COMP3900

COMPUTER SCIENCE PROJECT

PROJECT 23

Web Platform for Connecting Professionals with Projects
2023 T3

3900F11ABLIW

Liana Isaias-White	z5257234	Scrum Master / Frontend Member	z5257234@ad.unsw.edu.au
Brendan Ye	z5254583	Backend Member	z5254583@ad.unsw.edu.au
Ian Hou	z5324093	Backend Member	z5324093@ad.unsw.edu.au
Jinghan Wang	z5286124	Backend Member	z5286124@ad.unsw.edu.au
Raymond Li	z5258024	Frontend Member	z5258024@ad.unsw.edu.au

Submission Date: 17/11/2023

Table of Contents

Overview

Our team has created a professional web platform that connects companies offering projects with potentially interested individuals who are willing to carry on project work. This has been achieved by providing the base functionalities of a profile management system to a dashboard watchlist that introduces unique functionalities and social elements. Company users will be able to make modifications, updates, and manage their projects as necessary, whilst Professional users have the option to join projects listed by companies, give project ratings after completion, filter projects with advanced filtering options, and comprehensive profile editing functionalities, among others.

System Architecture

We used various third party libraries in both the Frontend and the Backend. To take a look in more detail, we have divided our applications into 4 different layers; **Interface Layer, Business Layer, Database Layer and Infrastructure Layer** (Figure 1)**.**

Interface Layer:
The frontend will be responsible for providing a user-friendly interface for professionals to interact with the platform. By using our expertise we have comprised with,

- *ReactJS* which will handle the dynamic behaviour of the application,
- *HTML* and *CSS* will be used for structuring and styling the user interface,
- *Material-UI* to create the UI and responsive components.

Business Layer:
The interface layer will consume the API served by the business layer. Using *FastAPI*, a modern python framework, the project can have convenient communication between the frontend and backend. Due to the API's ease of use, speed and robustness it will make our tech stack more compatible and overall simpler. From the API, the backend will be written in *Python* and serve as the business logic layer of the application. We decided on Python since it is well-suited for handling projects matching, user authentication and other core functionalities.

Database:
By considering the storage of user profiles, project data and other relevant information, *SQLAlchemy* is used as the object-relational mapping tool for interacting with the database and "give the full power and flexibility of SQL" (2023, SQLAlchemy). This database provides a pythonic way to work with databases making it compatible with our current management system. Also, it can be integrated with FastAPI for data retrieval operations.

Infrastructure:
The platform will be *UNIX-based* systems compatible. Having both a multitasking and multi-user operating system, we can develop numerous applications in a standalone workstation.

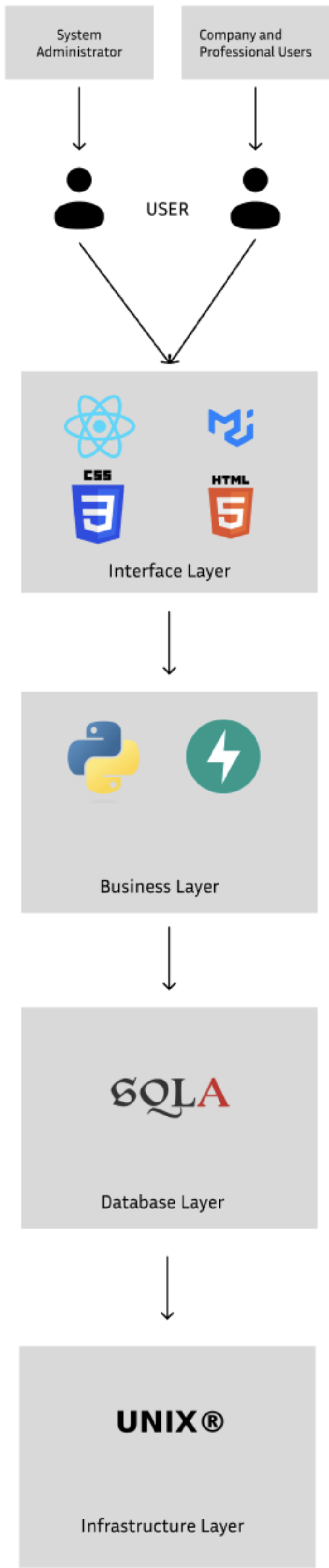


Figure 1: System Architecture Diagram

Functionalities and Objectives

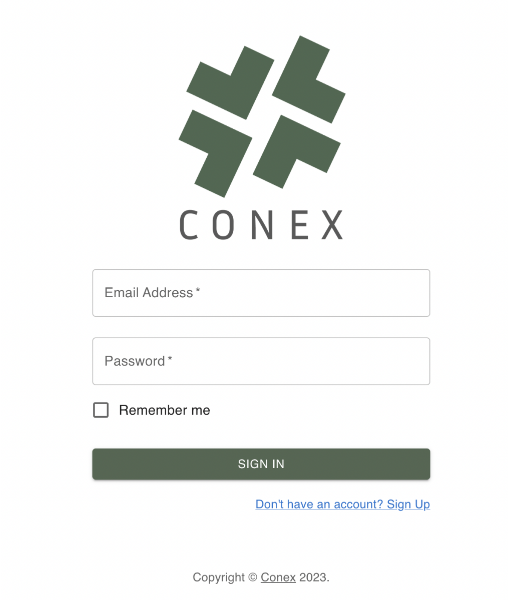
Functionalities

1. Login/Registering and Authentication

This functionality allows users to create an account to view, apply, create and edit projects depending on their user type. Upon first visiting CONEX, users will be required to register or login to their account in order to perform certain functions based on their permission levels (Figure 2, 3). Once logged in or registered, they will remain logged in for 1 hour before being logged out again. All functionality within the website cannot be performed without first logging in or registering.

This functionality maps to the following objectives:

- *Users with different roles and responsibilities can create their account, login and operate the system based on their roles and responsibilities.*
- *A company can create/register their account and login.*
- *A professional (individual looking to join projects) can create/register their account and login.*



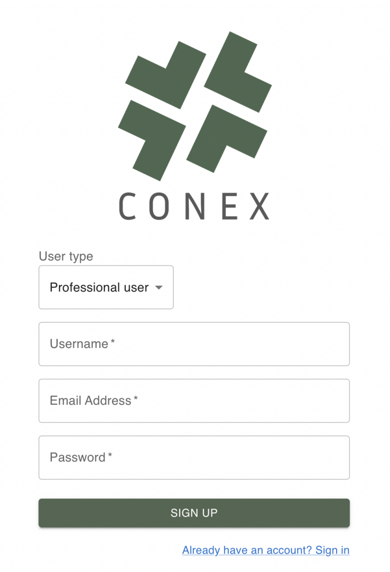


Figure 3: Register Screen

Figure 2: Login Screen

2. Edit Profile and Details

This functionality allows any user of CONEX to maintain their own profile details (Figure 4). This includes making any changes to their personal or company account. Professional users are also able to update their qualifications and professional biography whilst company users can edit, delete or make updates for projects made (Figure 5).

This functionality maps to the following objectives:

- *Users can edit/maintain their profiles.*
- *A company can edit their account by updating basic account information.*
- *A professional can edit and update their account/project.*

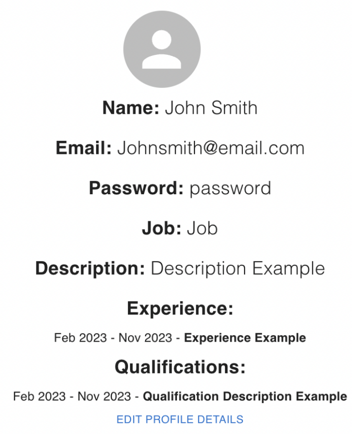


Figure 4: Profile Details

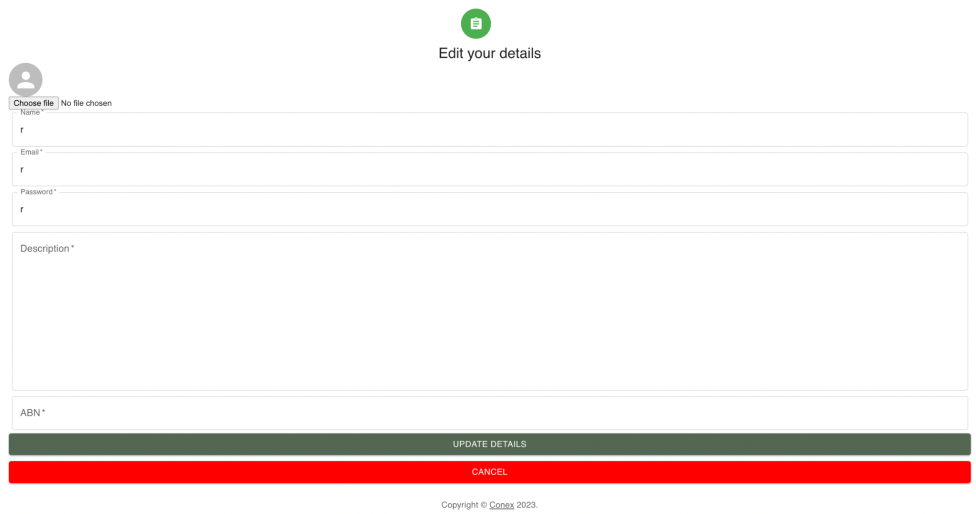


Figure 5: Profile Details Edit

3. Dashboard Details

This functionality allows users to view relevant projects. They will be able to see key information such as the project title, the owner, the project's start and due date and tags. Users are able to click on project titles to view the full details of that project.

Professional users will see a dashboard which has two tabs (Figure 6). The first tab shows all available projects. The second tab shows any project that they have started an application for or applied to but have yet to be accepted or rejected by the project owner. Professional users will also be able to apply to the project directly.

Company users will see a dashboard which has two tabs. The first tab shows all the projects that this company user has created. The second tab shows all available projects created by any user.

This functionality maps to the following objectives:

- *A company can view their project lists and its details (and filter it) and can edit any project information.*
- *A company can view a list of all available projects (project lists) and can choose to see the details of any project with filters.*
- *A professional can select to view the full details of any project from the search results.*

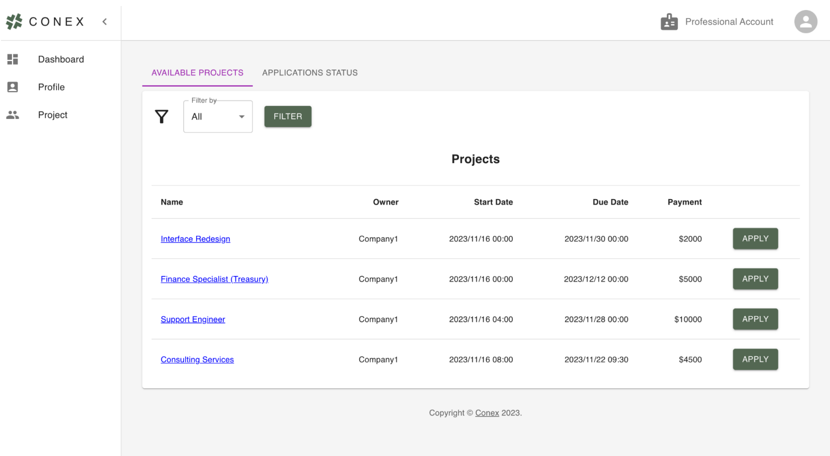


Figure 6: Dashboard

4. Search and Filter Functionality

This functionality allows users to search for projects from the dashboard using a filter so it is easier within a big database (Figure 7). This filter includes the status of the project (open, ongoing, complete or closed), number of project members, title and listed price. We also implemented a search bar to aid users in finding projects by name.

This functionality maps to the following objectives:

- *A company can view their project lists and its details (and filter it) and can edit any project information.*
- *A company can view a list of all available projects (project lists) and can choose to see the details of any project with filters.*

- *A professional can search and browse a list of available projects and refine the results with search filters.*

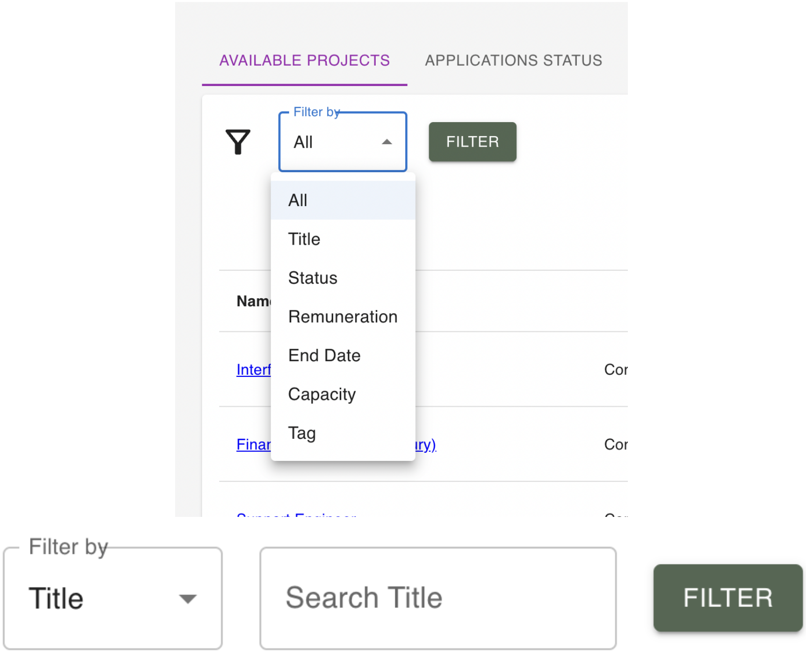


Figure 7: Filtering

5. Create and Edit Projects

This functionality allows company users of CONEX to be able to create projects for professional users (Figure 8). They are able to add information to the project itself such as the project title, description, payment, start and end dates, and the capacity (Figure 9). The company user will also be able to add application questions for applicants that want to join. After creating a project, they will see it on their dashboard and upon clicking the project, they will see a pen icon in the top right corner which allows them to edit the project when clicked (Figure 10). The company user can update all the fields of the project.

