



Capstone_MVP

An online portfolio to showcase capstone projects

Team 7:

Name	Role
Elsa Tse - htse460	Web & UX/UI Specialist, Front-end developer
Renrui Zhao - rzha180	Database Specialist, Back-end Developer
Jiajun Liu - jliu576	Full- stack Developer, Quality Assurance Manager
Jack Tian - ctia490	Full- stack Developer, Scrum Master
Fengning Yu - fyu855	AWS Specialist, Front-end developer

Executive Summary

Many computer science students are worried about their future work, some of them have enough abilities to complete a project but could not find an ideal job. Also many employers or clients are too busy to find these excellent students as their employees. In that case, a website which could show students' projects, allow students and clients to contact each other is needed for both students and clients. The success of similar platforms, such as LinkedIn, shows the potential impact of an online showcase website for the capstone course. LinkedIn has become a popular platform for professionals to present their work history and skills publicly. This has allowed individuals to stand out to potential employers and has led to more efficient hiring processes. The online showcase website for the capstone course can have a similar impact by providing a centralised platform for students to present their work and connect with potential employers or clients.

Our project-Capstone MVP allows students to work in teams of 5 – 6 people on a substantial project that aims to solve a real-world problem. Students could login and upload their projects to the public so that clients could find their most interesting one and contact that team or leave comments under the project just like a social media. students could also find others who have similar interests and work together for their future work. In addition, the admin could award badges to the winning projects which is not only an encouragement to students but also a chance to let more of the public view these excellent work.

As developers of this project, our team use C# and MySQL for backend part, JavaScript, HTML, CSS for frontend part. We are committed to developing a website that is visually appealing, easy to navigate, and provides all necessary information for students, employers, and clients. As developers of the online showcase website, our team is excited to bring this project to life and help bridge the gap between students and potential employers.

Table of Contents

Executive Summary.....	2
Table of Contents.....	3
Introduction.....	3
Aims and Objectives.....	3
Background.....	6
Project Implementation.....	20
Results & Evaluation.....	25
Future Work.....	27
Conclusion.....	29
References.....	30
Appendix.....	31

Introduction

Aims and Objectives

The vision of this project is to allow students to showcase their work to others; industry respective, staff, friends and family, the website as a platform to show people what they have accomplished at school so far such as linkedin. Moreover, students can connect with peers who share the same interests and skills and engage in professional networking activities. It provides a platform for job seekers to explore more employment opportunities, recruiters to identify potential candidates, and professionals to showcase their skills and expertise.

Target audience

The target audience are students who participate in compsci 399, students who also study computer science at university of auckland, lecturer and recruiters who look for potential candidates.

Scope

1. Student Project Gallery:
 - Develop a visually appealing gallery to showcase student projects.
 - Present student projects in an engaging and attractive manner.
2. Student Login and Project Details:
 - Implement a login system for students, enabling them to access the platform.
 - Provide students with the ability to specify and update their project details.
3. Visitor Engagement:
 - Incorporate a visitor login system, allowing them to interact with the projects.
 - Enable visitors to leave comments on projects and express their liking through likes or upvotes.
4. Teaching Team Functionality:
 - Enable the teaching team to log in to the platform with their credentials.
 - Provide the teaching team with the ability to award badges to the winning project(s).
 - Allow teaching team to delete student project

By achieving these objectives, the project will create a platform where students can showcase their projects, visitors can engage with the projects through comments

and likes, and the teaching team can recognize and reward outstanding work by awarding badges to winning projects.

Approach

Capstone-MVP is a full-stack project, 3 main tasks were splitted within the project; database, front-end and back-end. As 10 weeks time was given, we have set up a GitHub Repository to create a new repository on GitHub to host our project and initialise the repository with the necessary files and directories. A database schema was developed through SQL for storing student profiles, project details and other relevant information, also used to determine the relationships between. C# was used to implement backend functionality to handle user authentication, student profiles, project management, and database interactions. While HTML and CSS were used to create the structure and styling of the website for developing frontend interface. Our group has designed responsive and visually appealing web pages for different sections of the website and utilises JavaScript for dynamic behaviour and interactivity.

After developing the website task by task, we have integrated Backend with Frontend by connecting the backend functionality with the frontend interface. User authentication is implemented using the ASP.NET framework's built-in authentication libraries thought C#, such as ASP.NET Identity, which provides features for user registration, login, secure storage of credentials, session management, and authorization controls based on roles. Different roles have different accessibility, for example, admin has granted access to award and delete projects while students can upload their project and visitors can only like and commit. JavaScript was used in conjunction with Swagger to interact with APIs on the frontend, to allow projects to be uploaded in the website. Testing and debugging were applied at the end of the project to test the website's functionality, both frontend and backend, to ensure it works as expected. Moreover, our team has debugged any issues or errors encountered during the testing process.

Instead of relying on client meetings and progress meetings, we have adopted a proactive approach to maintain regular communication throughout this period, ensuring everyone involved stays informed and the project progresses smoothly. Through frequent calls and self-organised meetings at our school, we monitor the project's progress, addressing any emerging issues or bugs to keep everything on track. This proactive communication strategy allows us to merge our individual efforts efficiently

Outcome

The important outcome of our project is that we have successfully developed a fully functional university student portfolio showcase website with user registration, profile creation, portfolio management, and project showcase features. This has not only allowed students to showcase their projects, achievements, and skills to potential employers, peers, and the wider community and increased networking opportunities for students to connect with industry professionals, recruiters, and other students with similar interests. It also delivers a tool to assist the teaching team to better manage the capstone project and award them and allow others to provide feedback to the projects that leads to continuous improvement and a better user experience.

Background

Provide preliminary background information that puts your project in context. Education, particularly in the field of technology, has always placed high emphasis on real-world application and problem-solving skills. Students are often required to complete substantial projects that demonstrate their capability to leverage their learned skills to solve complex problems. Unfortunately, these computer science students have difficulty finding ideal job opportunities despite having the necessary abilities to complete projects. The current methods of showcasing these projects, usually through personal portfolios or project reports, often do not provide an optimal platform for exposure and collaboration.

Existing solutions: Several existing solutions like LinkedIn and GitHub do provide a platform for professionals and students to showcase their work, but they don't specifically focus on capstone projects and don't provide an easy and straightforward way to collaborate and form teams. Moreover, these platforms are not focused on helping students meet potential employers in a more personalised manner. For example, LinkedIn is a general platform for all types of professionals, and GitHub is more focused on code, lacking certain features that would allow for comprehensive project showcasing.

