



Final Deliverable – Software Project



Authors:

Wei Shan Sim
Aishwarya Singh Tomer
Muhammad Iatisam Nawaz
Kar Hock Teh

Team 62 (Tutor Group 7)

Table of Content

Headings	Page
<u>Background</u>	2
<u>Planning and Research</u>	3
<u>Prototyping and Iteration</u>	11
<u>Design</u>	24
<u>System Development</u>	41
<u>Analysis</u>	66
<u>Evaluation</u>	68
<u>User Guide</u>	73
<u>Conclusion</u>	75
<u>Individual Reflection</u>	76
<u>Reference</u>	77

Background

The market for portfolio platforms has grown significantly in recent years, driven by the increasing demand for online portfolios as a means of showcasing work and skills. According to a report by MarketsandMarkets, the portfolio platform market is expected to grow from \$1.5 billion in 2020 to \$3.5 billion by 2025, at a compound annual growth rate (CAGR) of 18.6%.^[1]

Our project aims to meet the needs of freelancers, creative professionals, students, and educators by providing a platform that allows them to easily create and customize professional portfolios. We recognize that individuals in these fields often face challenges in promoting themselves and their work, and that existing portfolio platforms may not always meet their specific needs.

By creating a platform that is tailored to the needs of our target audience, we believe that we can help them to more effectively showcase their work and connect with others in their industry. Our platform will allow users to create profiles that showcase their professional portfolio, skills, and interests. Users will have the option to set the visibility of their profile sections and individual section items to either public or private. The website will provide a search function that allows users to browse and view other users' profiles, using filters such as profession and location to narrow down results. Users can choose to publish their profile at any time, making it visible in search results and featured on the home page, and can download their resume in various templates using the data from their profile.

Overall, our project aims to fill a gap in the market for portfolio platforms that are designed specifically for our target audience, and we believe that there is significant demand for such a platform. Through our project, we hope to make it easier for individuals in these fields to showcase their work, connect with others in their industry, and ultimately achieve their professional goals.

Planning and Research

Overview

As a team, we recognize that thorough planning and research are essential components of any successful project. With this in mind, we dedicated significant time and resources to these activities during the initial stages of our project. We conducted extensive research to gain a comprehensive understanding of our target audience, their needs, and the existing solutions available in the market. We also engaged in discussions to identify the project's goals, scope, and requirements, ensuring that everyone on the team was on the same page. In this section of our project report, we will describe our planning and research process in detail, highlighting the methods and tools we used to achieve our goals.

Front-end

Front-End Development is a crucial aspect of any web project as it is responsible for creating the user interface and providing an interactive experience for the end-users. In the initial two weeks of our project, the team dedicated their efforts to understanding different libraries and frameworks that would be useful in building our front-end.

Front-end Time and Progression log					
Tasks	Start Date	Duration	Complete	Incomplete	Participants
Project design and planning	Jan12, 2023	16 hours	Overall app layout	Code Implementation	All Members
Installing libraries, setting up a GitHub repo, configuring the programming environment with Express.js, EJS, BootStrap, PHP and Node.js.	Jan 15, 2023	2 days	Set up new GitHub repo with support for Express.JS, PHP, Node.js and EJS	Code Implementation	Teh Kar Hock Aishwarya Tomer
SignIn page front-end implementation	Jan 20, 2023	2 days	Completed front-end functionality of SignIn	Design of navigation bar not finalized	Wei Shan Sim
SignUp page front-end implementation	Jan 23, 2023	2 days	Completed front-end functionality of SignUp page	Design of navigation bar not finalized	Wei Shan Sim
EditProfile and ViewProfile front-end implementation	Jan 21, 2023	8 days	Completed front-end functionality of EditProfile	Responsive design not implemented, design of navigation bar not finalized	Md. Iatisam Nawaz
Home page front-end implementation	Feb 5, 2023	3 days	Completed front-end functionality of ViewProfile page, design of navigation bar finalized	Responsive design not implemented	Md. Iatisam Nawaz

About and Template page implementation	Feb 12, 2023	2 days	Completed the implementation of About and Template pages, Responsive design implemented		Wei Shan Sim Md. Iatisam Nawaz
--	--------------	--------	---	--	-----------------------------------

Back-end

To ensure that the Back-end team had the necessary skills and understanding of key technologies, the project allocated three weeks at the beginning of the development process for learning. During this time, the Back-end team focused on gaining proficiency in key technologies such as Express.js, PHP, Node.js, and SQLite3.

After the initial learning period, the teams held at least two meetings per week to collaborate on the project and review each other's code. Individual team members continued to work on their allocated portions of the project, which were stored and shared through a repository on GitHub.

In addition, weekly meetings were held for the full team, including both the Front-end and Back-end teams, to discuss the overall status and progress of the project. This provided an opportunity to share specific updates that had taken place during the week and troubleshoot any issues that had arisen.

that require concurrent access to data. It's used in a variety of applications, ranging from embedded systems to web and mobile applications.

By utilizing these technologies, the Back-end team was able to effectively manage the flow of data between the Front-end and the database. This ensured that the application was both efficient and secure.