This functionality maps to the following objectives:

- *A company can create a new project by providing project information.*
- *A company can view their project lists and its details (and filter it) and can edit any project information.*

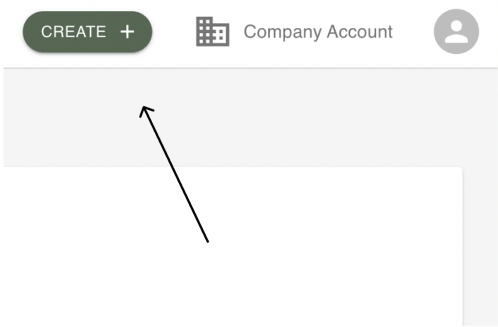


Figure 8: Create Button

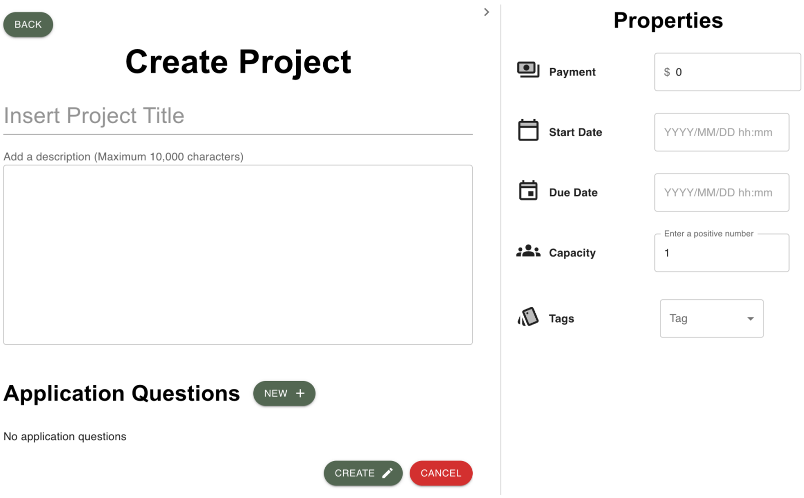


Figure 9: Create Project Page

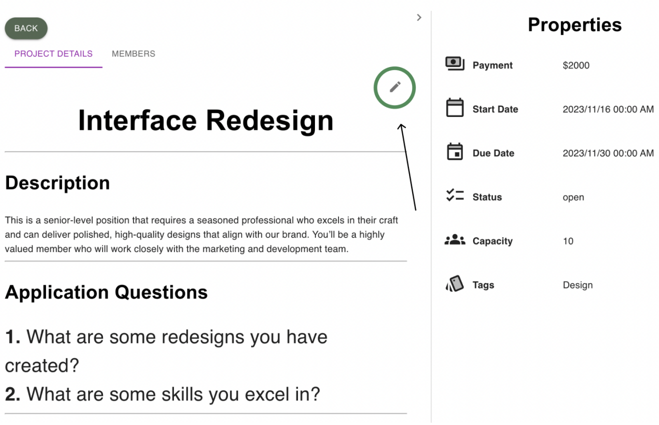


Figure 10: Project Page Edit

6. Manage Projects

This functionality allows company users to manage the applicants of their projects. Upon viewing the full details of a project, they can switch tabs from the project information to the members. The company user will see two sections. The first section shows the professional users who have been approved to work on the project. The second section shows the professional users who have applied to the project.

The company user can manage the applicants and approved users of the project by clicking the edit icon button on the top right of the page's main section (Figure 11). This will allow the company user to remove a member who was previously approved to work on the project. The company user will also be able to approve and reject any professional user applications (Figure 12).

This functionality maps to the following objectives:

- *A company can approve/reject individuals requested to join the project.*
- *A company can view/edit the list of professionals who are approved to join a project.*

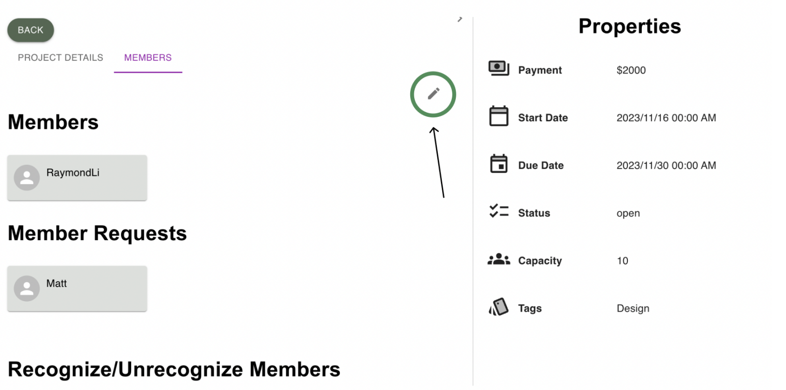


Figure 11: Member List

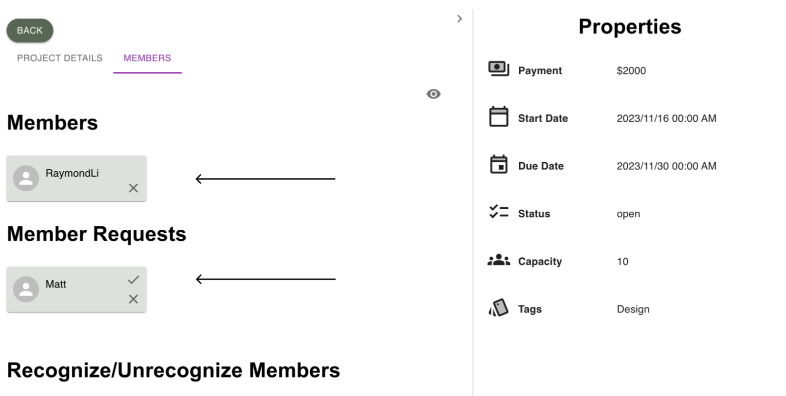


FIgure 12: Member List Edit

7. Project Details and Applications

This functionality allows professional users to be able to view the project and its detailed credentials. This includes a sidebar of all the specifics whilst outlining the description of the project itself (Figure 13). It will provide a greater depth of the project for users to inform themselves on the specifics and the project's aim. For a professional user, they will thus be able to apply for the project via a button if interested, being asked their qualifications and answering application questions made by the project owner (Figure 14, 15).

This functionality maps to the following objectives:

- *A professional can select to view the full details of any project from the search results.*
- *A professional can request to join/apply for a project of interest (this includes sharing relevant information from their profile).*
- *A profession can see the project page if they are approved by a company.*

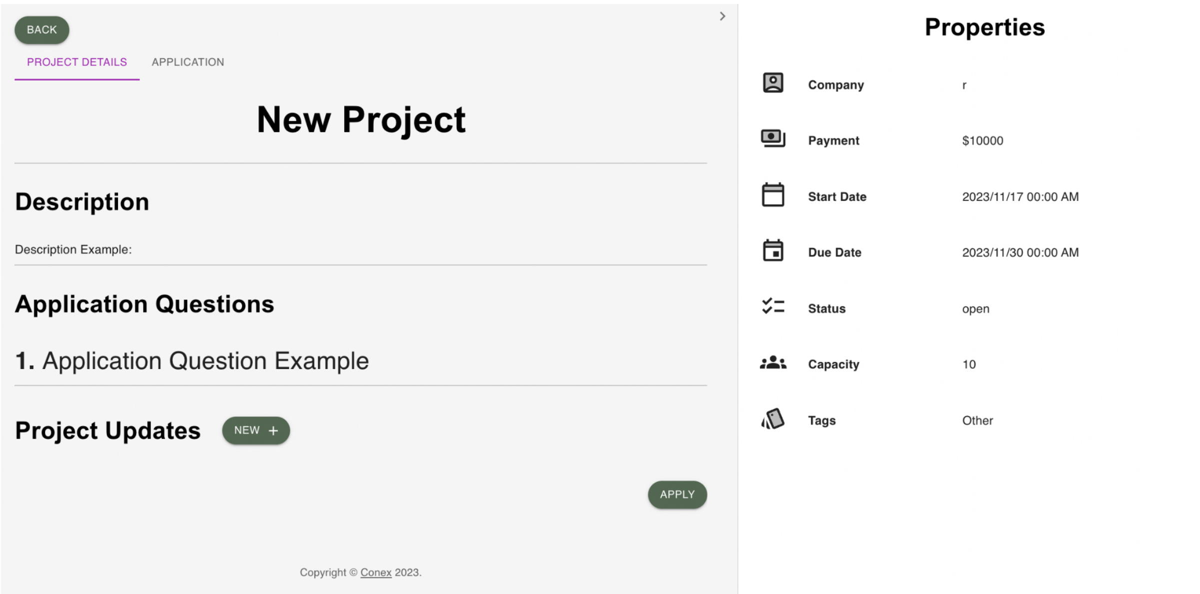


Figure 13: Project Detail

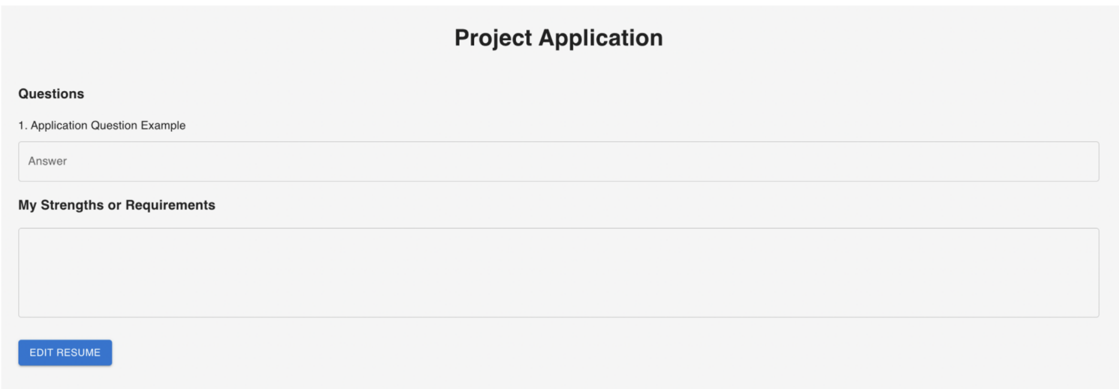


Figure 14: Project Application Process

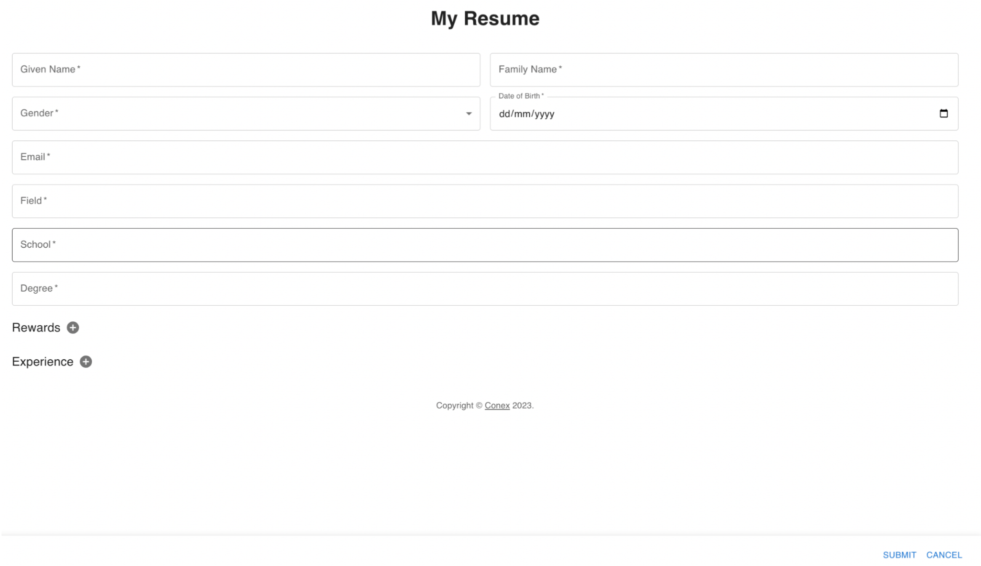


Figure 15: Project Application Resume

8. Inbox Notification

This functionality allows both the company and professional user to be notified via email on their project request (Figure 16, 17). Upon application request, both parties will be notified on their receiving end. Company users will be able to check their email inbox and can approve or deny the member request from the professional user's application.