Value of the project: The Capstone-MVP lies in its ability to help students leverage their capstone projects for career progression. It provides them a platform to showcase their skills, find collaborators, and get exposure to potential employers, effectively bridging the gap between academia and industry. It can also provide a boost for employers, giving them a platform where they can easily find talented students with proven skills. By doing so, it offers several benefits:

1. Enhanced visibility:

Students can showcase their projects to a wider audience, including industry professionals, staff, peers, and potential employers. This increased visibility can lead to more opportunities for recognition and career advancement.

2. Networking opportunities:

The platform isn't just about showcasing projects; it's also a networking hub. Students can connect with peers who have similar interests and complementary skills, fostering a sense of community. This opens up potential for collaboration, mutual learning, and even the formation of teams for future projects.

3. Improved hiring processes:

For potential employers and clients, the platform serves as a valuable resource for talent acquisition. They can browse through the capstone projects to identify students whose skills and expertise match their requirements. This can streamline the hiring process significantly, as they can

get a comprehensive view of a candidate's capabilities through their project before even conducting an interview.

Methods and tools: The methods and tools that our solution, Capstone MVP, is based on reflect the complex, yet intuitive nature of the platform. These technologies have been carefully chosen to provide the best possible performance, security, and user experience:

1. Backend Development:

The backend of our platform is powered by C# and MySQL, two powerful and robust technologies for server-side development and data management. C# is known for its versatility and security, making it an ideal choice for developing complex web applications, while MySQL's efficient data storage and retrieval capabilities enable swift and seamless interaction with the platform.

2. Frontend Development:

The user interface is crafted with HTML, CSS, and JavaScript - the cornerstone technologies of web development. HTML is used for the structural framework of the platform, CSS is employed for the platform's visual aesthetics, and JavaScript enables the dynamic and interactive features of the platform, providing users with a seamless and engaging experience.

3. Web API[1]:

Communication between the frontend and backend is facilitated through a Web API developed with ASP.NET Core and Entity Framework Core. This allows for a seamless data exchange between the user interface and server, enabling efficient implementation of various functionalities of the platform.

4. Database Management:

The project employs SQLite Studio for managing the database. This includes the creation of the database schema and ensuring data integrity, which are crucial for the reliable functioning of the platform.

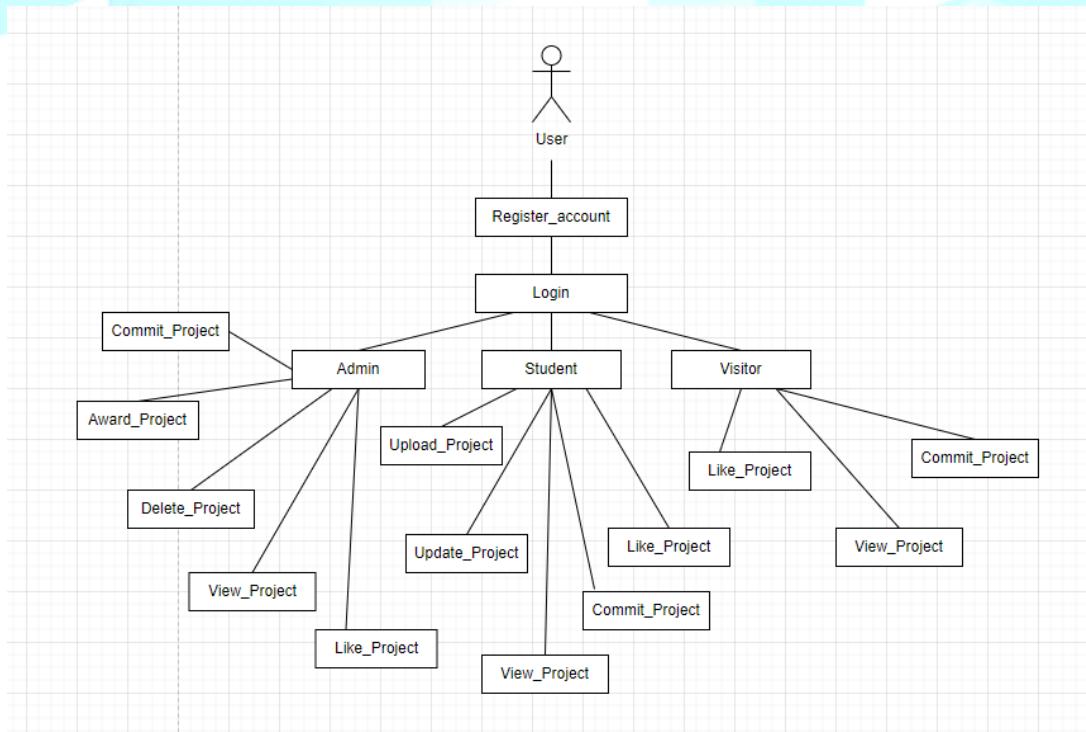
5. Testing and Debugging:

Testing and debugging are integral parts of our project lifecycle, and for this purpose, tools such as Swagger are used. Swagger helps test API endpoints and validate their expected behaviour, which is instrumental in ensuring that the platform operates as intended.

Specification & Design

Project Specification

Capstone-MVP provided a website for users to access the capstone project through admin, student and visitor.



All users:

- A homepage contains introduction and purpose of the website
- A register page for users to sign up for an account with their first name, last name, email address, password, interest and their register type.
- A login page for users to login to the account with their login information and selected type
- A project page to view the gallery of uploaded project and the awarded project
- A project detail page to view the detail of a specific project including team name, project name, semester, overview and more.
- A comment box for users to leave comments and like
- A logout function for users to logout their account

Student:

- A upload page for student to upload their own project
- A update page for student to update their own project
- A people page for student to search for their own teammates and join a team

- A create team page for student to create their own team with team name and team member name

Admin:

- A reward page for teaching team to award the project, including excellence award, community impact award and people's choice award
- A project page for teaching team to delete the unrelated project or repeat project
- A update page for teaching team to update some project with wrong information