Back-end Time and Progression log					
Tasks	Start Date	Duration	Complete	Incomplete	Participants
Installing libraries, setting up a GitHub repo, configuring the programming environment with Express.js, EJS, BootStrap, PHP and Node.js.	Jan 15, 2023	2 days	Set up new GitHub repo with support for Express.JS, PHP, Node.js and EJS	Couldn't integrate PHP with Express.js	Teh Kar Hock Aishwarya Tomer
Back-end code for SignIn page	Jan23, 2023	2 days	Completed back-end code for SignIn page	Use SQLite3 database instead of MySQL in the PHP middleware Couldn't integrate PHP with Express.js	Teh Kar Hock
Back-end code for SignUp page	Jan 25, 2023	2 days	Completed back-end code for SignIn page	Use SQLite3 database instead of MySQL in the PHP middleware Couldn't integrate PHP with Express.js	Teh Kar Hock
Back-end code for viewProfile page	Feb 4, 2023	4 days	Completed back-end code for viewProfile page	Use SQLite3 database instead of MySQL in the PHP middleware Couldn't integrate PHP with Express.js	Aishwarya Tomer
Back-end code for editProfile page	Feb 5, 2023	7 days	Completed back-end code for editProfile page, add ejs templates to create forms to	Use SQLite3 database instead of MySQL in the PHP middleware Couldn't	Aishwarya Tomer



			update and add data into the database	integrate PHP with Express.js	
Back-end code for Home page	Feb 10, 2023			Use SQLite3 database instead of MySQL in the PHP middleware Couldn't integrate PHP with Express.js	Teh Kar Hock
Establish communication between PHP server and Express.js server to exchange information	Feb 10, 2023		Use SQLite3 database instead of MySQL in the PHP middleware Couldn't integrate PHP with Express.js		Teh Kar Hock Aishwarya Tomer

Team Subdivision

The team for the project comprised four active members, whose names are mentioned in the report's first page. As soon as the team was formed, it became apparent that dividing it into sub-teams would enhance the project's efficiency. This approach would allow for better management of the team's efforts and enable members to focus on specific areas of interest. Consequently, two members formed the front-end team, while the other two joined the back-end team.

Each sub-team focused on tasks related to their domain and held frequent meetings throughout the week. Furthermore, the sub-teams regularly met with each other to ensure the application's overall functionality.

Initially, the team consisted of five members, but Sieun Park, one of the team members, stopped attending meetings and failed to respond to messages. As a result, the remaining team members had to distribute the workload among themselves. The issue was addressed with Sieun Park, but the problem persisted. The team reported the problem to the appropriate student support and mentioned it in the project proposal.

Project Management Tools

In today's fast-paced and interconnected world, effective communication and collaboration are key to the success of any project. With so many tools available, it can be challenging to decide which ones to use for your team's specific needs. For our current project, we carefully selected a range of tools that will help us stay connected, share ideas, gather feedback, and work together seamlessly. In this section, we will briefly describe each tool, highlighting its features and benefits. By using these tools, we were able to streamline our workflow and enhance our productivity, ultimately leading to the success of our project.

Synchronous Communication: Google Meet

Google Meet is a video conferencing tool that allows team members to communicate with each other in real-time. With Google Meet, you can schedule and join video meetings from anywhere with an internet connection. You can also share your screen, record the meeting, and collaborate on documents together in real-time.

Asynchronous Communication: Slack

Slack is a messaging platform that enables team members to communicate with each other in real-time, as well as asynchronously. With Slack, you can organize conversations by channels or topics, share files and links, and integrate with other tools your team uses. Slack also offers various features, such as the ability to search conversations, set reminders, and customize notifications.

User testing: Google Forms

Google Forms is a survey tool that enables teams to gather feedback from users. With Google Forms, you can create surveys and questionnaires that can be distributed to users via email or social media. You can also track responses and view results in real-time, which can help you make data-driven decisions for your project.

Writing: Google Docs

Google Docs is a word processing tool that allows team members to create and collaborate on documents in real-time. With Google Docs, you can write, edit, and comment on documents with other team members simultaneously. You can also track changes, revert to previous versions, and export documents in various file formats.

Version Control: Git

Git is a version control tool that allows teams to manage and track changes to their codebase. With Git, you can create different versions of your code, collaborate on changes with other team members, and merge changes back into the main codebase. Git also offers features such as branching, tagging, and stashing.

Repository Hosting: GitHub

To ensure effective collaboration, the team set up a Git repository on GitHub, a web-based hosting service for version control using Git. This allowed us to track changes, manage different versions of our codebase, and collaborate with each other seamlessly. Furthermore, we created an internal document that documented important project information and resources used by the team to learn and research. This document was useful in keeping everyone on the same page and ensuring that we all had access to the necessary information to complete our tasks efficiently.

Prototyping and Iteration

Color Blindness:

While designing our website we ensured that the colors had good contrast and separation. We approved the color by choosing accessible color palettes. The mockups were well tested and approved by the users, this gave us the design approval for the colors and implementation. Similarly, the team ensured that the layout was kept simple so people with all ages are able to learn and navigate along the page using colors. However, to make our testing more robust we used a tool Color Oracle to test and check our website for color blindness. In the testing we used color filters that depicted how people with color blindness looked at our website. We used deutanopia and grayscale color filters to test the most common cases. All the changes that were made were also checked and approved.

User Testing

The initial mockups created in the design part were highly tested. The colors, navigation and flow of the website was thoroughly tested to ensure a user centered design. This allowed the front end team to stick closely to the high fidelity prototypes that were created and designed on Figma. However, after development, the first round of testing was conducted to test the implementation of the website and test it among new and old set of users. This allowed us to get a concrete evaluation of the website. We approached 10 cs students and asked them to test and evaluate our website.