This functionality maps to the following objectives:

- *A professional receives a notification of their request to join a project.*

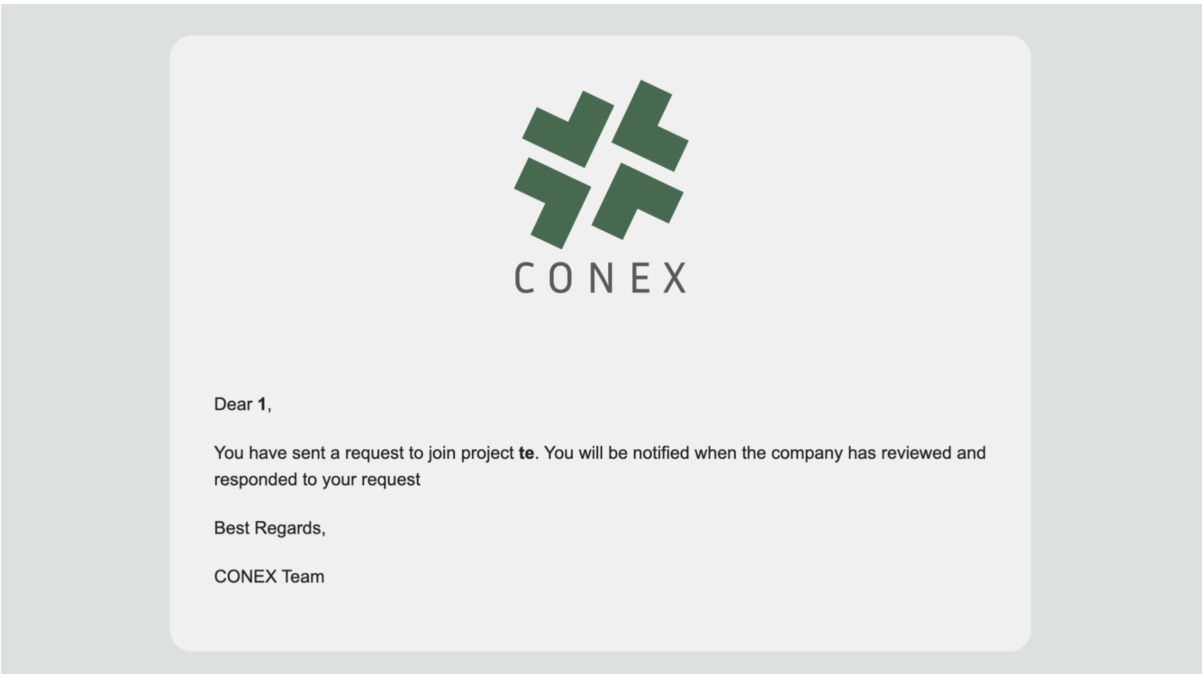


Figure 16: Email Inbox for Application Request

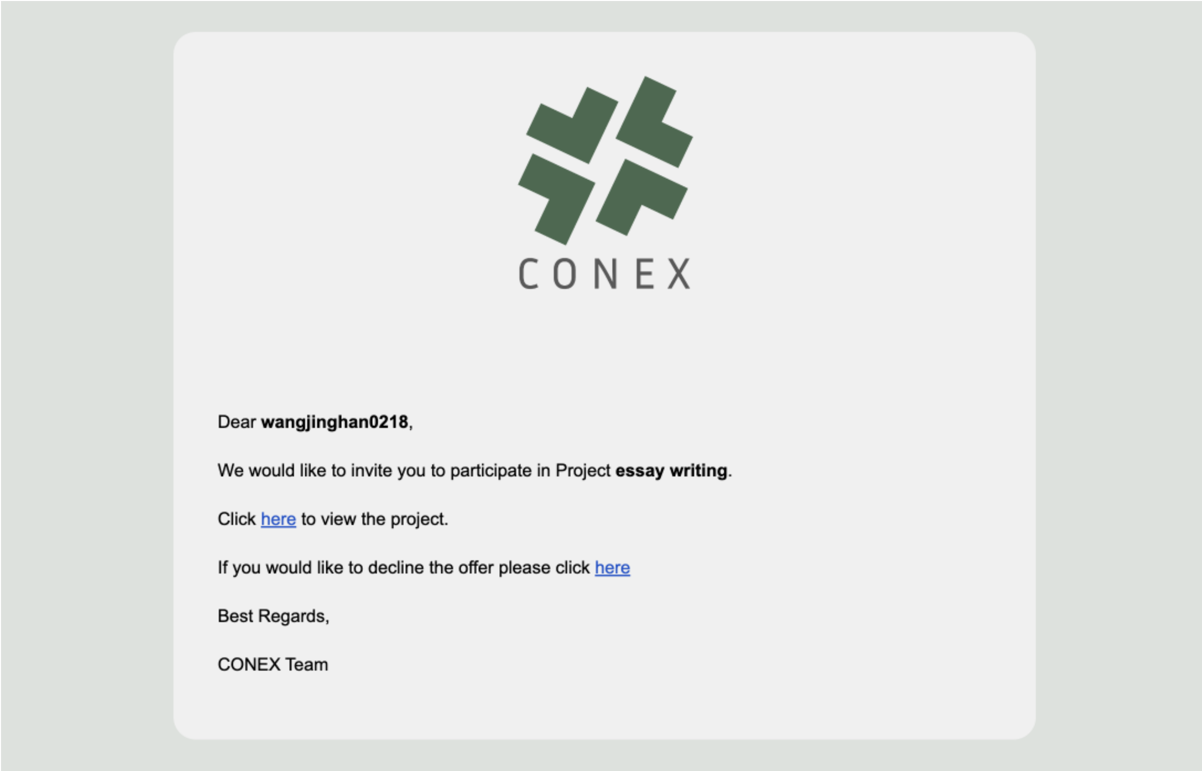


Figure 17: Email Inbox for Accepted Request

9. Progress Logs

This functionality allows participating users and the project owner to communicate through a chat/progress log with the on-going project (Figure 18). Both parties can message each other and view previous log messages within the given project.

This functionality is an additional functionality which was added on top to differentiate it from existing systems and improve on the given project objectives. Thus, it does not map to any of the given specification objectives.

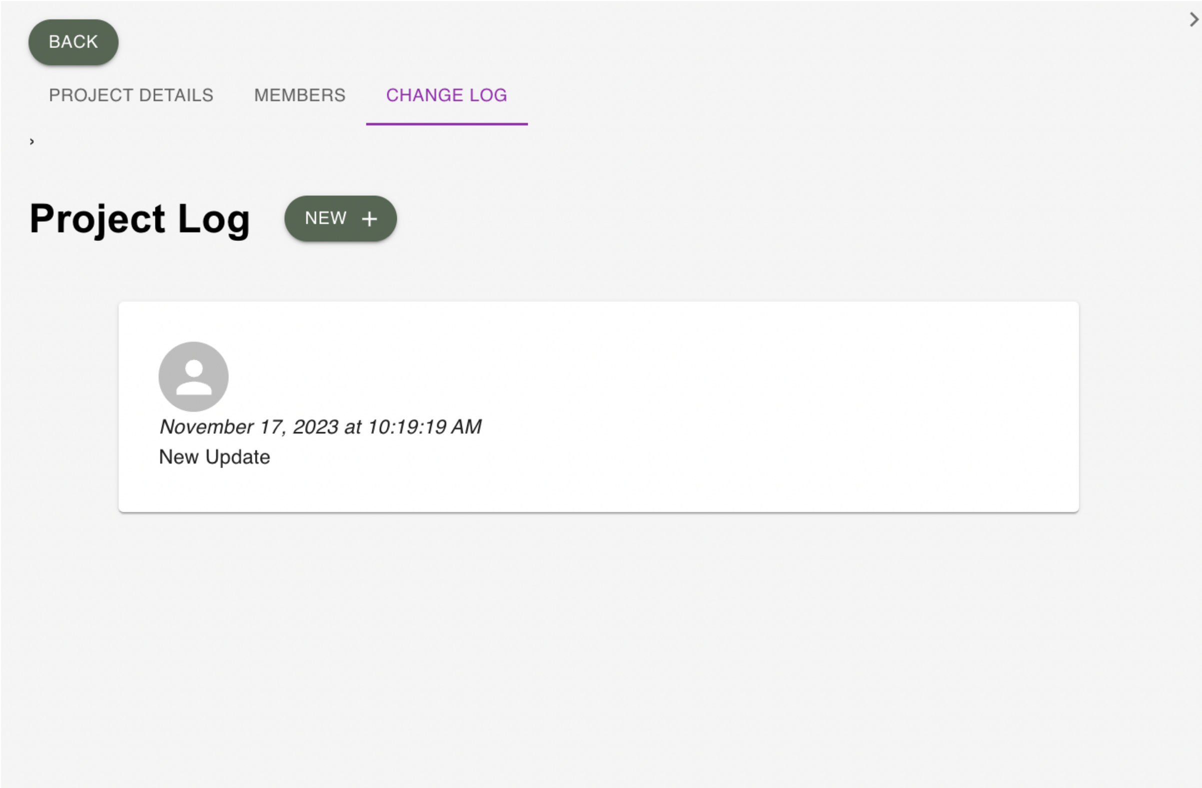


Figure 18: Project Logs/Chat

10. Rating Completion

This functionality allows users from both participating parties to rate the project itself and the members involved (Figure 19). The rating system consists of a 5 star rating, and upon completion of the project they are awarded on their account details (Figure 20).

This functionality maps to the following objectives:

- *A company can rate and recognize the participants who contributed to their projects at the end of the project.*
- *A professional can rate their experience with the project when the project is completed.*

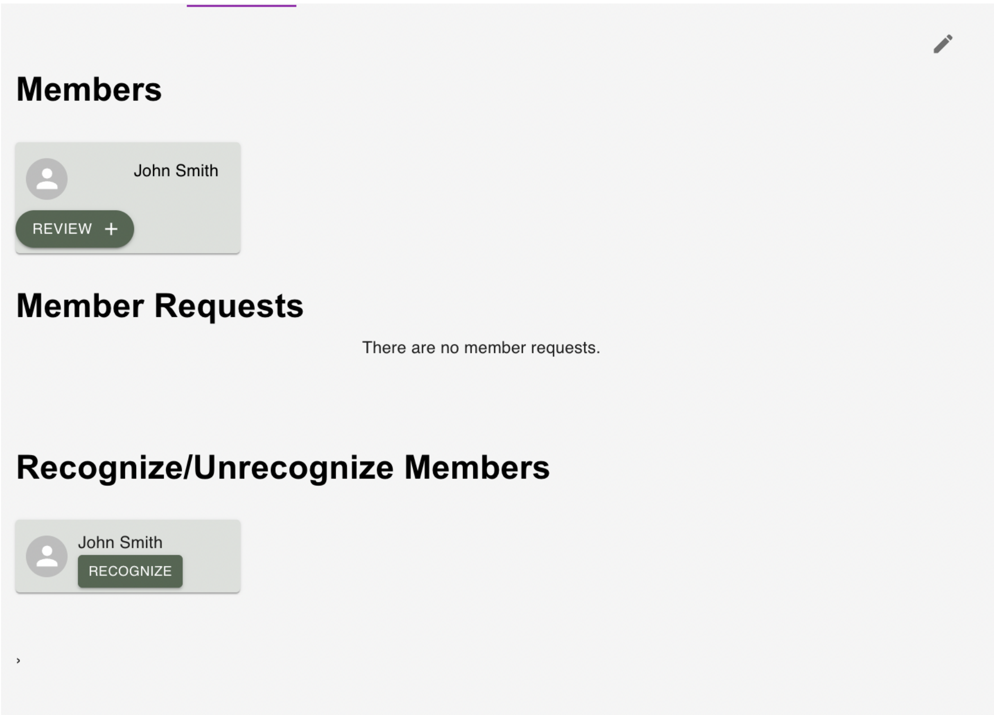


Figure 19: Company User Review + Recognised

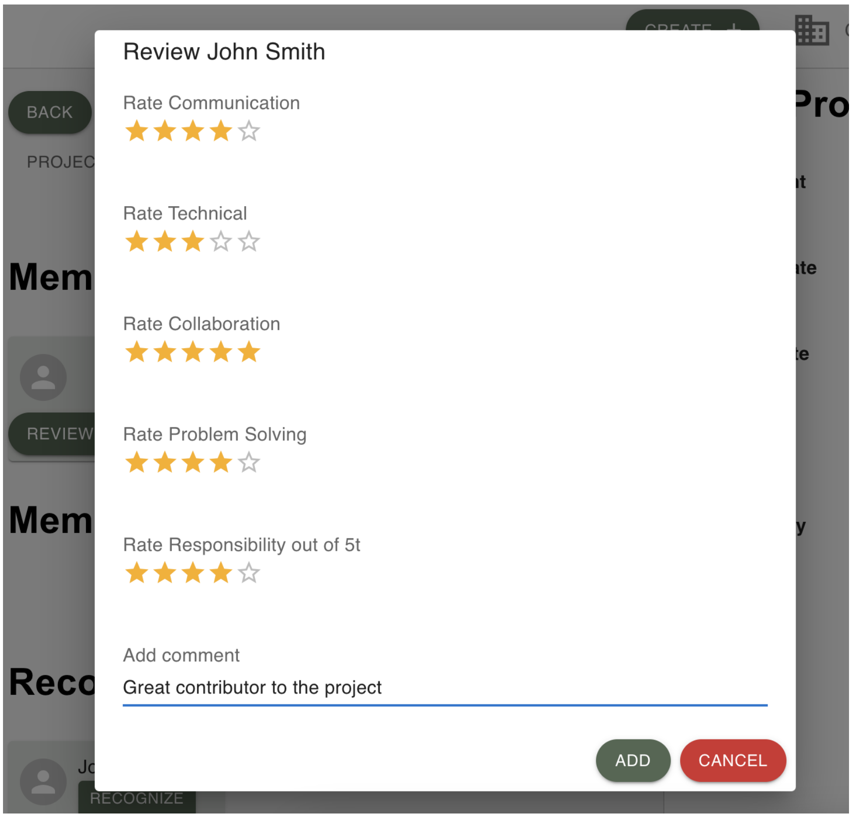


Figure 20: Rating Applicants

11. Certificates and Achievements

This functionality allows company users to reward certificates of recognition and for professional users to obtain the certificate (Figure 21). Upon successfully completing a project, the professional user will be awarded a certificate which they can view on their profile page and on the project (Figure 22).