Use cases

Use Case 1: Student Project Submission

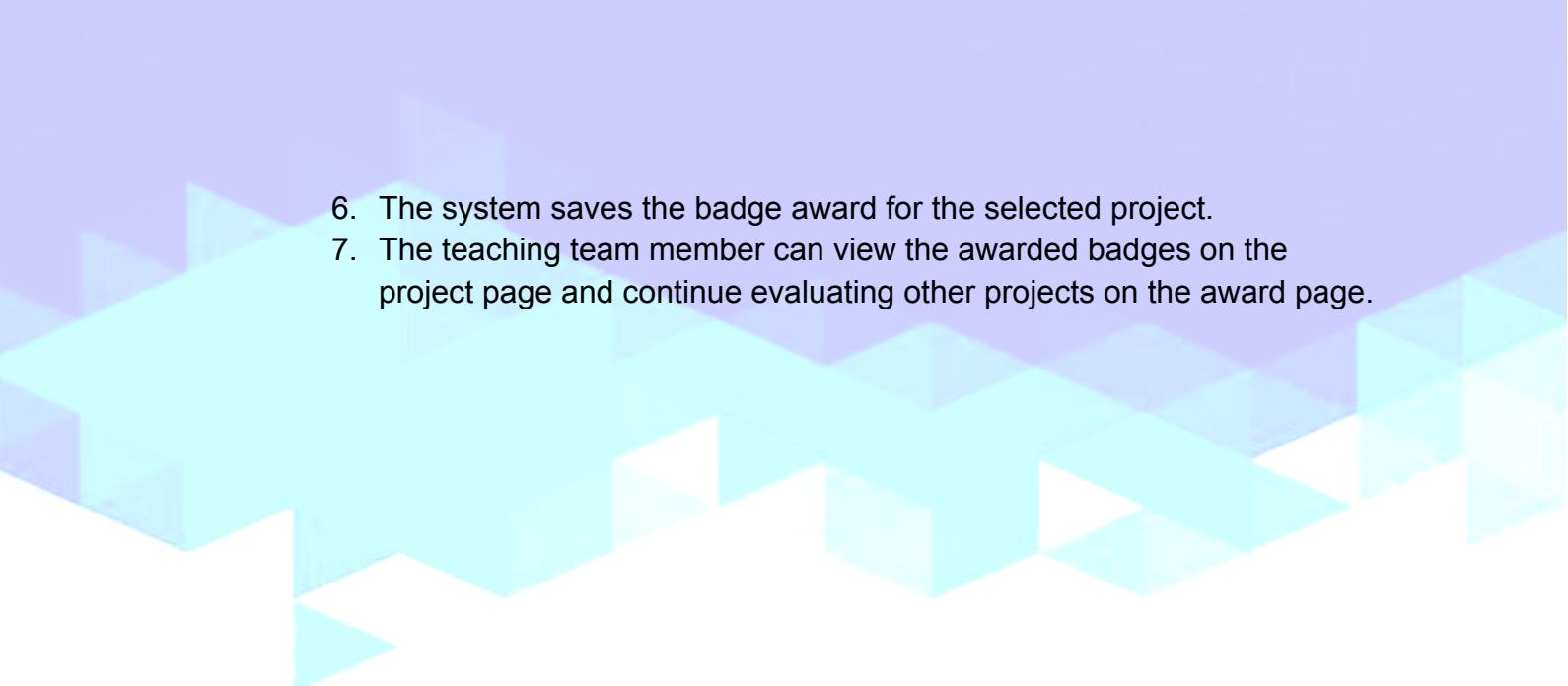
- User: Student
- Preconditions: The student is registered and logged into the system.
- Main Flow:
 1. The student accesses their account.
 2. The student selects the upload page to submit a project.
 3. The system prompts the student to provide project details, such as title, description, images, videos, and relevant links.
 4. The student fills in the required information and attaches any necessary files.
 5. The system validates the input and saves the project to the project page.
 6. The student views the project page to confirm their project has been successfully submitted.

Use Case 2: Visitor Interaction and Feedback

- User: Visitor (other students, faculty, industry professionals)
- Preconditions: The visitor is registered and logged into the system.
- Main Flow:
 1. The visitor explores the gallery to view various student projects.
 2. The visitor selects a project of interest.
 3. The system displays the project details, including images, descriptions, and other relevant information.
 4. The visitor reads the project details and examines the visuals.
 5. The visitor leaves comments, asks questions, or provides feedback on the project.
 6. The system validates the comment and adds it to the project's comment section.
 7. The visitor has the option to like the project to show their appreciation.

Use Case 3: Teaching Team Badge Awarding

- User: Teaching Team (instructors, mentors, judges)
- Preconditions: The teaching team member is registered and logged into the system.
- Main Flow:
 1. The teaching team member accesses their account.
 2. The teaching team members select the reward page to award a project.
 3. The system presents a list of projects available for award.
 4. The teaching team member evaluates the project based on predefined criteria.
 5. If the project meets the criteria for a badge award, the teaching team member selects the appropriate badge to assign to the project.

- 
6. The system saves the badge award for the selected project.
 7. The teaching team member can view the awarded badges on the project page and continue evaluating other projects on the award page.

Project Design

User interface

In terms of User interface design, the website was designed based on Nielsen's Heuristics and Gestalt's principles. Our website embodies an aesthetic and minimalist design approach, strategically crafted to captivate the user's attention while avoiding overwhelming visual elements. We create a user experience that is both visually appealing and user-friendly, allowing users to focus on the essential content and navigate through the interface.

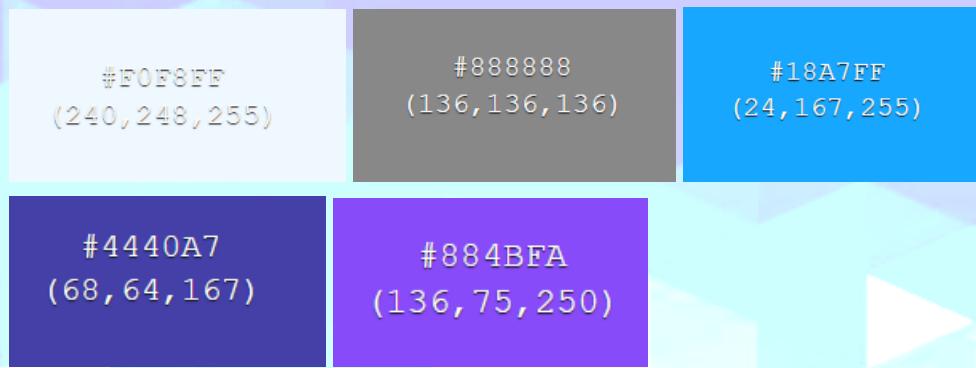
For visibility of system status, the page name on the top left corner will allow the user to access to any page at any stage. The login on the top right corner allows the user to login and after login, a user type will appear and a logout button once they click on the user type to logout of the account any time they want in order for them to feel in control.