First Iteration

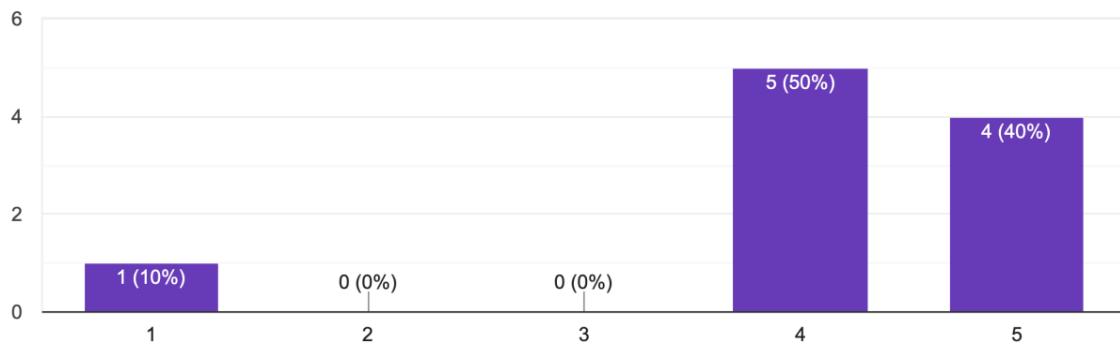
Design

The first testing focused more on the design choices of the website and if people were able to navigate through the screens easily.



What is your overall impression of the website's design?

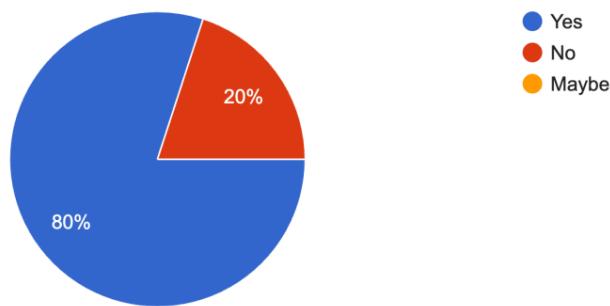
10 responses



User Testing (i): First Iteration (Design)

Were you able to navigate the website easily?

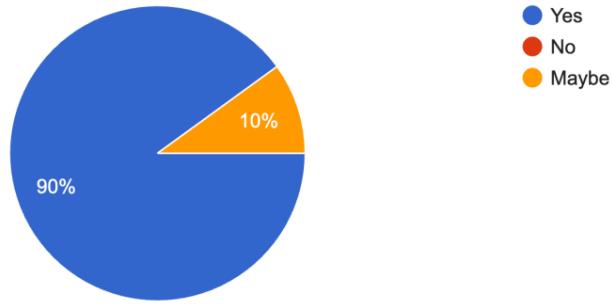
10 responses



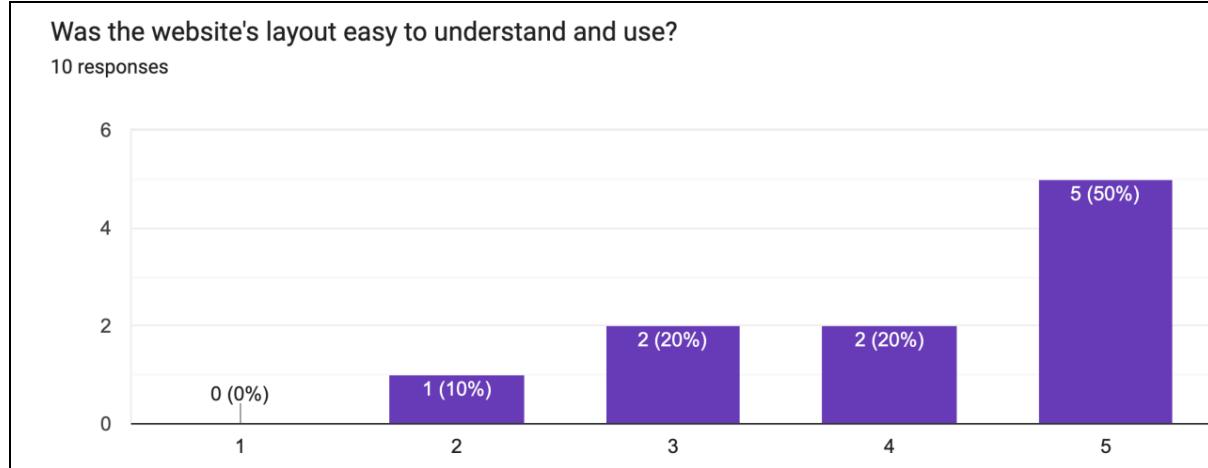
User Testing (ii) : First Iteration (Design)

Was it clear what the website's purpose was?

10 responses



User Testing (iii) : First Iteration (Design)



User Testing (iv) : First Iteration (Design)

Did you encounter any issues with the website's functionality? If yes then briefly mention

2 responses

The design is not responsive

The profile information was too big

User Testing (v) : First Iteration (Design)

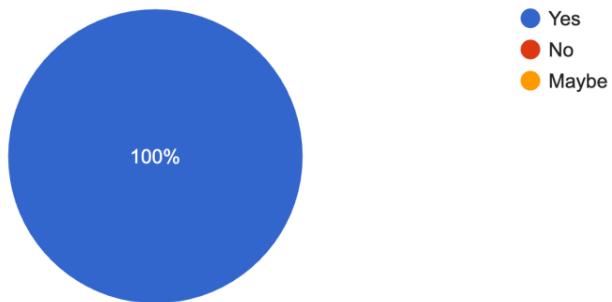
Functionality:

The second part of testing focused on functionality and usability of the website. As the website was dynamic and had a lot of user input therefore we separated the first Iteration testing into design and functionality.



Were you able to login and Register ?

