting the password will be stored here.

```
class UserCode(db.Model):
    user_code_id = db.Column(db.Integer, primary_key=True)
    vcode = db.Column(db.Text())
    email = db.Column(db.Text())
    create_at = db.Column(db.TIMESTAMP)
```

Project

The core of this system represents the projects created by Industry Partners.

```

class Project(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.Text())
    publish_date = db.Column(db.TIMESTAMP)
    location = db.Column(db.Text())
    job_classification = db.Column(db.Integer)
    problem_statement = db.Column(db.Text())
    requirement = db.Column(db.Text())
    payment_type = db.Column(db.Integer)
    opportunity_type = db.Column(db.Integer)
    project_status = db.Column(db.Integer)

    desired_outcomes = db.Column(db.Text())
    required_skill = db.Column(db.Text())
    potential_deliverable = db.Column(db.Text())
    expected_delivery_cycle = db.Column(db.Text())
    user_id = db.Column(db.ForeignKey(User.user_id))
    user = db.relationship(
        "User",
        primaryjoin="User.user_id == Project.user_id",
        lazy=True,
    )

```

Group

Groups created on behalf of student users.

```

✓ class Group(db.Model):
    group_id = db.Column(db.Integer, primary_key=True)
    group_name = db.Column(db.Text())
    group_description = db.Column(db.Text())
    limit_no = db.Column(db.Integer)
    # 0:public 1:private
    is_private = db.Column(db.Integer)
    creator_id = db.Column(db.ForeignKey(User.user_id))
    creator = db.relationship('User', back_populates='created_groups', foreign_keys=[creator_id])
    members = db.relationship('User', secondary='user_group', back_populates='groups')

✓ def __init__(self, name, description, limit_no, is_private=False):
    self.group_name = name
    self.group_description = description
    self.limit_no = limit_no
    self.is_private = is_private

```

UserSaved

Projects saved on behalf of student users and Academic Supervisor users

```

    < class UserSaved(db.Model):
        id = db.Column(db.Integer, primary_key=True)
        user_id = db.Column(db.ForeignKey(User.user_id))
        user = db.relationship('User', foreign_keys=[user_id])
        saved_user_id = db.Column(db.ForeignKey(User.user_id))
        saved_user = db.relationship('User', foreign_keys=[saved_user_id])

```

ApplyProject

Projects applied on behalf of student users and Academic Supervisor users

```

class ApplyProject(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    teacher_uni = db.Column(db.Integer)
    teacher_resumes = db.Column(db.Text())
    student_uni = db.Column(db.Integer)
    student_resumes = db.Column(db.Text())
    teacher_id = db.Column(db.ForeignKey(User.user_id))
    teacher = db.relationship('User', foreign_keys=[teacher_id])
    student_id = db.Column(db.ForeignKey(User.user_id))
    student = db.relationship('User', foreign_keys=[student_id])
    group_id = db.Column(db.ForeignKey(Group.group_id))
    group = db.relationship('Group', foreign_keys=[group_id])
    project_id = db.Column(db.ForeignKey(Project.id))
    project = db.relationship('Project', foreign_keys=[project_id])
    apply_status = db.Column(db.Integer)

```

Feedback

Feedback on behalf of users

```

class Feedback(db.Model):
    feedback_id = db.Column(db.Integer, primary_key=True)
    project_id = db.Column(db.ForeignKey(Project.id))
    project = db.relationship('Project', foreign_keys=[project_id])
    demo_feedback = db.Column(db.Text())
    final_feedback = db.Column(db.Text())
    evaluating_deliverables = db.Column(db.Text())
    problems = db.Column(db.Text())
    contributions = db.Column(db.Text())
    student_experience = db.Column(db.Text())
    supervisor_experience = db.Column(db.Text())

```

Routes

/user/<user_id>

- GET: Get the profile information of the user with this ID

/profile

- GET: Get the profile of the currently logged in user
- PUT: Edit user's profile

/sendCode

- POST: Send the code to email

/register

- POST: Register a user

/login

- POST: Sign in

Password

- /resetPassword/sendCode
 - ❖ POST: Send verification code to reset password
- /verifyCode
 - ❖ POST: Verify code to reset password
- /resetPassword
 - ❖ PUT: Reset user's password

Project

- /project
 - ❖ GET: Get and search all projects based on conditions
 - ❖ POST: Create a project
 - ❖ PUT: Edit a project
- /project/<id>
 - ❖ GET: Get a project detail
 - ❖ DELETE: Delete a project
- /project-created
 - ❖ GET: Get all projects by user created

Group

- group
 - ❖ POST: Create a new group
 - ❖ PUT: Edit a group
- /group/join/<group_id>
 - ❖ GET: Join a group
- /group/leave/<group_id>
 - ❖ GET: Leave a group
 - ❖ /group/<group_id>
 - ❖ GET: Get a group detail
 - ❖ DELETE: Delete a group
- /notInGroup
 - ❖ Get: Get user not join and public group
- /joinedGroup
 - ❖ GET: Get user already joined group

Save

- /savedProject

- ❖ GET: Get user all saved project
- /saved/project/<project_id>
 - ❖ GET: Add a project to save list
- /unsaved/project/<project_id>
 - ❖ GET: Remove a project to save list
- /savedUser
 - ❖ GET: Get all saved users
- /savedUser/<user_id>
 - ❖ GET: Save a user
- /unSavedUser/<user_id>
 - ❖ DELETE: Unsave a user