For user control and freedom, it's easy for them to update their project once they have reckoned a mistake after uploading a project. Admin can also delete a project if they think the project is unrelated or repetitive.

To match between the system and the real world, we design the website with easily-understandable texts without specialised words. A logo of Google has been used on the register page which we are familiar with.

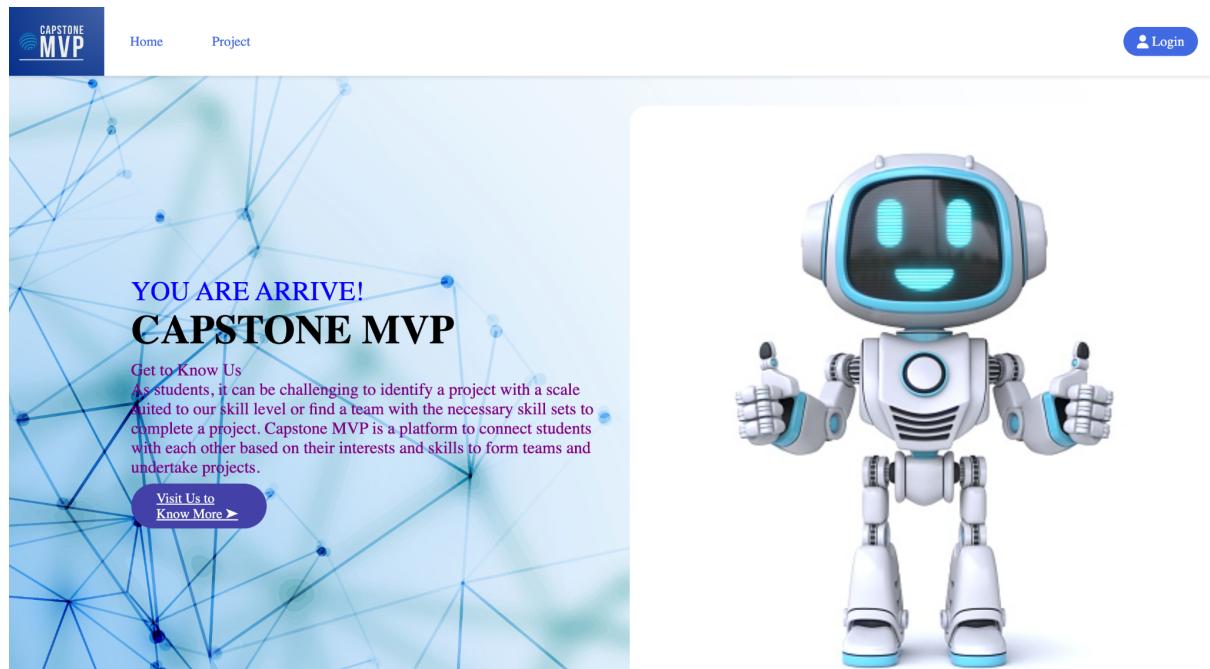
To help users to recognise, diagnose, and recover from Errors, when the error happens, the pop-up will come up to remind the occurrence of error. For example, when users input the wrong project id to delete the project, the pop-up will show "Not found" to remind the users to input the existing project id.

To ensure the consistency of our website, we make the colour, the front style, button, consistent throughout the wesbite. In this website, we adopt a blue colour scheme combined with a white background to showcase our project in a visually pleasing and professional manner (#f0f8ff, #888888, #18a7ff, #4440A7, #884BFA). The use of blue gives a sense of reliability and credibility in the project. The white colour complements the blue tones, creating a clean and uncluttered backdrop that allows the project content to stand out and grab the user's attention effectively.



In every web page that contains the input field. Every input field is aligned at the left side, which follows the common fate principle. Then, every button in the navigation bar is close to each other on a horizontal line. This provides a continuous presentation to users and follows the continuity principle. The login, user registration, project upload, project update, and project delete forms are shown in the centre of the website, and the space on the left and right of those forms are the same. This follows the balance and symmetry principle.

Home page



In the home page, when the user opens the website, there will be only two buttons, “Home” and “Project”, this is because, before the user login, they are only allowed to see the Project Gallery page.

Project Gallery

The screenshot shows a landing page for a project gallery. At the top, there's a navigation bar with 'Home' and 'Project' links, and a 'Login' button. Below the navigation is a large, stylized blue network graph. The main content area features two project cards side-by-side.

Project Card 1: Retype Spelling Game

- Introduction:** Describes the game as a keyboard-only ReType game where users type words to earn points. It highlights the use of arrow keys for correction.
- Outcomes & Conclusion:** Mentions a 10% increase in user satisfaction compared to other games. It also notes that the team learned about the importance of user feedback.
- Technologies Used:** Includes Node.js, Nginx, PHP, .NET, and Docker.
- System Architecture:** A diagram showing a client-side browser connected to a server-side Node.js application, which interacts with a database and various services.
- Future Work:** Plans to add more word categories and implement a mobile version.
- Team:** Shulei Jiang, Jason Si, Shiqing Duan, Yihang Li, Alvin So, and Seng Han.

Project Card 2: Peepers.js (Team 21)

- Introduction:** States the goal was to create a gaze tracking library for web developers.
- System Architecture:** A detailed diagram of a client-side application architecture involving a browser, a service worker, and a server.
- Technologies Used:** Includes Java, Python, C++, OpenGL, and OpenCV.
- Conclusion and Future Work:** Discusses challenges like cross-domain issues and future plans for real-time tracking and AI integration.
- Team:** Akash Arora, Vashon Reger, Theresa Lam, Kevin Ren, Jacky Chen, and Ricky Kafafong.

Then, in the project gallery page, there will be a search bar for users to search the project by typing key words, and even one letter. This is case insensitive. Below the search bar, there is a dropdown list that can be used to filter projects by different semesters.

Project Detail page

The screenshot shows a detailed view of the 'Retype Spelling Game' project. On the left, the project's main card is displayed, identical to the one in the gallery. To the right, there's a sidebar with a list of user comments and a comment form.