This functionality maps to the following objectives:

- *Upon successful completion, a professional can obtain a certificate of recognition from the company offering the project.*
- *Professionals can browse a list of achievements (completed projects and reviews).*

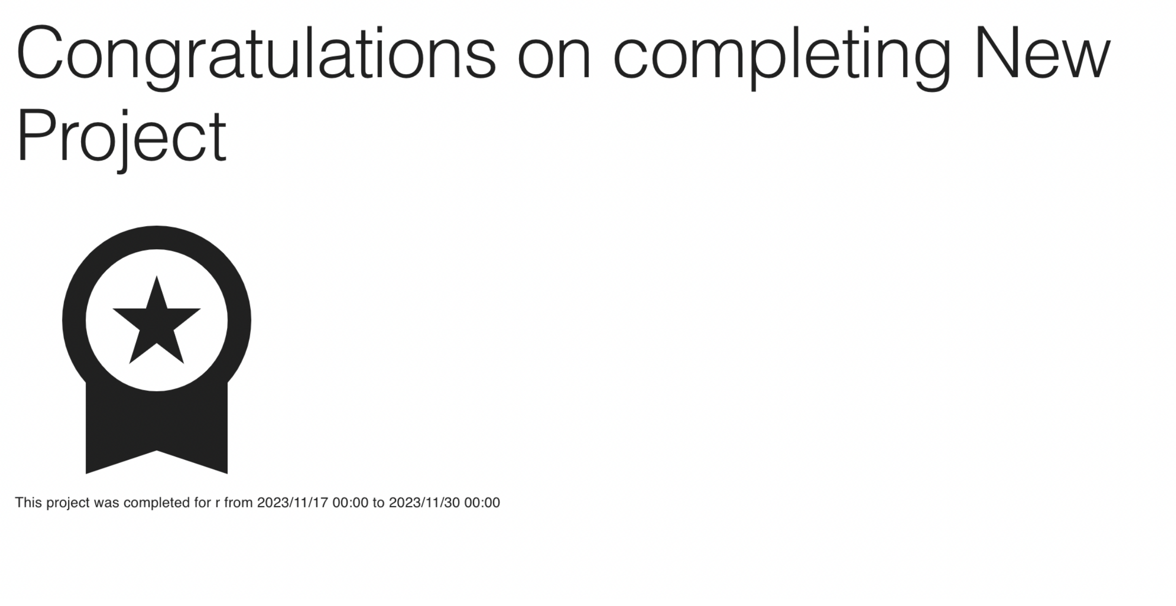


Figure 21: Certificate for Project Completion

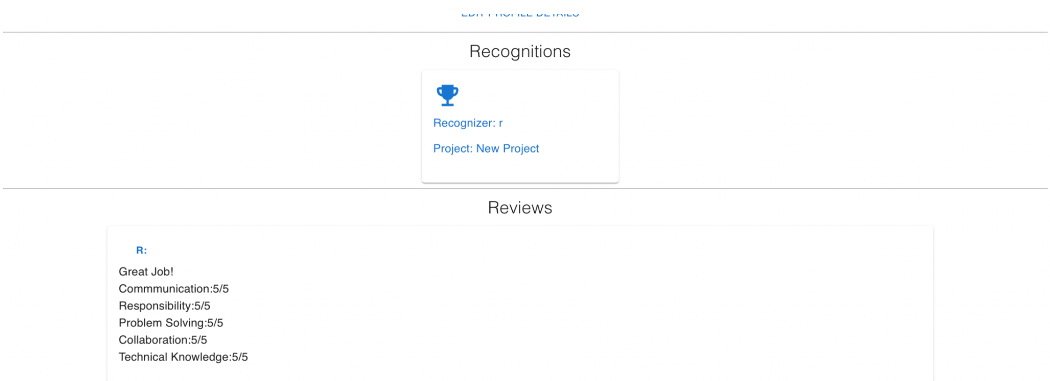


Figure 22: Profile Achievements

Mapping Objectives to Functionalities

While the above *Functionalities* section does show which objectives are mapped to each functionality, Table 1 below explicitly maps the project objectives to the functionalities implemented. Each project objective listed in the specifications is represented by a row in the table, with the actual objective in the first column. Each objective will have the corresponding functionalities which achieve this objective as the second column.

Table 1: Mapping of Project Objectives to project functionalities

Project Objective	Functionality
Administrator users have the full privilege in the system and should be able to manage and oversee the platform.	● All functionalities
Users with different roles and responsibilities can create their account, login and operate the system based on their roles and responsibilities.	● Login/Register and Authentication
Users can edit/maintain their profiles.	● Edit profile and details
A company can create/register their account and login.	● Login/Register and Authentication
A company can edit its account by updating basic account information.	● Edit profile and details
A company can create a new project by providing project information.	● Create and Edit Projects
A company can view its project lists and their details (and filter them) and can edit any project information.	● Dashboard Details ● Search and Filter Functionality ● Create and Edit Projects
A company can approve/reject individuals requested to join the project.	● Manage Projects
A company can view/edit the list of professionals who are approved to join a project.	● Manage Projects
A company can view a list of all available projects (project lists) and can choose to see the details of any project with filters.	● Dashboard Details ● Search and Filter Functionality
A company can rate and recognize the participants who contributed to their projects at the end of the project.	● Rating Completion
A professional (individual looking to join projects) can create/register their account and login.	● Login/Register and Authentication
A professional can edit and update their account/project.	● Edit profile and details
A professional can search and browse a list of available projects and refine the results with search filters.	● Dashboard Details ● Search and Filter Functionality
A professional can select to view the full details of any project from the search results.	● Dashboard Details
A professional can request to join/apply for a project of interest (this includes sharing relevant information from their profile).	● Project Details and Application

A professional receives a notification of their request to join a project.	● Inbox notification
A professional can see the project page if they are approved by a company.	● Project Details and Application
A professional can rate their experience with the project when the project is completed.	● Rating Completion
Upon successful completion, a professional can obtain a certificate of recognition from the company offering the project.	● Certificates and Achievements
Professionals can browse a list of achievements (completed projects and reviews).	● Certificates and Achievements

Third-Party Functionality

Interface Layer

The main third party library implemented was React. We decided to choose React as our frontend framework due to the experience the frontend developers had in React and the great UI component libraries it provides. We developed a single page application, improving the speed of the application hence the scalability as well. React's component based structure allows us to reuse components saving time on development; endorsing a consistent design and ease of expansion. Due to its popularity, the framework allowed us as frontend developers to gain a lot of resources and support online to work through and debug which was especially important as we had members still learning, thus making development easier and understandable.

Furthermore, we decided to build our application on a free online template built on top of Material UI (MUI) for React that provides a base for us to get started. This template provided a well set structure, additional components, icons and some page components that provided us with a cohesive design language throughout the site. Using this third party library allowed us the developers to focus more on the functionality rather than making new designs for the web page, reducing development time and saving redundant features from being created. Thus, we were able to design a website that has a consistent frame, while having a comprehensive set of functionalities.

MIT vs. commercial licenses

The MUI team has been building MIT-licensed React components since 2014 with Material UI. We are committed to the continued advancement of the open-source libraries. Anything we release under an MIT license will remain MIT-licensed forever. You can learn more about our stewardship ethos in this document from our company handbook.

Community plan

The community version of MUI X is published under an MIT license and is free forever. This version contains features that we believe are maintainable by contributions from the open-source community.

Figure 23: MIT Licensing for React/Material UI

The licensing for Material UI (See: Figure 23) grants us permission free of charge to use copy and modify at their community pricing level because of its inherited MIT license. This application library can be used to do whatever we'd like with the web we designed using the templates from the library.

Business Layer

The team has decided in favour of Python compared to other languages and frameworks such as Node.js or PHP to develop our backend code. This has been selected as the programming language due to its ability to efficiently develop a web application and our developer capabilities. Its integration with many online APIs to get, request, post and patch data and its convenience in a variety of libraries can reduce the number of lines of codes and hence allow for dynamic typing.

Open-source

Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation.

Figure 24: Python Licensing

For any further versions of python after 3.8.6, the licensing of its software and documentation is provided under the Python Software Foundation License Agreement (See: Figure 24). Specifically using version 3.9, this coding language has become beneficial in our program.

With the addition of FastAPI, one of Python's fastest frameworks, we were able to make use of its supportive features in our project. For its short and intuitive coding documentations we were able to include it into our backend. As the API follows the MIT License, we had the rights to use it efficiently to extend the functionality to suit the project.

Database Layer

We used SQLAlchemy for our project due to its ease of use. It is an open-source SQL toolkit that also is run through the MIT License. It is a high-level database that can give the full flexibility of SQL. Since it is used through the MIT License, the software is able to be published and used through the code without consideration.

Infrastructure Layer

Our application is able to run on UNIX operating systems as it is a web application. This operating system supports all third party libraries and languages required to run the application. Specifically using the web browser, Google Chrome, it must be installed to access all functionalities of our website. This is due to the inherited features within ReactJS and if used otherwise in other web platforms, issues will occur.

Implementation Challenges

Authentication Management and API Security (Tokens)

After registering or logging in, a user, whether it be a company user, professional user or admin user, the user's identity is stored, along with their session. The user's identity is often displayed in a small piece of authentication information called a "*token*". It is important to store this token (authentication information) in a secure location and make it hard to alter, otherwise people can access it. For example, it will be a major problem if a hacker without access to the website is able to obtain a token and pretend to be a user. Only authenticated users with a correct token will be able to access the functionalities of CONEX.

Furthermore, different users will have different permissions. If a hacker is able to obtain and alter a token to change their permission levels from a professional user to a company user, then this will be a large security concern. They will then be able to add themselves to projects or change the project status to completed, effectively earning money for work that they did not do. Therefore, it is extremely important to correctly and securely store a user's token..

CONEX uses tokens to identify users and subsequently, changes its view to fit the user's permission levels (either professional, company or admin). The naive way of creating a token is to have the backend generate a count-based token. Every time a user logs in, the token value increases by 1. This token is then mapped to a user ID, which identifies the user and its user type. Whilst this method is very easy to implement, it is simply insufficient. If a hacker is able to access this information they will be able to change it at will, especially if it is stored publicly and unencrypted.

Therefore, a better way to create the token is to use a hashed value of a user's ID. Even if a hacker is able to obtain the token and alter it, it will be very difficult to actually alter the token to a value that aligns with the hashing function. Only if the changed token aligns with the hashing function will the token be able to work as user authentication.

CONEX uses this method of encryption for security. Specifically, CONEX uses JSON web tokens (JWTs) with a *`HS256`* hashing algorithm to encode a user's ID, with a hidden secret. Without this secret, a hacker will not be able to decrypt the token, making it virtually impossible for the hacker to change their authentication and permissions. Furthermore, all API requests that are sent with a header that contains the JWT will be encrypted, adding a further layer of security (IBM, 2023).

Next, comes the challenge of storing the token in a safe location. The naive method that is taught at many universities is to store it in `localStorage`. However, this is a major security vulnerability. The major one is that it cannot be accessed on the server-side (the backend), and thus, its management is entirely on the client side (frontend). This makes it susceptible to cross-site scripting attacks, where people can access the frontend JavaScript code directly. `LocalStorage` also must be removed manually. JWTs have an expiry time and the manual removal of JWTs at the exact expiry time is difficult and will slow the app down. Furthermore, if done incorrectly, it will lead to many errors on the client side (the frontend) as API requests that require authentication will be called and the token (stored on the localStorage) will be incorrect. Thus, a better solution is required.

For CONEX, the tokens were stored in a cookie. Cookies can be set on the server side, which is what CONEX does. This helps alleviate some of the cross-site scripting attacks. By setting a httpOnly flag, the cookie cannot be accessed using JavaScript at all, and so this further prevents cross-site scripting. Therefore, CONEX utilised this httpOnly flag.

Additionally, cookies can be set to transmit only via encrypted connections (*`https`* rather than *`http`*), which helps mitigate man-in-the-middle attacks (note: this safe transmission still works for `http://localhost`). This can be achieved by setting the secure flag to true, which CONEX does in the server-side (the backend) when setting a token.

Finally, cookies can be set to have expiry times. By setting an expiry time to be equivalent to that of a JWT, there will be no need to manually remove the cookie and worry about syncing this manual removal with the JWT expiry. Instead, by setting a JWT and then immediately setting a cookie in the server-side (the backend), the expiry of the token on the server-side and the client-side is automatically synced. This effectively eliminates the possibility of errors on the client side where API requests are sent with expired authentication.