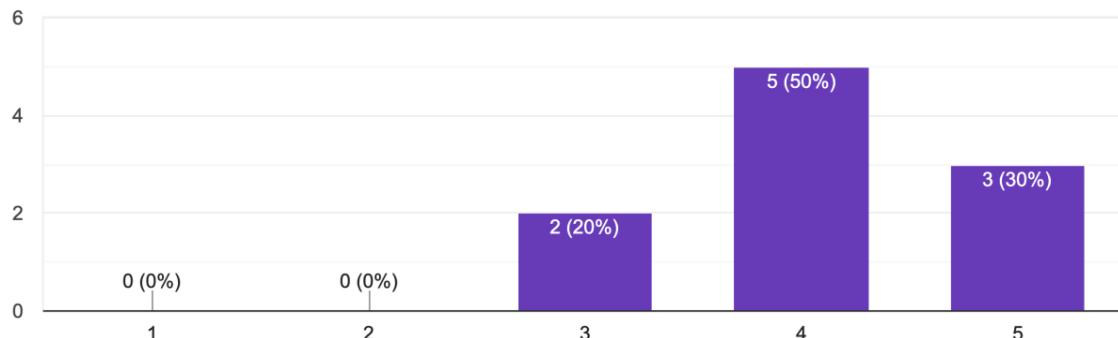
10 responses



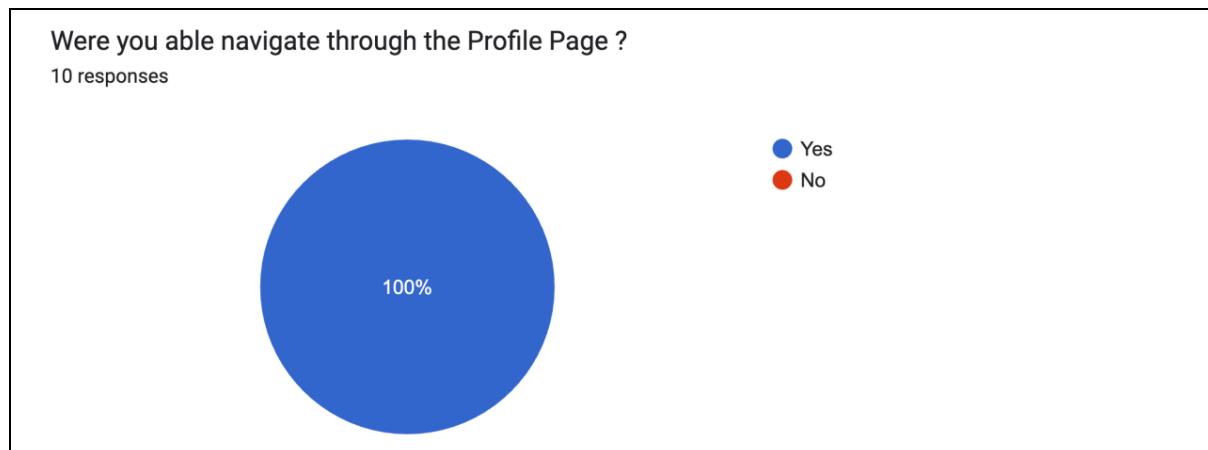
User Testing (i) : First Iteration (Functionality)

Rate the usability of the website ?

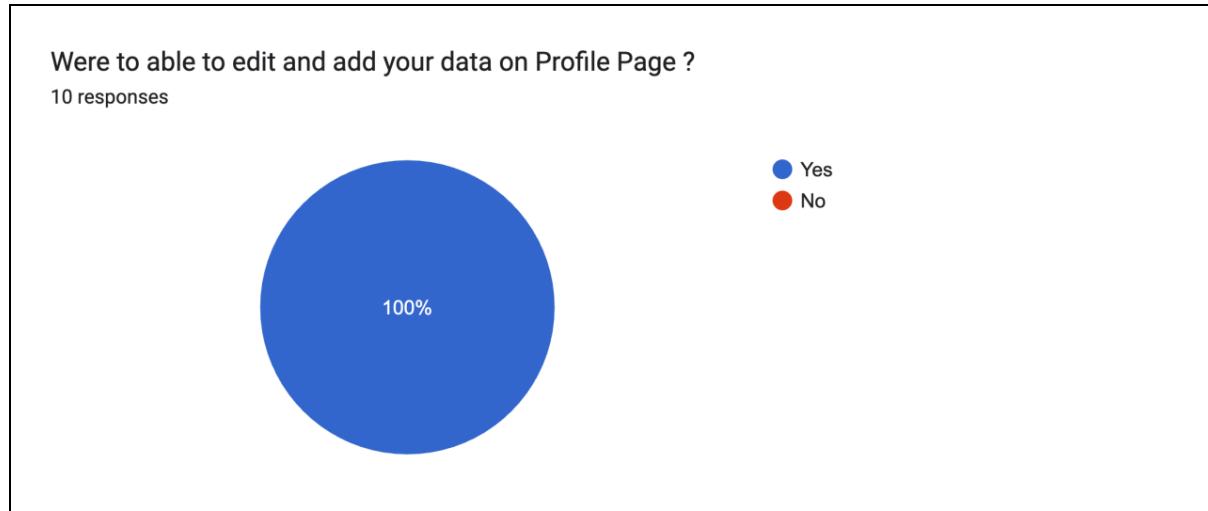
10 responses



User Testing (iii) : First Iteration (Functionality)



User Testing (iv) : First Iteration (Functionality)



User Testing (v) : First Iteration (Functionality)