Comments:

- asdf asdowid
- diprdg_
- null awda
- Anonymous sfdf
- Anonymous asdfsf
- null sfdf
- sfdf sfdf
- asdfsf sfdf
- Anonymous sfdf
- Jack great

Comment Information Form:

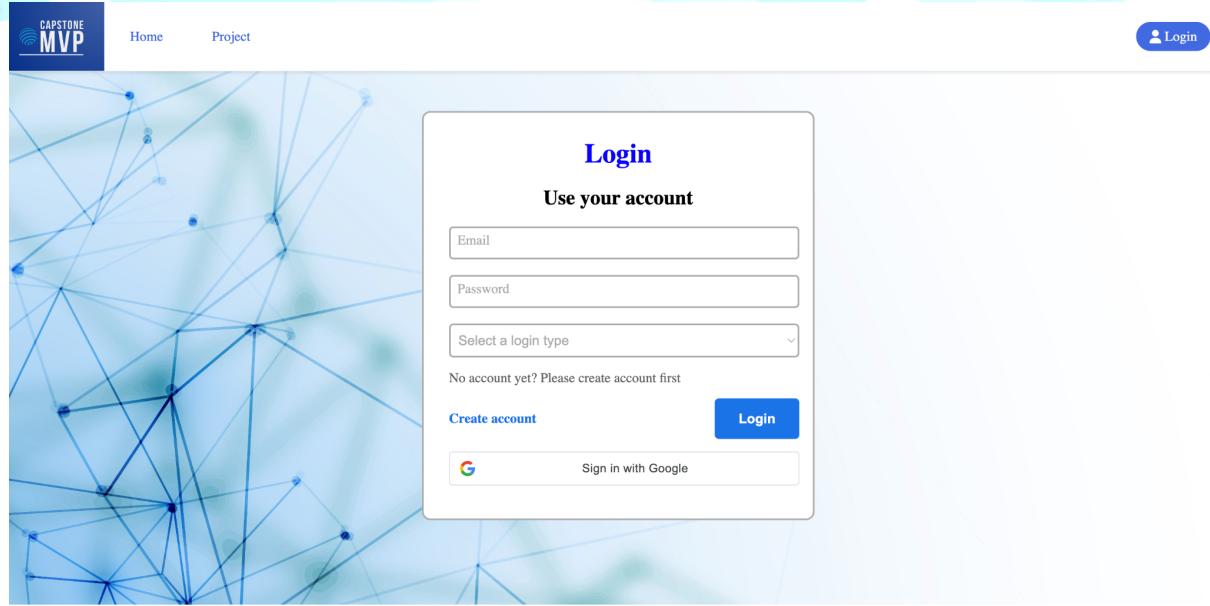
Please leave a comment.

Name:

CommentText:

When the user clicks any projects, the details of that project will be shown, and the user can like the project by clicking on that grey heart, and make a comment in the form called “Comment Information”. But only users that login with one of three login types can make comments. When the user leaves the comment without typing the name, the name of the comment will show “Anonymous”, otherwise, it will show the name typed.

Login page

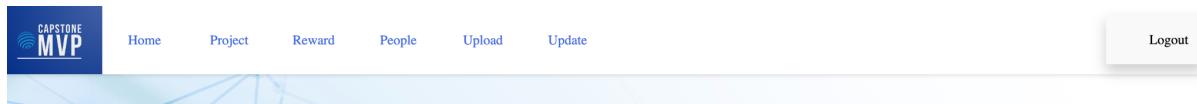


When the user clicks the “Login” button at the top right of the web page, they will enter into the login page. And, they need to type their email, password, and choose the login type to login. At the same time, the user can login with their Gmail account.

Registration Page

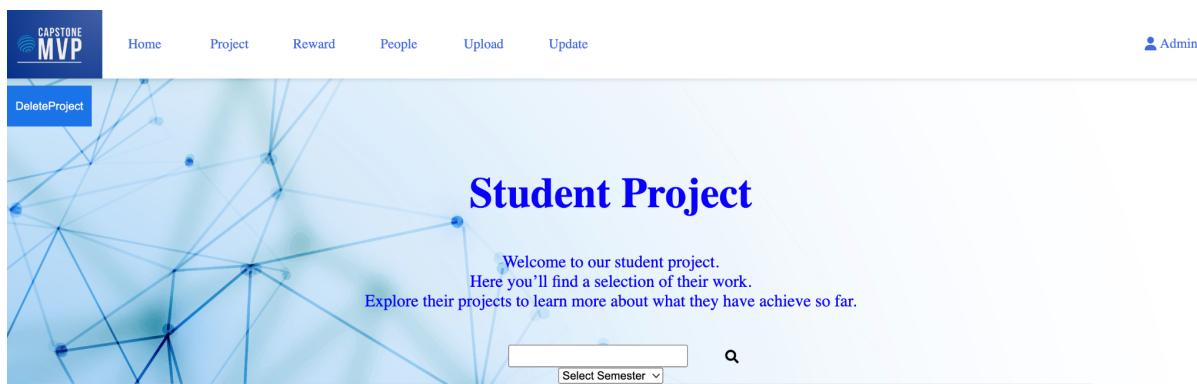
Create your Account	
Firstname	Lastname
Email	
Password	
Interest	
Select a register type	
<input type="checkbox"/> Show password	
Login instead	Register

When the user clicks on the “Create account” button on the “Login” form, the “Create your Account” form is shown to users to register their account by typing first name, last name, email, password, interest and selecting their register type. The invalid email and email that has already been registered will be identified and remind the users to input a new email. The user will go back to the Login page by clicking on “Login instead” or successfully registering their accounts. Furthermore, only one admin account will be registered with email, “admin@aucklanduni.ac.nz”, and the email with the ending as “@aucklanduni.ac.nz”, will be considered as a student email. Then, other valid email will be considered as a visitor email.



When the user logins, the website will go back to the home page, and the login type will show to replace the “Login” button, such as, “Admin”, “Student”, and “Visitor”. Then, when the user moves the mouse to that button, the “Logout” will show for the user to click on and logout.

The web page when the admin logins



When the admin logins, every button in the navigation will be displayed, and then “DeleteProject” in the project gallery page will be displayed as well. This is because there are some access rights that only the admin have, such as reward projects, delete projects.