CONEX sets an expiry time of one hour. After one hour, the user will be automatically logged out after a re-render is triggered on the client side. This short time was set to prevent the token cookies from being stolen. A common tactic used to maintain log-in without a logout will be to have a refresh token which refreshes the token that is in the cookie at specified times whilst also having a token that is set for a longer period of time. The corresponding JWT in the backend will also refresh. However, as CONEX uses ReactJS in the frontend, it is a single page application (SPA), refresh tokens are not ideal. Since a refresh token will also include a token that has a long lifespan existing on the client-side, there is a possibility of it being stolen. This is especially relevant for SPAs as SPAs cannot authenticate the application against the token endpoint, which would prevent the long-living token from being stolen.

Summary

CONEX achieves correct and authentic authentication storage by using JSON web tokens (JWTs) to relay information about a user's identity and their session. Tokens are stored in the cookies and are a hashed value. These token cookies have a short lifespan (1 hour) to prevent them from being stolen. For further security consideration, tokens are set to httpOnly which prevents JavaScript from accessing the cookie, which mitigates attacks against cross-site scripting. The secure flag is also set which only transmits a cookie over an encrypted connection, which helps mitigate man-in-the-middle attacks.

The user is automatically logged out after expiration. Alternatively, the user can also manually log out using a button. This button simply sets the expiry time of the token to 1 seconds from now.

Email Notifications

The feature primarily offers a reminder service via email for professional users, such as during submission of applications, receiving company user approval or rejection of applications, project completion, and receiving ratings.

The backend utilises Python's smtplib and email libraries. Due to limited email-sending requirements in testing, an existing Gmail account was employed as the sender. Each email is crafted using HTML for its body, ensuring compatibility with most popular email services while maintaining aesthetic appeal. However, given that some email services like Gmail do not support embedded Base64-encoded images or SVGs in HTML, images are uploaded to Imgur, and URLs are used to ensure proper display.

During testing, it was observed that the SMS execution could occupy a considerable amount of runtime, potentially causing delayed backend responses. To address this, the SMS functionality was moved to a background task in FastAPI, ensuring backend APIs are not bogged down by prolonged SMS sending processes. This integration does not affect normal usage; even in the event of errors, it does not cause issues like database rollbacks.

The redirect links are defined in the backend and embedded into the HTML. They are activated upon accessing the frontend, which then redirects and displays the relevant function. All redirect links start with "$frontendUrl/jump/xxxx." On devices not logged in, the frontend saves the redirect link and routes to the login screen. Post-login, it follows the redirect logic to the appropriate page.

This functionality is contingent on the deployment location of the frontend. As login and email sending/receiving use public internet services, the redirect feature functions correctly when deployed on servers with public IPs or those using services like FRP for internal network penetration. For deployment in local networks or localhost, it is crucial to ensure these redirect addresses are accessible within the network where the application frontend operates.

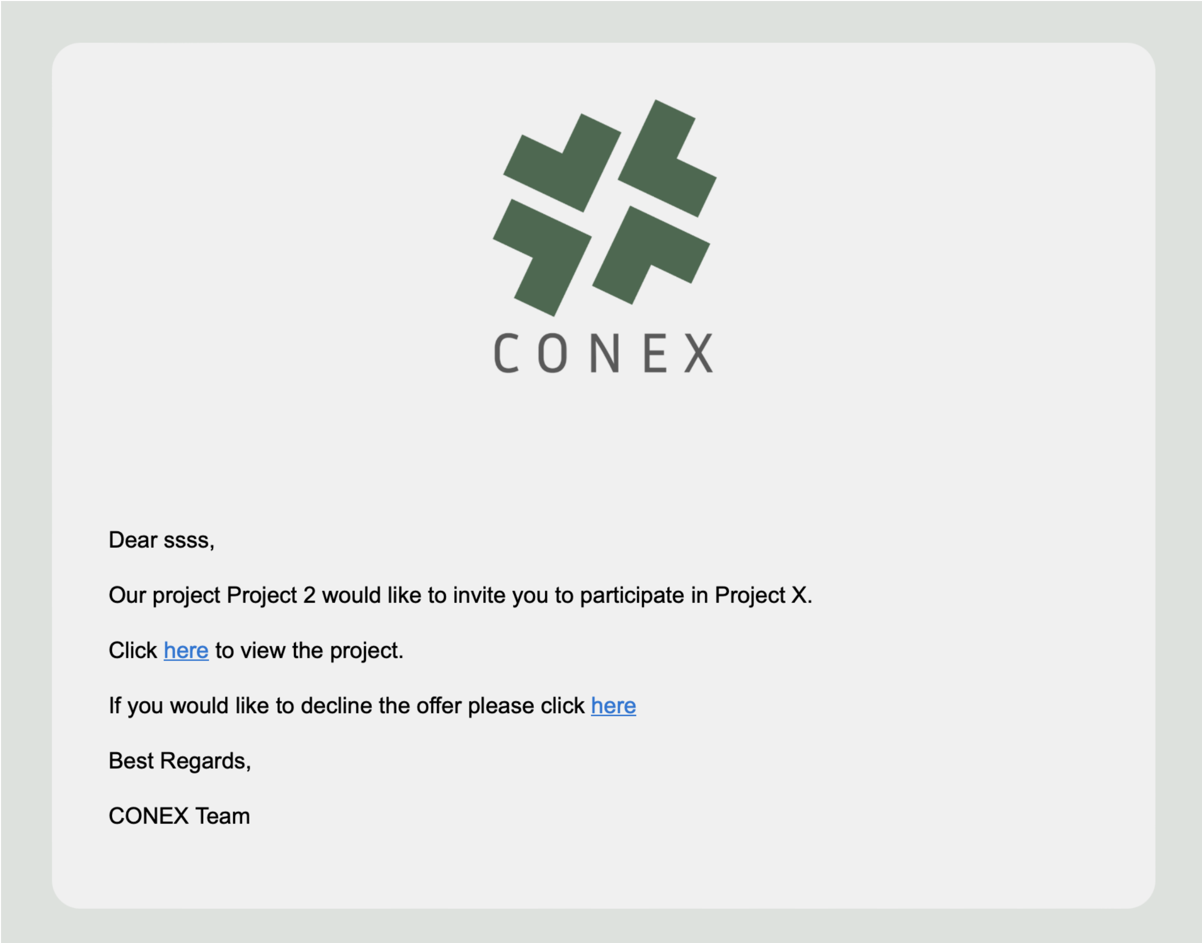


Figure 25: Email Request

The email content received by the user, as depicted in the image, includes essential information along with two redirect links (Figure 25). Both links require user login to display specific content. Particularly for the second link, it provides a functionality for the professional user to decline a project invitation. To mitigate accidental click, even when the frontend possesses the token, a direct click on the link does not immediately refuse the project. Instead, after validating the token, the user must explicitly click a button to reject the request, thereby enhancing user experience by adding a layer of confirmation. This design choice not only adds a security measure but also ensures that users make deliberate decisions, thereby preventing inadvertent actions that could disrupt the workflow or project management.

Installation Manual

Setup

NOTE: For a better viewing experience, please read the README.md located at the top-level directory.

Our frontend server setup is handled by npm, while our backend server setup is handled by `uvicorn`. In this setup we have also chosen to primarily create a requirement of using Python version 3.9 or higher to run the backend. To begin:

1. Download and extract the .zip from the master branch
2. Navigate to '`capstone-project-3900f11abliw`' directory
3. Navigate(command: `cd backend`) into the extracted backend folder on your terminal (folder name: `backend`)
    a. Create a file (`config.ini`)
    b. Open the file and configure your email details following the format below, if you don't want to use any real email and get notifications, just set the valuie open to any number other than 1(See the email setup part below for more information.)
    c. `[Email]`
    d. `Open = 1`
    e. `Sender = xxxxxxxx@xxxxx.com`
    f. `Password = xxxxxxxxxx`
4. Navigate back to the root of the folder (command: `cd ..`)
5. Ensure that you have `pip3` installed on your system. It is usually pre-installed once `Python3` is installed. Install `pip3` depending on your system if you do not already have it:
    a. Linux users use the command: `sudo apt install python3-pip`
    b. Mac users use the command: `brew install python3`
        i. Note that `pip3` is installed with `python3`
6. Run the following command from the root of folder (folder: `capstone-project-3900f11abliw`)
    a. '`pip3 install -r requirements.txt`'
    b. '`uvicorn backend.src.server:app --reload`'
7. Once the script has finished running, open a second tab in the terminal
8. Navigate(command: cd frontend) into the extracted frontend folder on your terminal (folder name: frontend)
9. Run the following command from the frontend folder
    a. `npm install`
    b. `npm start`
10. Note that once this script has finished running, it will take a few seconds for the website to load on your web browser.

The frontend will be available at '`localhost:3000`', and the backend API documentation will be available at '`http://127.0.0.1:8000/docs`'.

Email Setup

This email account will be utilized for providing SMS services to users.

If you prefer not to use it, set the 'open' value to any number other than 1.

The email functions normally with public email service providers. For the password, you may use either a plaintext password or an app-specific password as per the email service provider's requirements, with the latter being more secure. Here is a method for setting up an app-specific password with Gmail: [https://support.google.com/accounts/answer/185833].

Troubleshooting

- If there is a CORS issue at the front end that stops you from transferring data to the backend, try to delete the database file inside the project repository and rerun the whole system.
- For some system runtime errors that occur unexpectedly, we provided many specific error messages to indicate what error it is. For this project, we are unable to fix it but if it's in a real work situation, send the error messages and error code to the admin team to decrease the debugging difficulties and improve the error handling ability of the system
- If the current port numbers 3000(for frontend) and 8000(for backend) have been taken up by other applications, the application may not function properly due to the CORS protocol again. To make it work, inside the server file at the backend, change the frontend port number in the array called origins(shown below) into your current one to make it work.

```
app = FastAPI(debug=True)

origins = [
    "http://localhost:3000",
    "http://localhost:8000",
    "http://127.0.0.1:3000",
    "http://127.0.0.1:8000",
]
```

Figure 26: Port URL Addresses

Implemented Functionality

Logic Diagram

For our implemented functionalities, we created a logic diagram to show how each component operates. As you can see they are sectioned out into different logic elements in which we will display how it is implemented in our application (Figure 27, 28).