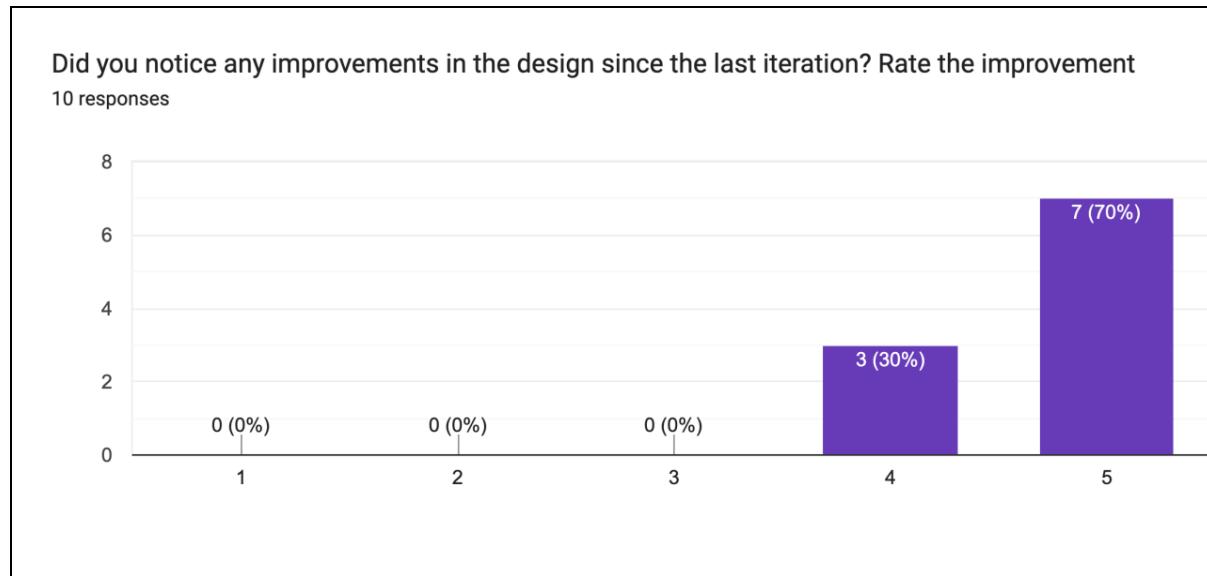
Analysis: First Iteration

The first set of user testing gave positive feedback overall. There were some issues that were noted in the design side. In the design side, the feedback ensured that the users were satisfied by the design choices and the implementation. However, among the users, some students used much smaller devices. This tested the responsiveness of the website on different screen sizes. After the feedback we ensured that the front-end team considered the user feedback and started working on the issues. Similarly, for the second part we got a lot of positive feedback where the users were able to use the website mostly.