After the user logins in one login type, the functionality for that login type will show to the user. When the visitor logs in, only the “Home” and “Project” in the navigation bar will be displayed. Then, when the students login, every button except for “Reward” in the navigation bar will be shown to students.

Update Project and Upload Project

Update Project

PROJECT INFORMATION ⓘ

ProjectID *
Enter your project ID...

Team name *
Enter your team name...

Project Name *
Enter your project name...

Semester *
Enter Semester...

Skill *
Enter your skill...

Project Description *
Enter your Project Description...

Approach *
Enter your approach...

Video *
Enter video link...

Upload Image

Click To Upload

Create Project

PROJECT INFORMATION ⓘ

Update Project

Only students can upload projects in the “Create Project” form and both students and the admin can update the projects. During uploading the project, when some input fields are not filled, the pop-up will remind the user to fill out them. Then, the students can upload the image for their projects by clicking on the blue box showing “Click to upload”. After the project is uploaded, the project will be shown on the “Project Gallery” page. In terms of updating projects, the admin or students just only need to input the project information they want to update.

People page

The screenshot shows two main sections on a web page.

Search for your teammates: This section contains a search bar labeled "Search" and a table with two columns: "Peoples" and "interest". The data is as follows:

Peoples	interest
Jack Zhao	AI
jack Zhao	AI
Roger Bob	AI
Roger Mack	AI
Roger James	AI
Roger James	AI
Roger James	AI
jack James	AI
jack James	AI
Mick Zhao	AI
Mick Wang	AI
string string	string

Join or Create a team that you are interested in: This section contains a search bar labeled "Search" and a table with three columns: "Team", "Teammembers", and actions ("Join" and "Delete"). The data is as follows:

Team	Teammembers	Join	Delete
Group1	Jack Zhao, Roger Bob, jack Zhao	Join	Delete
sdfgsdrg	sdf, df, Jack Zhao	Join	Delete

Below the tables, there is a message: "Did not find an interesting team? You can contact with admin to create your own team!" followed by a "Create Team" button.

Create New Team

TeamName _____

Student1 _____

Student2 _____

Student3 _____

Student4 _____

Student5 _____

Student6 _____

Create

When the user clicks on the “People” page, one box is to show students who login along with their interests. Students can find out some students with the same interest. The second box is to show the team for the capstone course. Students can join the team and only the admin can delete the team. There is a search bar in both boxes. Then, only the admin will be allowed to create the new team. When the admin clicks on the “Create Team” at the bottom of the people page. The “Create New Team” form will be shown to the admin.

Reward Page

The screenshot shows a reward page with a background image of a city skyline. A central white box contains the title "Rewarded Project" and a subtitle: "This is the rewarded Project Page. It's a great opportunity to help visitors understand the context and background of the best latest work. This allow the audience to see the people's favourite". Below this box, there are six award categories arranged in two rows of three:

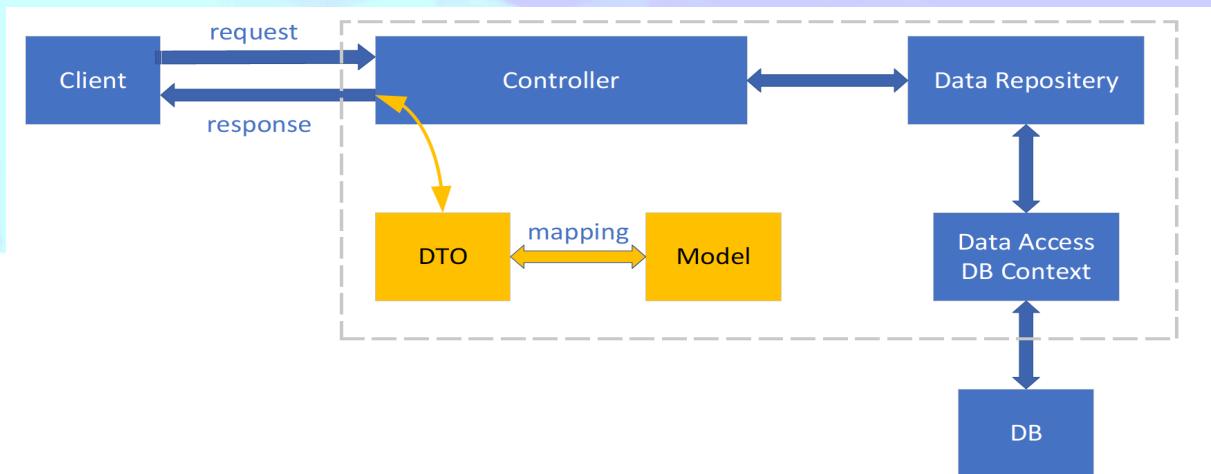
- Excellence Award Winner**: Shows a gear icon with "1st" and a dropdown menu labeled "Select a Project".
- Excellence Award Runners Up**: Shows a gear icon with "2nd" and a dropdown menu labeled "Select a Project".
- Community Impact Award Winner**: Shows a thumbs-up icon with "1st" and a dropdown menu labeled "Select a Project".

- Community Impact Award Runners Up**: Shows a gear icon with "2nd" and a dropdown menu labeled "Select a Project".
- Peoples Choice Award Winner**: Shows a heart icon with "1st" and a dropdown menu labeled "Select a Project".
- Peoples Choice Award Runners Up**: Shows a heart icon with "2nd" and a dropdown menu labeled "Select a Project".

A blue "Award" button is located at the bottom center of the page.

Only the admin is allowed to reward the projects. There are six awards. Below each award, there is a dropdowns list to show the project in the current semester. Since the current semester is semester 1 2023, only projects in this semester will be shown to the admin to be awarded. One award can not be given to more than one project in the current semester and one project can only be awarded with one award.

Project Implementation



The technology we use on the backend is ASP.NET Core, Entity Framework Core [2] that enables a database to work with .NET objects and eliminates the necessity for writing the majority of the data-access code that is normally required, and Web API. Then the software we use on the backend is Visual Studio, SQLite Studio, and Swagger.