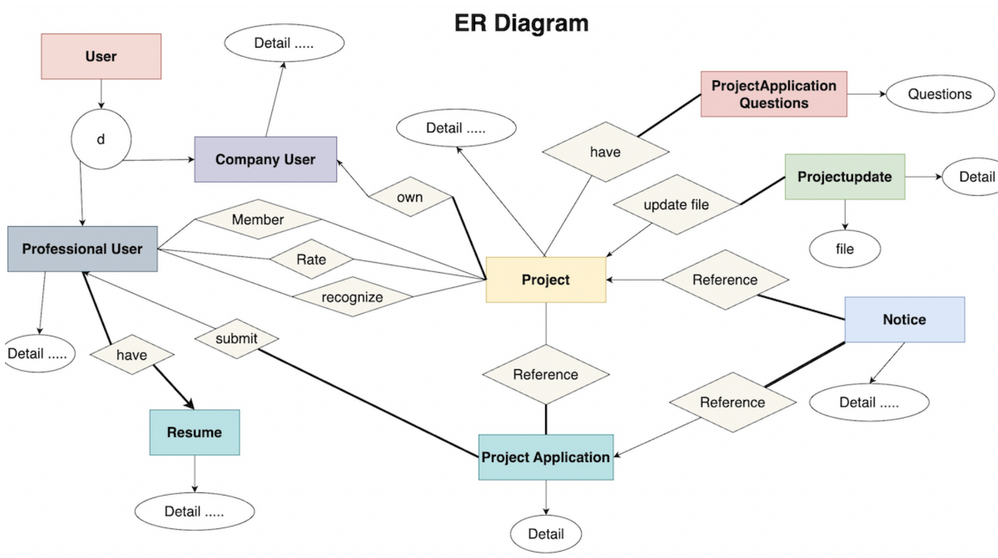


Figure 27: ER Diagram

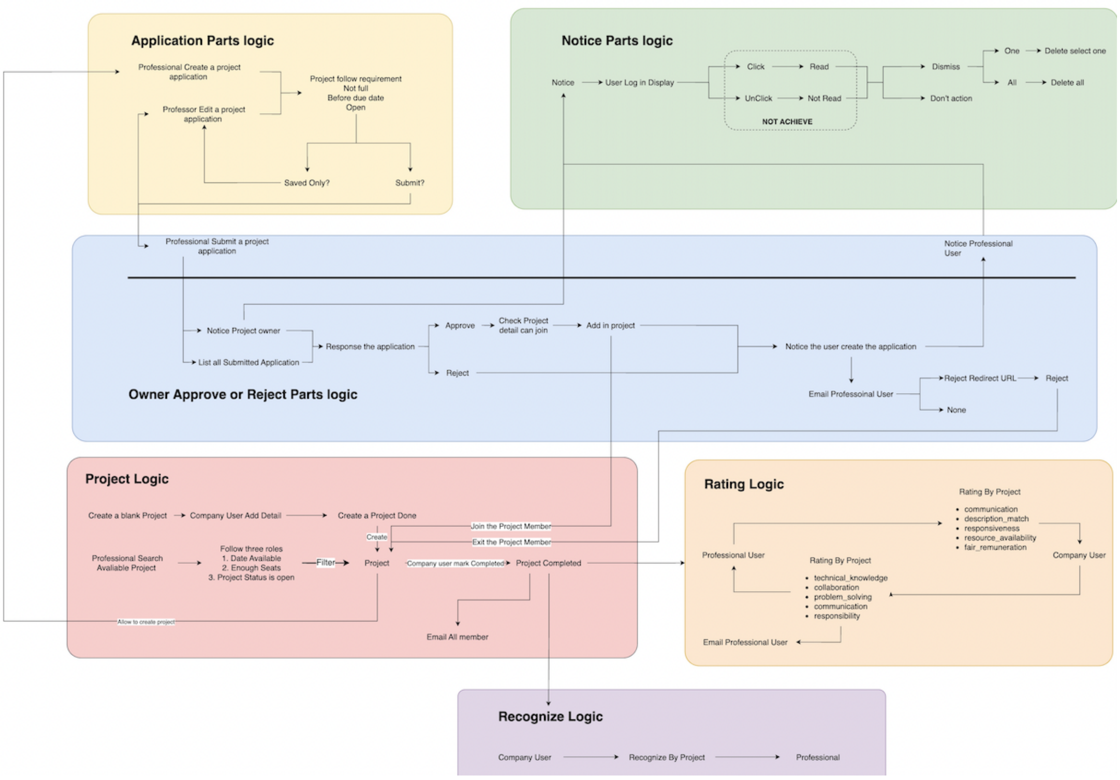


Figure 28: Logic Diagram

Register/Login/Logout

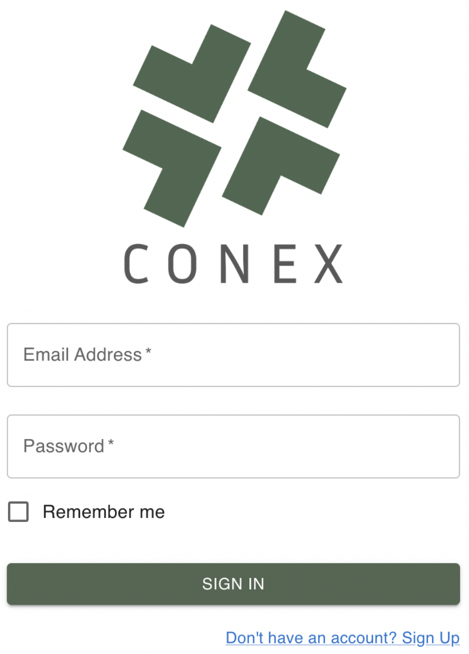


Figure 29: Sign in

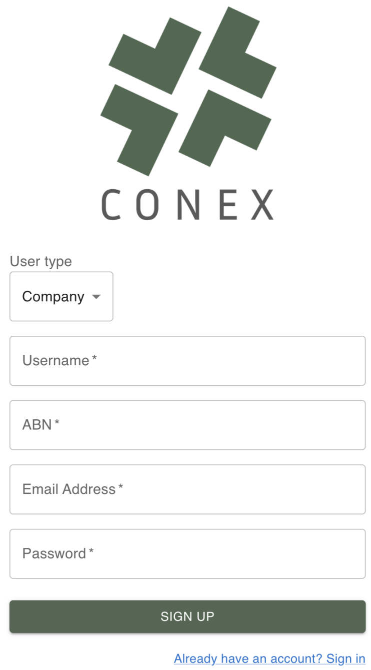


Figure 30: Registration

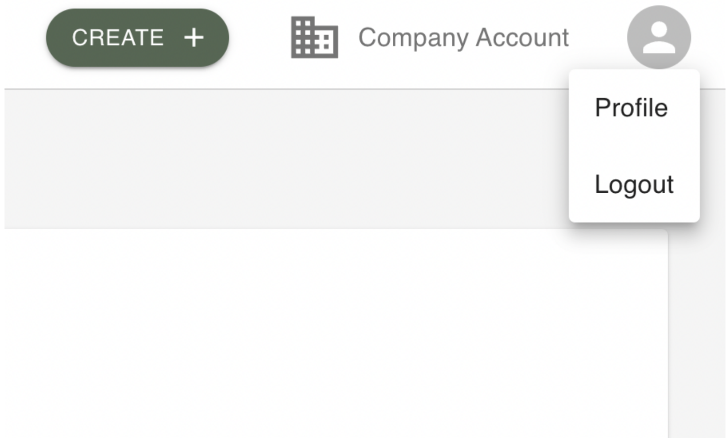


Figure 31: Top Bar Navigation

If you are new to the website, click 'Don't have an account? Sign Up' button in the bottom side of the screen as shown above to create an account (Figure 29). A new page will appear where you will input your details (Figure 30). The password must be unique. Once completed, press the 'Sign Up' button to create the account and it will prompt you to your newly logged in dashboard.

If you are an existing user, to log in, input your email and password and press 'Login' to be authenticated (Figure 29). To exit and logout of your account, press the profile widget in the top right corner and a selection will pop up (Figure 31). Press logout.

Dashboard/Filter

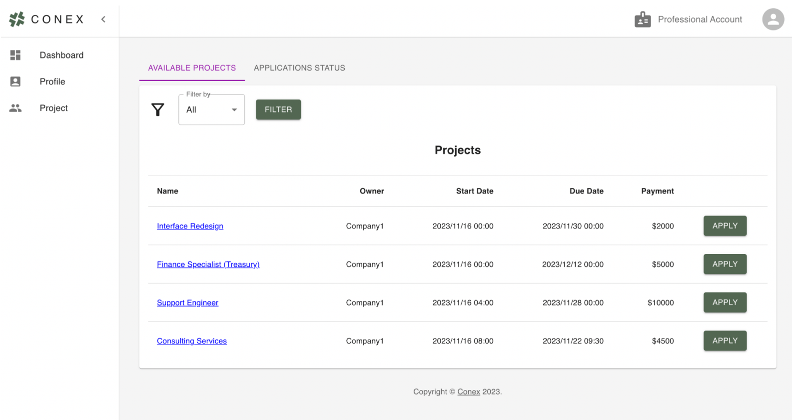


Figure 32 Dashboard

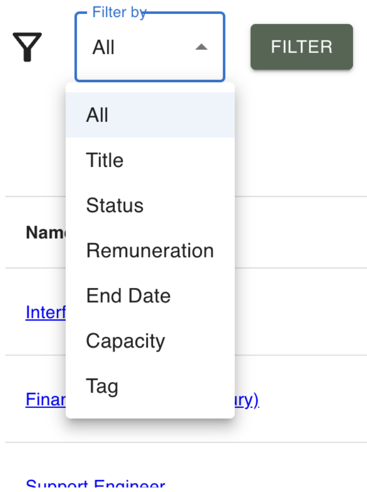


Figure 33 Filter Options

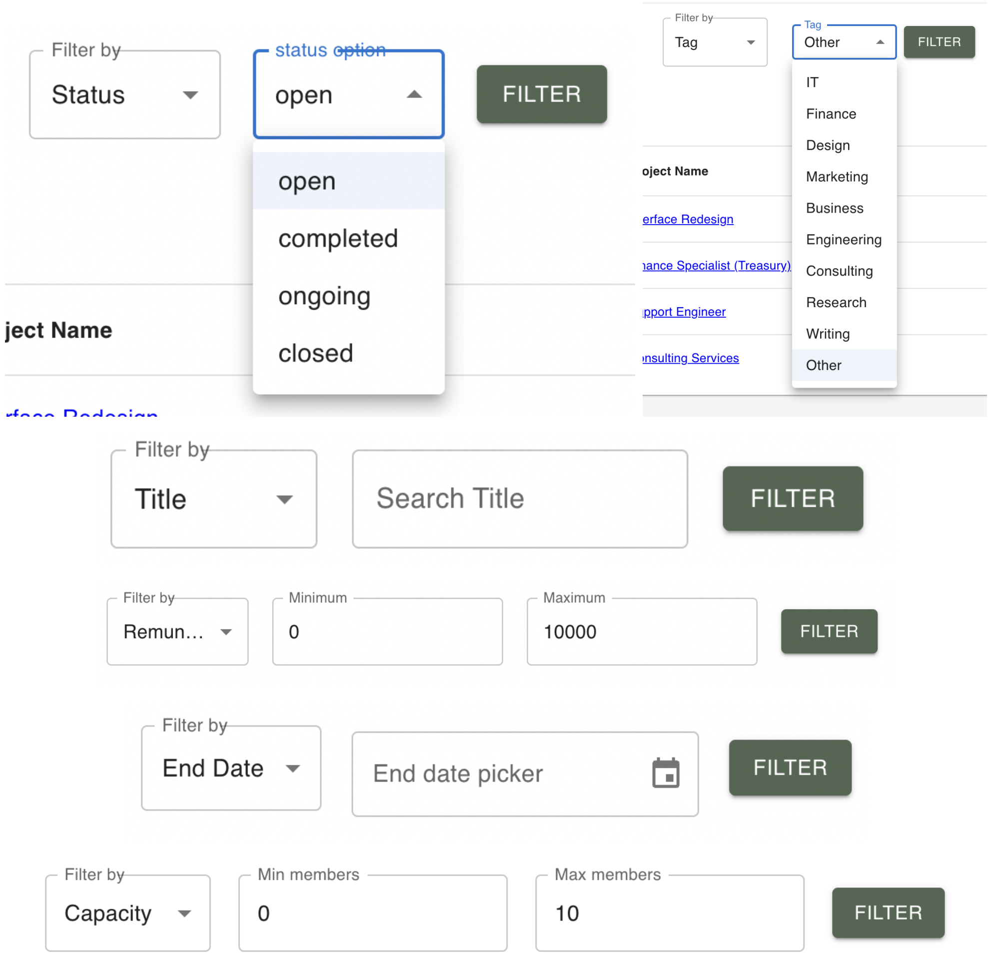


Figure 34: Filter Specifications

After login, you will be transported to the dashboard (Figure 32). As a professional user you will have all projects shown in and be allowed to filter through the list (Figure 33). As a company user you will have all the projects shown in your own in list at first, and another tab for all projects. In this filter within the dashboard we can navigate and seek certain parameters of the project you are looking for. This includes title, remuneration, end date, capacity, tags and status (Figure 34).

Project Creation

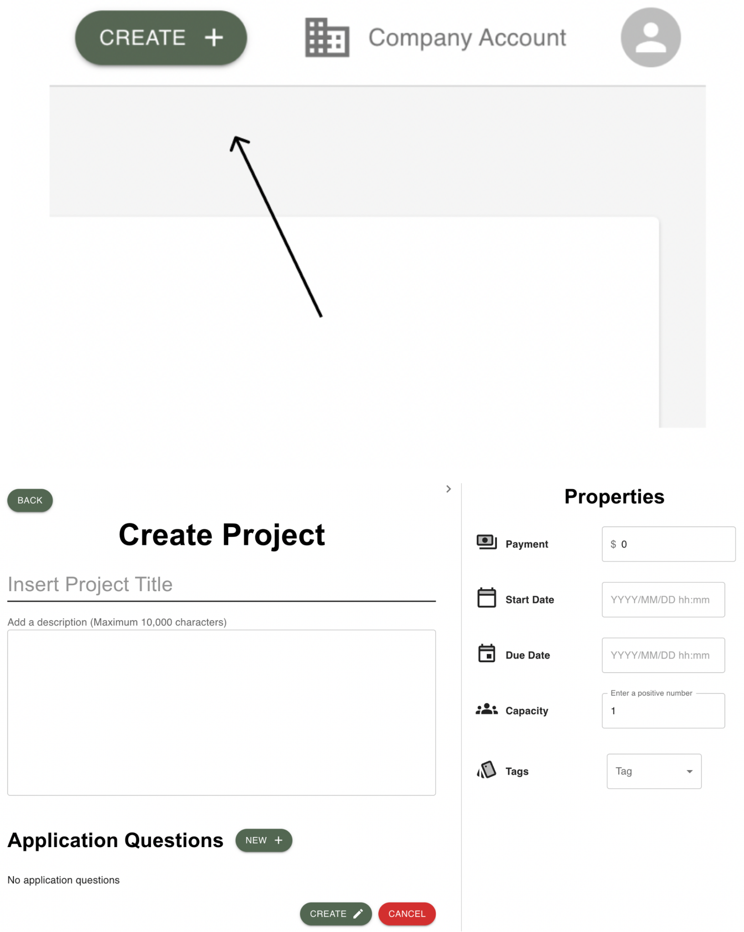


Figure 35: Project Creation

In order to create a project, a company user will have access to this feature within their top bar. This button will navigate you to a create project page in which you can enter properties and application questions for applying members (Figure 35). You will need to add project title, description, application questions, payment, start and end date, capacity and tags. After this is completed you can press create and the project will be live and open.