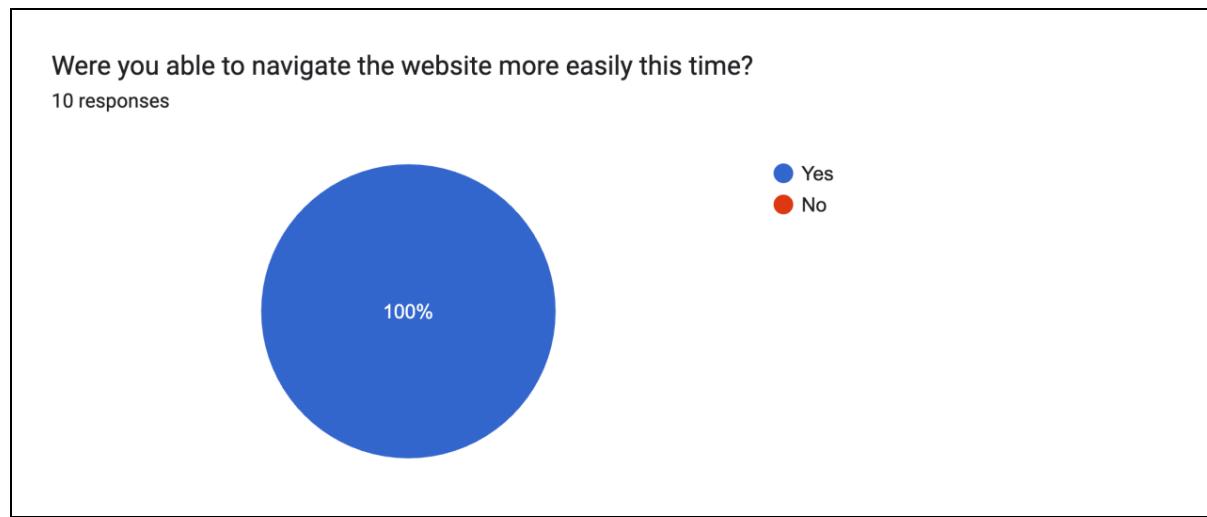
Second Iteration

Design

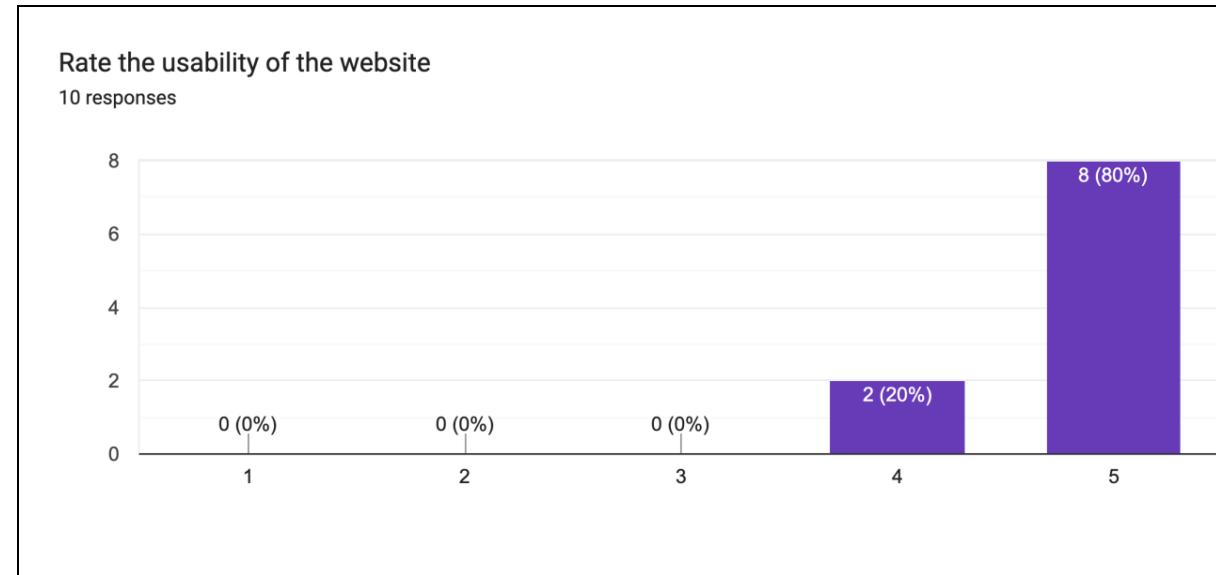
In the second iteration we made some design changes, to improve user experience, we made the pages more responsive, added a different layout for mobile users so students who used mobile phones could use it more smoothly. We reached the same students who did the initial testing and allowed them to test the design and functionality again. The target was to see if they had a better user experience this time.



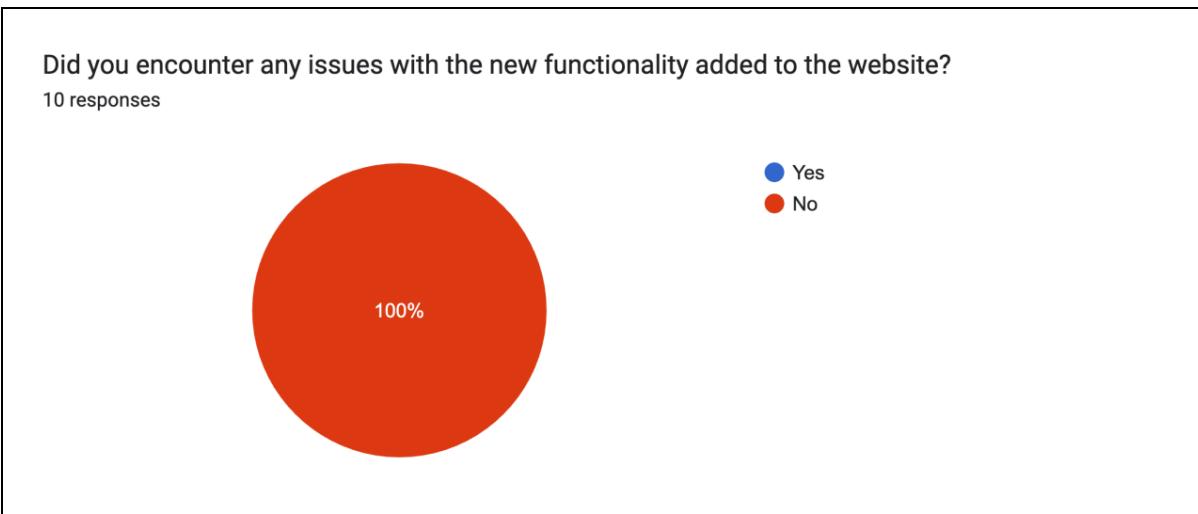
User Testing (i) : Second Iteration (Design)



User Testing (ii) : Second Iteration (Design)



User Testing (iii) : Second Iteration (Design)

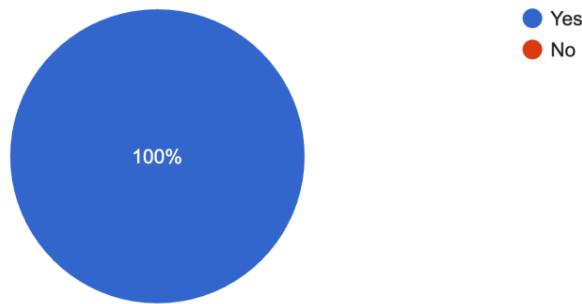


User Testing (iv) : Second Iteration (Design)

Functionality

Were you able to login to your account ?