We develop our project through the Web API. Firstly, we created a new ASP.NET Web API project on Visual Studio to initiate our project. We apply six components to develop our Web API, Data Model, Data Access, Data Repository, Data Transfer Object (DTO), Controller, and database. Firstly, we create the folder on the visual studio for the Data Model components to contain all the data objects in our projects, including Admin, Comment, Project, Student, Team, and Visitor. After that, we create a folder that contains the Data Access, and Data Repository components. And, we install the Entity Framework Core packages. Then, we generate the database, it can be shown in the SQLite Studio. For the Data Access component, we create a file to connect the database. For the Data Repository component, we create an interface file that contains multiple methods connected to the Data Access component file to access and manipulate the database, such as adding, deleting data in the database, or searching for specific data in the database. Then, there is also a file to implement the methods in the interface file. After that, we create a Data Transfer Object (DTO) folder that contains the InputDto, and OutputDto files. DTO is the object that carries information between the client and the system, and we use DTO on the Comment, Project, Team, Student, and User data objects. For example, when the students upload or update their projects, they just need to input some of the attributes in the project data object by input DTO, and when the project is showcased on the website, not every attribute will be shown by output DTO.

Finally, the Controller component is used to create some method to handle the

request from the client and make responses. We create the user, project, comment, and team controller classes to execute the related POST, GET, PUT, and DELETE methods. Then, each method will be considered as an API endpoint that refers to a URL. And, each URL can be fetched to the front end to satisfy the client's request. When some methods in the controller classes need to access the database, the methods in the data repository component can support it. Then, we use Swagger to test the methods in each API endpoint and make sure they work as expected.

When it comes to the authentication and authorization, we create a folder to contain a file as an authentication handler, and when three types of users, admin, students, and visitors, log in successfully, they will be issued an authentication ticket that includes the claim principle, claim identity, and one or more claims. Then, we register the authentication handler and the authorization policy with the service container in a file called program to operate the authentication and authorization. Then, when some methods in API endpoints in the controller component need authorization, it will show the authorization policy, like "Student Only" or "Admin Only" to show only students or admin have the right access to those methods. After that, the claims in the authentication ticket issued to clients after they are authenticated will be used by the authorization mechanism to check whether the client will be granted access permission to API endpoints that need the authorization.

There are all the methods in the controller component.

Project method

Add project: It is a POST method to get some attributes of project data objects from the student by inputDto to upload the project, including team name, project name, skill, project description, approach, video, and image. Then, it will add inputs about the project into the project database table.

List project: It is a GET method to get all the project data in the project database table. The data about projects uploaded and their detail will be shown on the "Project Gallery" and "Project Detail" pages

Get project by project name: it is a GET method to get the project by input. The related project data that contain the name inputted will be found in the project database table. It is case-insensitive, and it will be used to execute the functionality of the search bar on the "Project Gallery" page in the front end.

Get project by semester: It is a GET method to execute the functionality for the semester dropdown list on the project gallery page to filter out projects in the project database table by semester.

Get project by project ID: It is a GET method to get the project data from the database by project id to support the functionality of deleting projects and updating projects on the front end.

Upload images: It is a POST method to support the functionality of uploading images on the front end. We create a folder called “img” in the visual studio to store all the project images uploaded with their project name.

Get project photo by project name: This is a GET method to get the related image from the ‘img’ folder by the project name inputted. This supports different types of image files. This will be used to execute the functionality to show the related photo for the project on the project detail page.

Update project by project id: This is a PUT method to enable both students and the admin to update the project information in the project database table by typing a specific project id. “Not found” will show to users when the project with the typed project id is not in the project database table. Otherwise, “Great! You have successfully updated the project!” will show.

Delete Project: It is a DELETE method to only allow the admin to delete the project shown on the “Project Gallery” page from the team database table. It applies the authorization policy and authentication.

Reward project: It is a POST method to reward the project. This method will create the responses to users according to success status by accessing the database to check the award record.

Get the number of likes on the project by project id: It is a GET method to get the number of likes for the project by project id from the project database table.

Update the number of likes: It is a POST method to update the number of likes on the project database table when the users click on the “heart” to like the project.

User Method

Register: Three register methods are POST methods to add the admin, students, and visitors' registration information, like, first name, last name, email, password, and interest into the database. This method will create the responses to users according to the success status. If the account is not successfully registered, the related reason will be shown.

Login: Three login methods are GET methods, which will apply the authorization policy and authentication to check whether the email matches the passport and the user is registered. In the front end, the btoa method will be used for authentication.

Team Method

List Student: It is a GET Method to show all the students that login into the people table on the “People” page through output DTO to just show the first name, last name, and interests from the student database table.

Get student by name: It is a GET method to execute the functionality of the search bar to come out with search results that contain the keyword typed from the student database table. This also applies output DTO to hide the email and password.

Get team by name: It is a GET method to execute the functionality of the search bar to come out with search results that include the keywords typed from the team database table.

List Teams: It is a GET method to show all the information about the team on the “Team” table on the “People” page from the database including, the team name and name of six students.

Add Team: It is a POST method to execute the functionality of creating the new team by the admin, and only the team name is needed to be inputted. The team data will be added to the team database table.

Join Team: It is a POST method to enable only students that log in to join the team that has an empty place. The responses will be created to inform students whether they successfully join the team or the team is full. This method applies the authorization policy and authentication. When the student successfully joins the team, the combination of first and last names will be shown in the team database table.

Delete Team: It is a DELETE method to only allow the admin to delete the team from the team database table. It applies authorization policy and authentication.

Comment Method

Get Comments by Project ID: It is a GET method to get all the comments for the project by its project id from the comment database table. The project id is an attribute in the comment data object. When the project is shown on the “Project Gallery ” page, its project id will be used to fetch this method to get all the comments with this project id.

Write Comment: It is a POST method to allow students, visitors, and the admin to leave comments for the project. This applies the input DTO to only get the name and comment text as input, not all the attributes in the comment database table.

Results & Evaluation

Project goals: In our website, there are three types of users, one is a student who is a student in UoA, one is admin who is in the teaching team, the last one is user, who is anyone outside of UoA. Each user can look at all the projects that students uploaded. Students can login and upload their new projects or update the details of their uploaded projects. Any user who logged in can leave comments under the details of projects and like the projects. The teaching team can delete projects, create new teams and award projects. Therefore, We have satisfied all the four fundamental goals in our website. Those are A gallery showcasing student projects presented in a visually appealing manner, students can login and specify their project's details, visitors can login, leave comments and like the projects, the teaching team can login and award badges to the winning projects. Based on these, we have satisfied some external goals: students can join a team, teaching teams can create new teams and delete teams.