Project Details

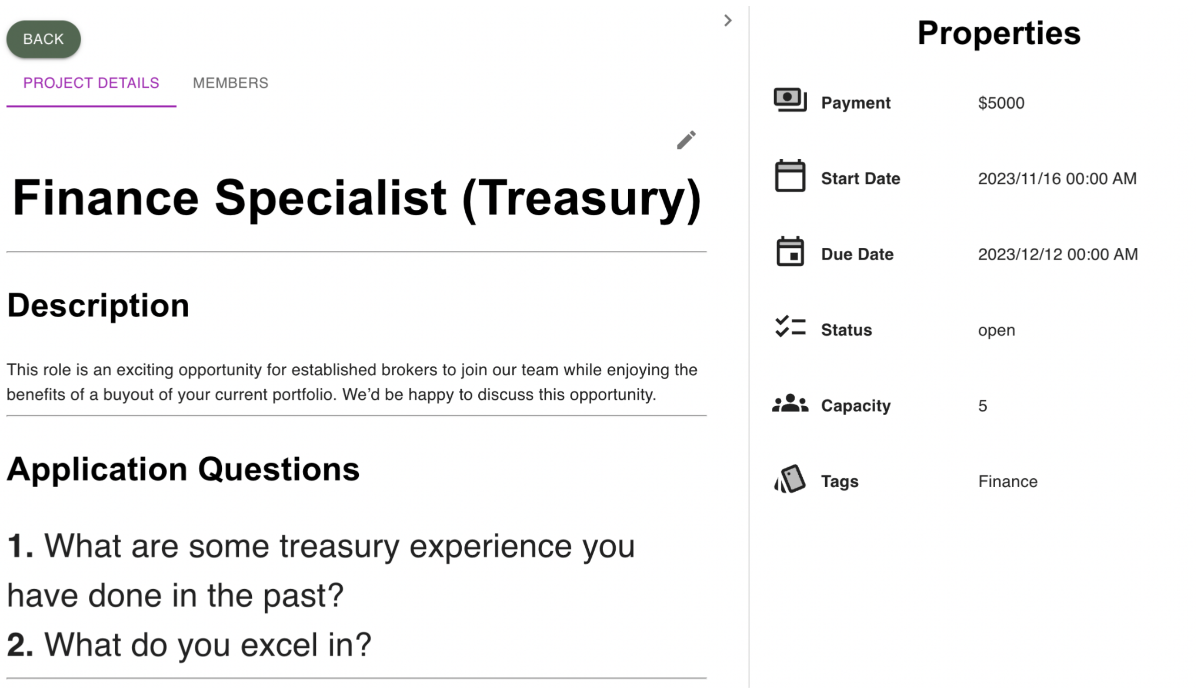


Figure 35: Project Details

Within the dashboard, users can navigate to the given projects that are available. By clicking on the title they will be transported to the project detail page in which the user can see the project in more detail (Figure 35). From this list users can read the following properties the project owner has listed and if they are willing to apply for this job.

Project Application

To apply for a project as a professional you can either apply from the dashboard by clicking 'Apply' or select the project (Figure 36). If you select the project you will be able to see more information about the project details, and can choose to apply in the project details by pressing 'Apply' (Figure 37).

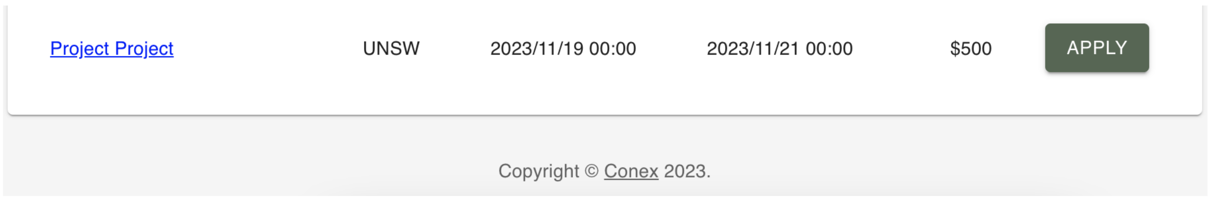


Figure 36 Applying to Project From Dashboard.

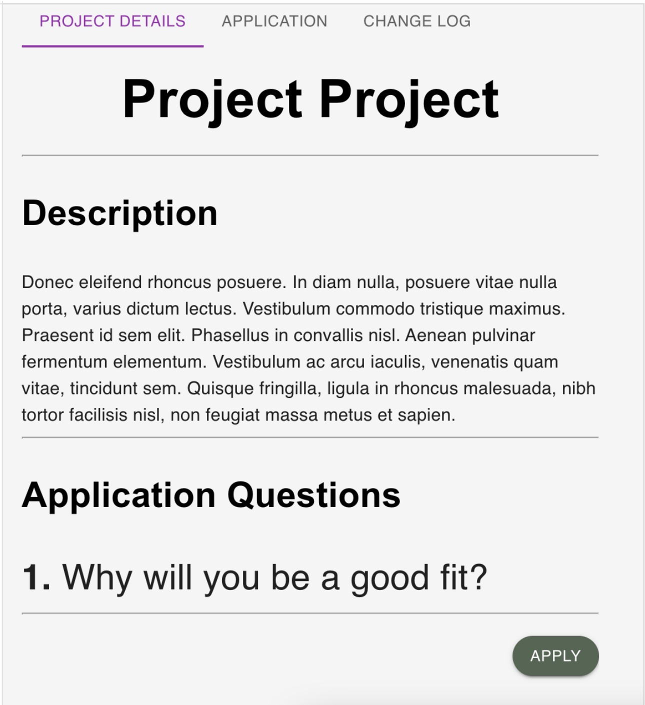


Figure 37 Applying from project details

Once you have clicked ‘Apply’, you will be taken to the project application where you will need to fill in any application questions and project details (Figure 38). You can also fill in details of a resume, by selecting ‘Edit Resume’ if you like to give the company more insight into your skillset (Figure 39). To finish the application you can press submit or you can choose to save or cancel your application.

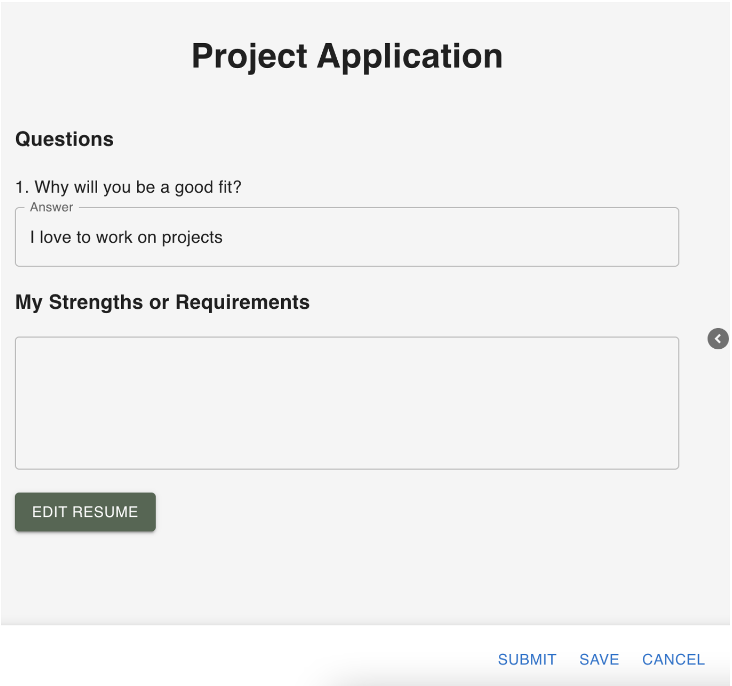


Figure 38 Project Application

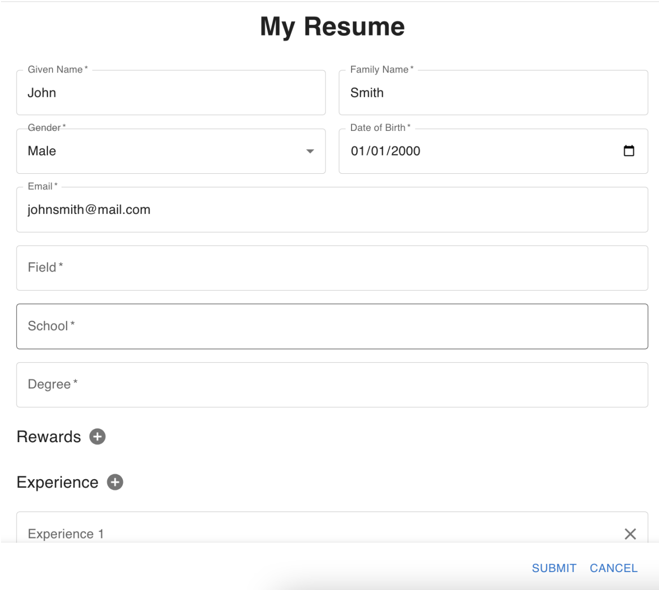


Figure 39: Resume details for project application.

If you choose to save you can retrieve your application under application status, in the dashboard, if you want to continue the application you can select view. In the application status you can also see the outcome of your application (accepted, rejected, pending, unperformed) (Figure 40).

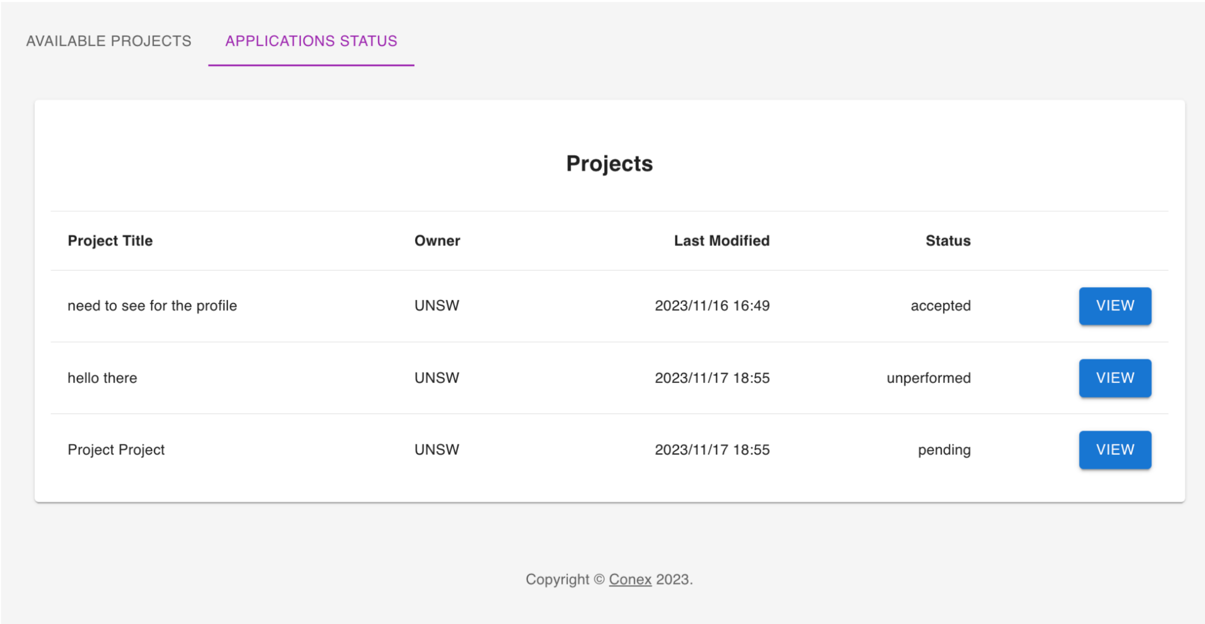


Figure 40 Application Status

You will be notified about the outcome of your application through email, where if you click on the link you will be able to choose if you would like to accept or reject the offer to join the application (Figure 41).

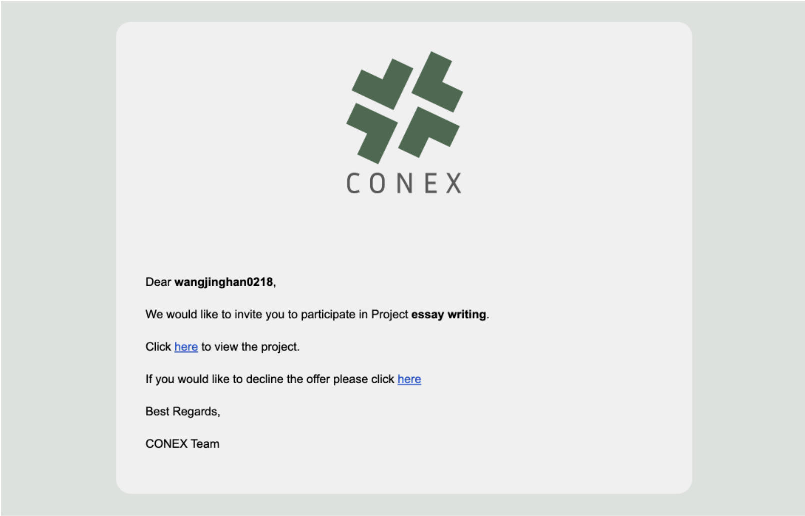


Figure 41: Email of Successful Application Outcome.

Certificates and Rating

Both company and professional users have the option to review each other upon completion of a project. Once the project has been completed the professional will be able to review the project in the project details sections (Figure 42). A Company can review a professional in the members tab of the project page (Figure 43). Upon clicking review a modal will appear, where you can select the number of stars to correspond to the assessee's performance on specific qualities and add a comment. When you click submit the rating and review will be visible on that user's profile (Figure 44).