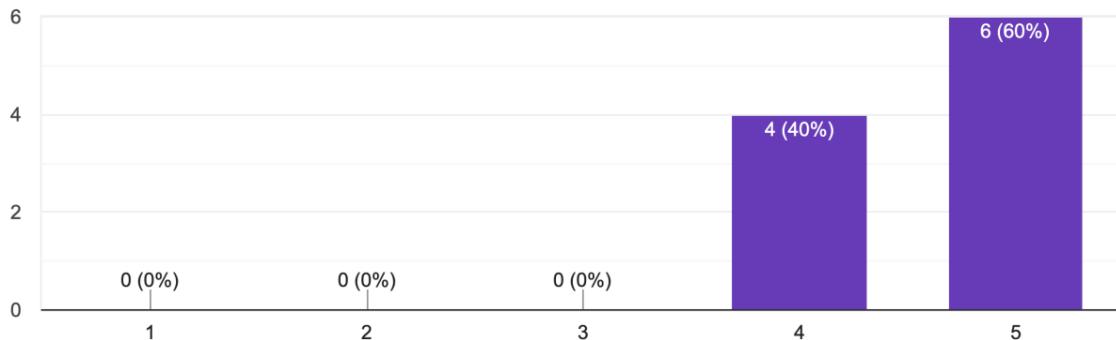
10 responses



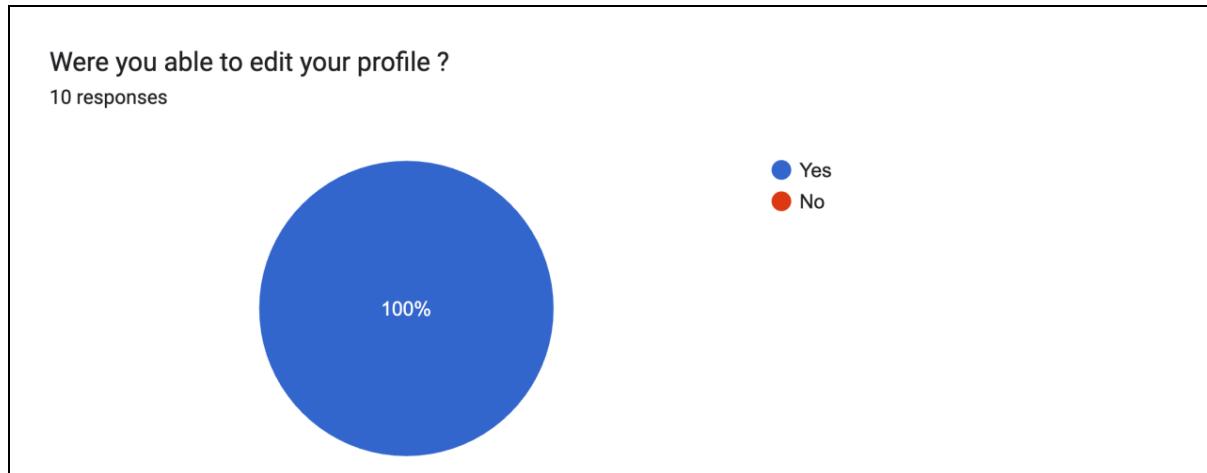
User Testing (i) : Second Iteration (Functionality)

Was the login experience better this time ?

10 responses



User Testing (ii) : Second Iteration (Functionality)



User Testing (iii) : Second Iteration (Functionality)

Second Iteration: Evaluation

In the second iteration, we were successful in fixing the issues. The user feedback from the first iteration allowed us to go through the areas where we were unable to figure out the issues. Similarly we were able to fix the responsiveness of the pages. Similarly in the login form, there were issues and we were able to figure out and fix the issue. All the students were able to login to their previous accounts.

Heuristic Evaluation:

User feedback provided valuable insights to the functionality and design, it allowed us to be confident on our designs and validate our choices, However it is not enough to give a concrete validation. By using multiple validation methods together, it helps the process to get smoother and allows the end user experience better. Therefore, we decided to conduct heuristic evaluation on the website. Heuristic analysis is a way of validation where we use a set of rules to configure issues on the website. We used System Usability Scale (SUS), that is a set of heuristics to evaluate usability of the website.

The set of rules are as following :

1. I think I would like to use this system frequently

Here we evaluate the system and respond on a scale whether we will use the system again, or more frequently. This allows us to see how enjoyable the system is.

**2. I found this system unnecessarily complex**

This indicates if the system is simple to use and does not make the user think that the system is complex .

3. I thought the system was easy to use

This tells how intuitive the system is and if it feels easier for the user.

4. I think i would need the support of a technical person to be able to use this system

This indicates whether the system is easier for the user to learn and use, It shows if the user will need assistance to use the platform.

5. I found that various functions in the system were well integrated

This depicts how consistent and cohesive the system is, This also indicates how positive the user id about the features added in the system

6. I thought there was too much inconsistency in the system

This indicates if the user thought the features did not go well with the system and if they find inconsistency in the system.

7. I imagine that most people would learn to use the system very quickly

This indicates how easy it will be for the user to get comfortable and learn the flow of the website

8. I found the system very cumbersome to use

This scale assesses the efficiency and effectiveness of the system

9. I felt very confident using the system

These scales assess the confidence of the user while using and interacting. This also indicates how comfortable the user felt.

10.I needed to learn a lot of things before i could get going with the system

This indicates if the user had to learn a lot before learning how to use the website.



Evaluation :

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I think I would like to use this system frequently	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found this system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I thought the system was easy to use	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I think i would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
I found that various functions in the system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I thought there was too much inconsistency in the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
I imagine that most people would learn to use the system very quickly	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I felt very confident using the system	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I needed to learn a lot of things before i could get going with the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Initial Design:

The mockups that were created were well thought and planned. As our UX and UI planning was well executed and we had done user testing for the design and theme. Our front end team confidently followed the design ideas that were proposed and approved. There were some minor tweaks in the design choices. For example changing the round design to a more elegant straight design on the landing sign up page. The initial flow and color theme was maintained as this was tested and had great user experience. However, It was challenging to create a responsive design that allowed the users to use it on multiple devices. Therefore, the front-end team did put effort in making the website compatible for mostly used screen sizes. The head part of the website was initially designed using the same layout for desktop.



Second Iteration

The second iteration was based on the user testing we performed. We took the user feedback for each page and for the overall execution. The main thing that we had to work on after having the first iteration was to make the profile page responsive in order to allow more users to interact. Other than that people rated design and usability good. For the responsiveness we created a slightly different layout to aid smaller screens. The user tests conducted after the second iteration approved the designs and the implementation. People were able to navigate through the page and were able to read the text. The figure below shows how the profile page looked after the second iteration. Users are able to view information cards and can keep the cards hidden to keep things minimal. We kept the about section below, this created a more practical and easy to navigate user experience. The image below shows how the page looks on a tablet.