Recommend

- /recommend/teacher
 - ❖ GET: Get all recommend academic supervisors
- /recommend/project
 - ❖ GET: Get all recommend academic supervisors

Apply

- /applyProject
 - ❖ POST: Students and academic supervisors apply a project
 - ❖ GET: Get all project applications
 - ❖ PUT: Industry partners handle a project application
- /applyStudentProject
 - ❖ GET: Students obtain application records for individual projects
- /applyStudentGroupProject
 - ❖ GET: Students obtain application records for group projects
- /applyProject/<project_id>
 - ❖ GET: Get a project application detail

Feedback

- **/feedback**
 - ❖ POST: add a feedback for user
- **/feedback/<project_id>**
 - ❖ GET: Get a project feedback

Implementation challenges

There are complex relationships between the database models of our system, including one-to-one, one-to-many, and many-to-many. To handle these complex relationships, you need to understand the use of foreign keys and master the advanced usage of flask_sqlalchemy. Our system involves some complex business processes. For example, the process of creating a group, joining the group, and apply-related processes.

Project application

Project application is a complex interactive process, where data transmission and front-end/back-end interaction consume a lot of our time. We need to consider various situations, such as students can only apply for projects supervised by a supervisor, students must have their own group when applying for group projects, and if multiple users apply for a project, the industry partner will automatically reject other users after accepting one of them. These complex functions and condition judgments caused us to be confused for a while. We made multiple attempts, and through multiple debugging and testing, we found the direction and solved this problem.

Recommendation system

When designing the recommendation system for this project, we initially encountered many difficulties because we didn't know how to recommend the project and academic supervisors to other users. After a series of discussions, we decided to add an intention option to the user's resume. The system will recommend corresponding projects to users based on their resumes.

Installation/User document/manual

Environment:

Backend:

1. Install postgresql and python
2. Installation dependencies
3. pip install -r requirements.txt
4. Config .env

Here is our .envsample file

```
DATABASE_URI=postgresql://postgres:postgrespw@localhost:5432/3900pro
JWT_SECRET_KEY=iKun3900
MAIL_SERVER=smtp.qq.com
MAIL_PORT=465
MAIL_USERNAME=867105725@qq.com
MAIL_PASSWORD=nxyevhedviufbecc
```

Frontend:

1. Install node.js and npm

Run server and source code:

1. Here is our github link:

[unsw-cse-comp3900-9900-23T3/capstone-project-3900f13aikun](https://github.com/unsw-cse-comp3900-9900-23T3/capstone-project-3900f13aikun) created by GitHub Classroom

Download the zip file in github

2. In your IDE, run the command: **cd backend**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jjy\Desktop\capstone-project-3900f13aikun-main> cd backend
```

3. run the command: **Python -m flask run**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jjy\Desktop\capstone-project-3900f13aikun-main> cd backend
PS C:\Users\jjy\Desktop\capstone-project-3900f13aikun-main\backend> python -m flask run
E:\python3.10\lib\site-packages\flask_marshmallow\_init_.py:25: UserWarning: Flask-SQLAlchemy integration requires marshmallow-sqlalchemy to be installed.
  warnings.warn(
  * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

4. In the new terminal, run the command: **cd frontend**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jjy\Desktop\capstone-project-3900f13aikun-main> cd frontend
PS C:\Users\jjy\Desktop\capstone-project-3900f13aikun-main\frontend>
```

5. run the command: **npm install**

6. run the command: **npm start**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\jjy\Desktop\capstone-project-3900f13aikun-main\frontend> npm install
up to date, audited 1717 packages in 11s

259 packages are looking for funding
  run `npm fund` for details

9 vulnerabilities (2 moderate, 6 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\Users\jjy\Desktop\capstone-project-3900f13aikun-main\frontend> npm start
```

7. Then you can use our system follow the instruction we mentioned above

(in the **Functionality description and how to use them part**)

Reference:

- [1]Welcome to flask¶ (no date) Welcome to Flask - Flask Documentation (3.0.x). Available at: <https://flask.palletsprojects.com/en/3.0.x/> (Accessed: 18 November 2023).
- [2]JWT-extended's documentation¶ (no date) Flask. Available at: <https://flask-jwt-extended.readthedocs.io/en/stable/> (Accessed: 18 November 2023).
- [3]Flask-Mail¶ (no date) flask-mail - Flask-Mail 0.9.1 documentation. Available at: <https://pythonhosted.org/Flask-Mail/> (Accessed: 18 November 2023).
- [4]Sqlalchemy¶ (no date) Flask. Available at: <https://flask-sqlalchemy.palletsprojects.com/en/3.1.x/> (Accessed: 18 November 2023).
- [5]Marshmallow: FLASK + marshmallow for beautiful apis¶ (no date) Flask. Available at: <https://flask-marshmallow.readthedocs.io/en/latest/> (Accessed: 18 November 2023).
- [6]V6.19.0 (no date) Home v6.19.0. Available at: <https://reactrouter.com/en/main> (Accessed: 18 November 2023).
- [7] (No date) *Material UI: React components based on Material Design*. Available at: <https://mui.com/material-ui/> (Accessed: 18 November 2023).