Testing

In order to detect possible issues and make the final testing be easier and quicker. We tested our code at every stage. After we create each API in the backend, we will test the API to make sure for specific input, there is always a specific result that will come out. Then after we created all the APIs that we need in our project, we will run the entire backend and test all the APIs to figure out are there any errors to run the backend and do the APIs produce the expected results. This is how we test the backend. To test the frontend, we also tested each function when we add this function. When we implement an API, we will enter the input and see what the result that API will respond to. After we added one more function in Javascript, we will test the function to see if it works as what we expected. Our testing and programming is on the same track. After we finish the programming, we will test all the functions in our project to see if there are any errors in our project and all the functions work as expected. After that we improved our project, try to add more functions that we needed. Then test all the added functions. Before the deadline, we tested all the functions again to make sure the final version of the project worked as expected. By using this test method, we can find the errors in time. For the API that we created for list all the projects, we find out the origin API will output all the informations we stored in database even there is no values, it will output null, but we only need the informations that have values, we find this error in time, it help us to save time when we implement this API to frontend. When we implemented the register API in the frontend, we found that after we input the values, there was no change in the database, then we knew that there were some errors when we implemented the API. It makes our final test easier and faster.

The tool we used to test the backend is Swagger. The Swagger will automatically open when we run the project. In our backend, there are 29 APIs used in our projects, 10 APIs for POST method, 15 APIs for GET method, 3 APIs for DELETE

method, 1 API for PUT method. All the APIs returned a status code of 200 if the parameter is valid, otherwise, it returned a status code of 400.

Our code passed all the tests and the results of all the functions are what we expect. And they met the requirements of what the client asked.

Strength:

- Each API works individually, and they are easy to modify.
- The name of each function and variable is quite simple and related to their function, it is easy to know the usefulness of each function and variable when doing the testing.
- The UI design of the website is simple, the user is easy to understand how to use the website.

Weakness:

- After awarding the projects, the icon for the award can not come out immediately, the user has to refresh the website to display the icons.
- As more projects are uploaded, the website might be slower when displaying the projects, because we output all projects at the same time.

Future Work

While our website successfully fulfils the fundamental requirements outlined by our clients, we acknowledge that, due to time constraints and the extensive scope of the project, certain additional features could not be implemented at this stage. That said, we have identified several enhancements that can add significant value to the platform in future iterations:

- **Award Management:** Allow teaching teams to update awards, but restrict modifications once the semester has concluded.
- **User Profiles:** Provide each student and visitor with a personalised profile, allowing them to introduce themselves and list their skills.
- **Project Preferences:** Enable students to specify their top 'n' project choices, fostering better alignment with their interests and expertise.
- **User Authentication:** Implement Google login functionality for easier and faster user access.

Based on these features, they can also add the following features:

- **Image Storage:** Rather than storing uploaded images in a folder, they should be saved directly into the database for easier and more secure data management.
- **User Type Recognition:** Automate user type recognition upon login, so the website can identify if the user is a student, visitor, or admin. This would obviate the need for users to select their login type manually.
- **Team Collaboration:** Implement a feature that allows students to form teams within the platform, enhancing collaboration on shared projects.

In addition, some features can improve the user experience in the website:

- **UI Design:** Enhance the overall aesthetic appeal and usability of the website with a more modern and attractive UI design.
- **Liked Projects:** Ensure that users' liked projects persist and are displayed in subsequent sessions, providing a more personalised browsing experience.
- **Skill-Matching Algorithm:** Develop an algorithm to match students and projects based on skills, interests, and career goals. This could greatly improve the relevancy of projects suggested to students and help employers find suitable candidates more efficiently.

- **Project Reviews and Ratings:** Allow users to review and rate projects. This will not only provide valuable feedback to the students but also help other users in selecting projects of interest.

Conclusion

The aim of this project was to develop an online portfolio to showcase capstone projects in order to provide more chances to show their professional skills to the public by working in teams of 5 – 6 people on a substantial project that aims to solve a real-world problem and post their project in this project website. In that case, Capstone MVP would help students to find their prospective employers, clients, which could provide them a better job opportunity.

Throughout this project, Jack and Roger did a great job on developing back end functions so that the rest of team members could use the url to develop their front end job. In addition, they also helped others about the front end job so that the project could be completed in time. Elsa pays lots of attention to optimise the website by setting parameters in the CSS file so that the website seems more professional and reliable for others. Fenning and Renrui also made an effort to develop different pages in the website, providing enough functions for users.

Similarly, despite its initial difficulties, the choice of an agile based project management style was highly effective for the team. Through weekly meetings, large segments of the codebase were able to be completed, while having in depth discussions about the current project status as well as what needed to be prioritised next. Working on the project for long periods of time in person with the whole team helped facilitate a highly supportive environment, which led to the creation of a successful product.

This project has many significant outcomes, Jack, Roger and Renrui are currently deployed using AWS which is available for anyone to sign up and start creating projects and finding teammates. Capstone MVP is substantial platform that clearly solves the issue of missing great abilities students because of busy on their business for clients; provide a platform for students could work with others to show their projects to others, which could receive identify with public, meet other students who have similar interest as well. By making it easier for them to find projects, there is more time and opportunity to learn and become better at their chosen field - a win for both students and employers alike.

References

- [1] Microsoft. (2023). Tutorial: Create a web API with ASP.NET Core. Retrieved from:
<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-7.0&tabs=visual-studio>
- [2] Microsoft. (2021). Entity Framework Core. Retrieved from:
<https://learn.microsoft.com/en-us/ef/core/>

Appendix

Gantt Chart



Capstone-MVP Project Github Repository:

<https://github.com/uaa-compsci399-s1-2023/project-team-7>

Capstone-MVP Project Presentation video link:

<https://www.youtube.com/watch?v=CrFY0aRDWdY>

Declaration of Authorship

Section	Author(s)	Signature(s)
Executive Summary	Renrui Zhao	Renrui Zhao
Introduction	Elsa Tse	Elsa Tse
Background	Fengning Yu	Fengning Yu
Project Specification	Elsa Tse	Elsa Tse
Project Design	Jiajun Liu	Jiajun Liu
Project Implementation	Jiajun Liu	Jiajun Liu
Result & Evaluation	Jack Tian	Jack Tian
Future Work	Jack Tian and Fengning Yu	Jack Tian Fengning Yu
Conclusion	Renrui Zhao	Renrui Zhao