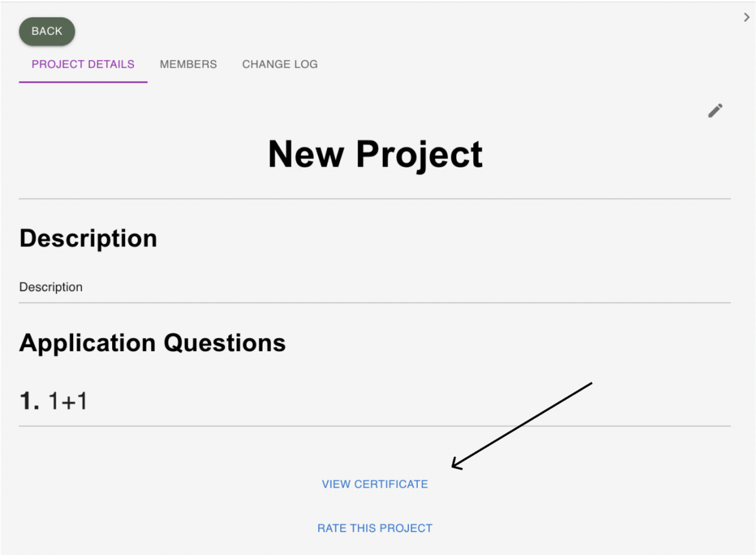


Figure 42: Professional Account View Certificate and Rate Project

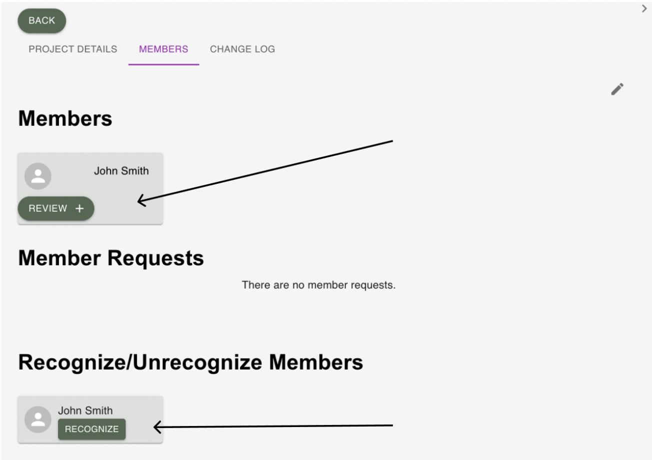


Figure 43: Company Review and Recognise Participant

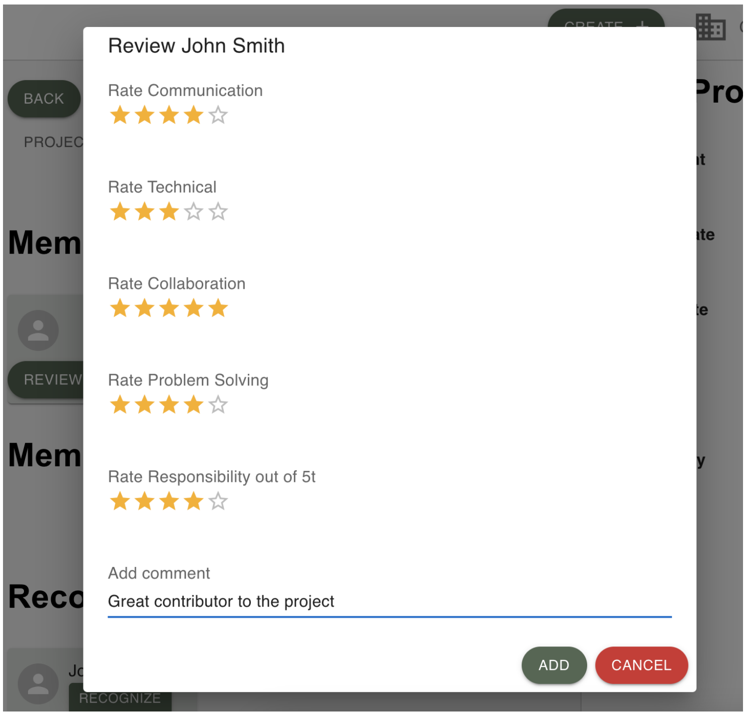


Figure 44: Review Modal

Figure 45 Certificate of Completion

As a professional and company user you will be able to see your review in their profile (Figure 46, 47). As a company user you can recognise a professional user in the members section for their contribution (Figure 43). As a professional you will be able to see this recognition in your profile (Figure 46). As a professional you will also be able to view your certificate, by clicking view certificate in project details section, which will lead you to your certificate (Figure 42, 45).

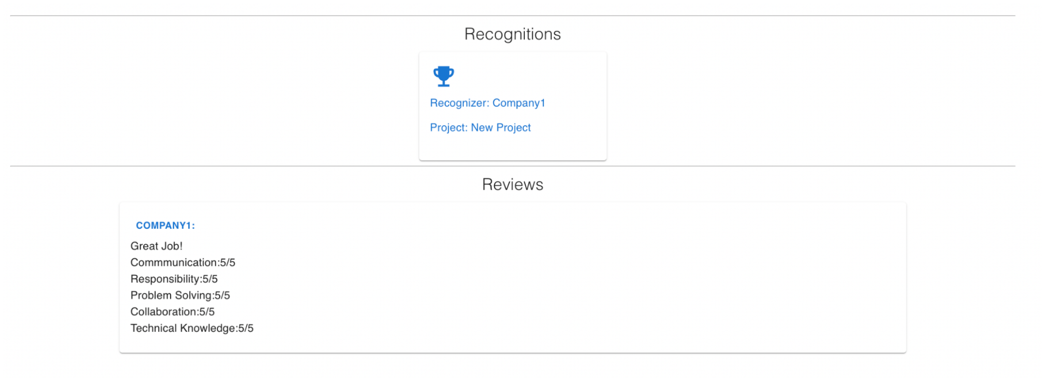


Figure 46: Professional Recognition and Reviews

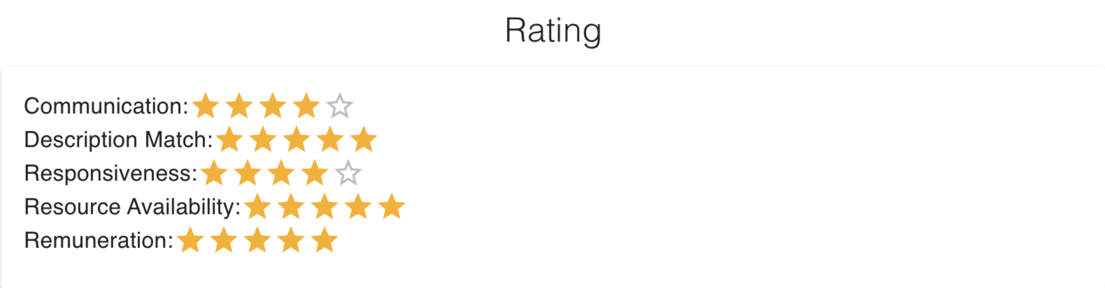


Figure 47: Company Ratings

Profile Page

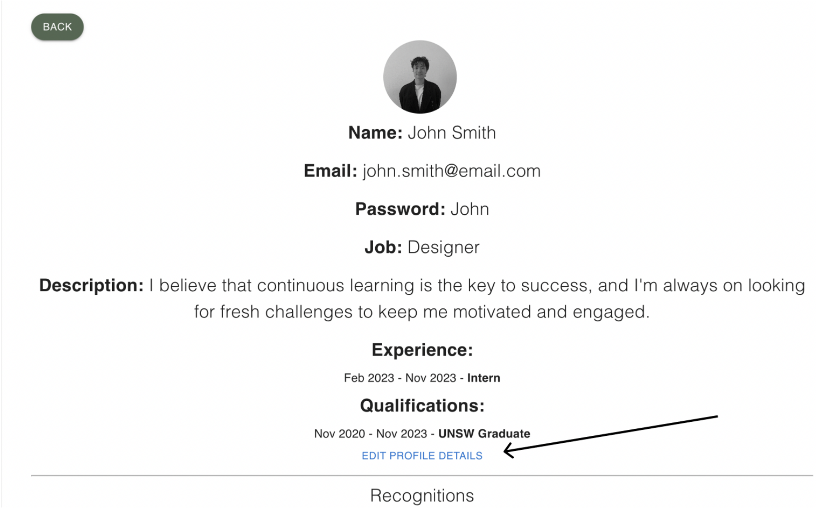


Figure 48: Professional User Profile

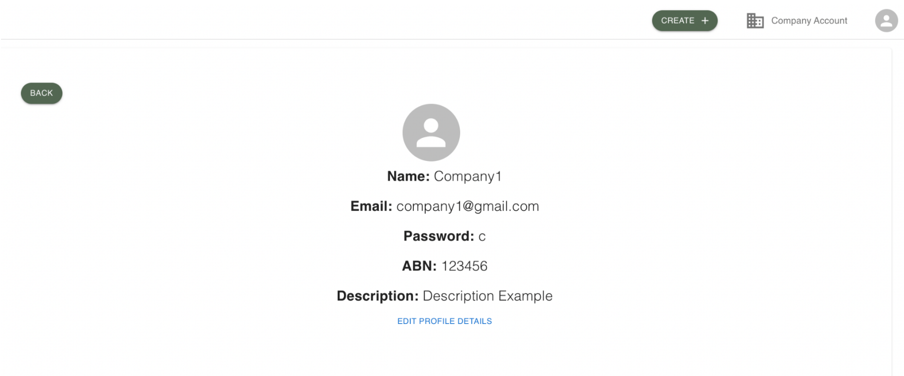


Figure 49: Company User Profile

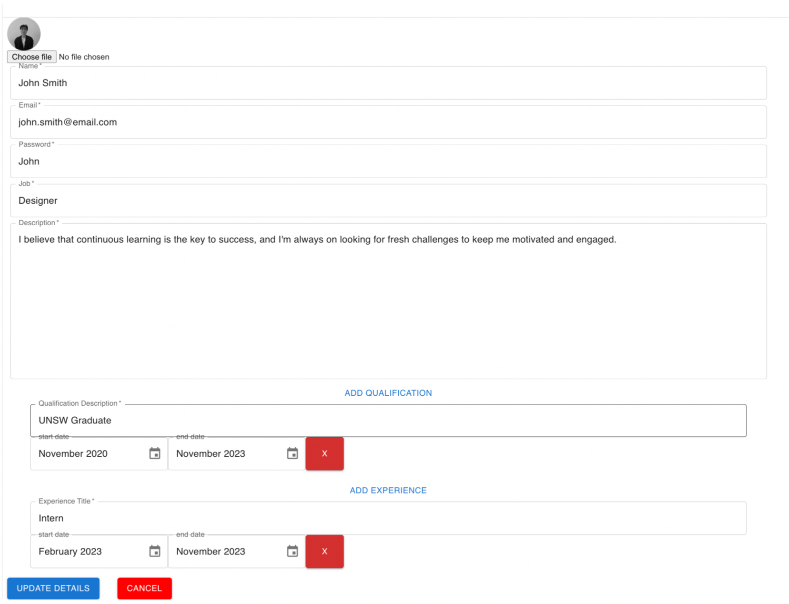


Figure 50: Edit Professional Profile

From the side bar on the left hand side, both company and professional users are able to view their profile details. This will include details from sign up and also recognition and reviews from previous projects completed (Figure 48, 49). By clicking the 'edit profile details' button, they will be navigated to the edit list for each property in their profile (Figure 50). For professional users this includes, profile picture, name, email, password, job, description, experience and qualifications. For company users they will be able to edit profile picture, name, ABN, email, password and description.

References

SQLAlchemy 2.0.21 (2023) Python. Available at: https://pypi.org/project/SQLAlchemy/ (Accessed: 15th November 2023).

2023. Python Logo. [Image] Available at: https://www.python.org/static/community_logos/python-logo-master-v3-TM-flattened.png(Accessed: 15th November 2023).

2023. HTML Logo. [Image] Available at: https://it.m.wikipedia.org/wiki/File:HTML5_logo_and_wordmark.svg (Accessed: 15th November 2023).

2023. FASTAPI Logo. [Image] Available at: https://seeklogo.com/vector-logo/425420/fastapi (Accessed: 15th November 2023).

2023. CSS Logo. [Image] Available at: https://commons.wikimedia.org/wiki/File:CSS3_logo.svg (Accessed: 15th November 2023).

2023. SQLAlchemy Logo. [Image] Available at: https://en.m.wikipedia.org/wiki/File:SQLAlchemy.svg (Accessed: 15th November 2023).

2023. Mui Logo. [Image] Available at: https://seeklogo.com/vector-logo/440988/mui (Accessed: 15th November 2023).

2023. ReactJS Logo. [Image] Available at: https://en.m.wikipedia.org/wiki/File:React-icon.svg (Accessed: 15th November 2023).

2023. Windows Logo. [Image] Available at: https://commons.wikimedia.org/wiki/File:Windows_logo_-_2012.png (Accessed: 15th November 2023).

2023. UNIX Logo. [Image] Available at: https://uz.m.wikipedia.org/wiki/Fayl:UNIX%C2%AE.png (Accessed: 15th November 2023).

2023. Sign in with app passwords Available at: https://support.google.com/accounts/answer/185833 (Accessed: 15th November 2023).

2023. MUI Licensing Available at: https://mui.com/x/introduction/licensing/ (Accessed: 16th November 2023).

2023. Python Licensing Available at: https://www.python.org/about/#:~:text=Python%20is%20developed%20under%20an,by%20the%20Python%20Software%20Foundation. (Accessed: 16th November 2023).

2023. The Python SQL Toolkit and Object Relational Mapper Available at: https://www.sqlalchemy.org/#:~:text=SQLAlchemy%20is%20the%20Python%20SQL,power%20and%20flexibility%20of%20SQL. (Accessed: 16th November 2023).

IBM (2023). JSON Web Token (JWT). [online] www.ibm.com. Available at: https://www.ibm.com/docs/en/cics-ts/6.1?topic=cics-json-web-token-jwt (Accessed 15 Nov. 2023).