About MySelf:

Placeholder text for the 'About MySelf:' section.

Skills:

Express

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Express

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Express

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

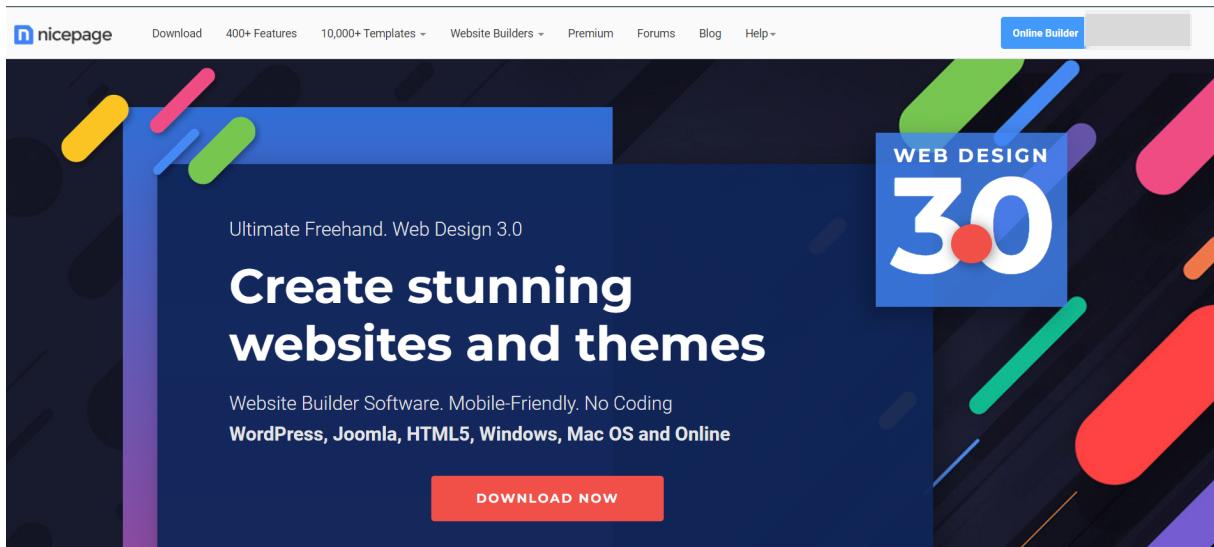
We also separated the input page to add user data. This allowed the users to have a more intuitive and easier experience to add data on the profile page. While we performed heuristic analysis on every page we found out that when we use bigger screens it was easier to add information but on smaller screens it got challenging to choose the editable boxes and it was difficult for users to get a mistouch. Therefore, we added an edit page that would allow users to add their information in a more easier way.

Design

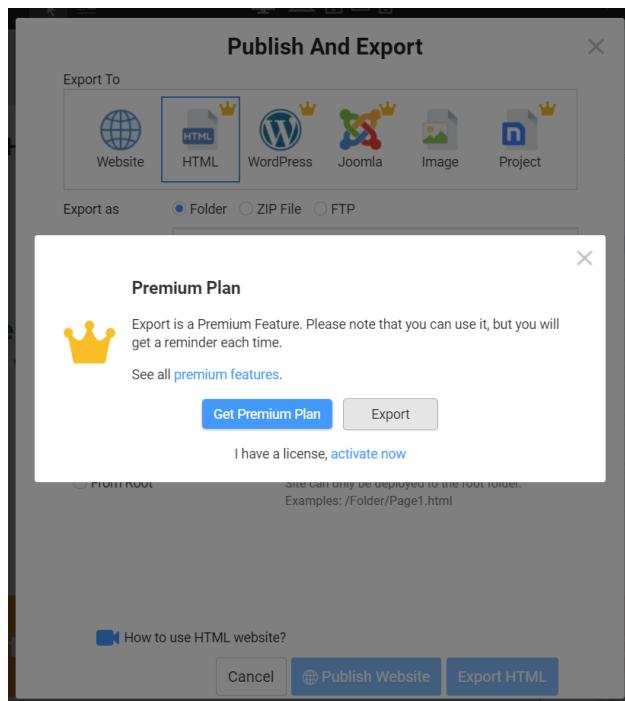
Our development team is trying to produce the end product that mostly matches the high fidelity prototype where we proposed in the project proposal and we also need to pay attention to the time frame that we set.

Front end Supported Tool

To pursue the quality of product and follow the time frame, we used a website builder software called nicepage[2] to produce the front end part. Nicepage is a “simple drag-and-drop website builder software to implement any ideas” and coding needed during producing the web pages, we are able to visually add, edit, move, and modify with no coding, this builder became a great tool in our development process.



Nicepage allows users to export the designed web pages to HTML for free of purchase, and there are limited functions provided to users using the free plan for sure. The approach that we choose is to register as a free plan user, try our best to implement the functions we need by using the limited functions provided from nicepage. For the parts which are unable to be implemented straight in nicepage, we export it into HTML and edit it in the HTML and CSS code manually to match our plan.



Another advantage is it is able to adjust the arrangement of the design of web pages to fix the various sizes of screen. It makes our web pages still keep the design even if it's open by the smaller screen compared with desktop, such as laptop and tablet, it will not make the web pages become messy in the smaller window view.

Perfect Results On Any Device

Web pages are absolutely mobile-friendly and look perfect in all device views

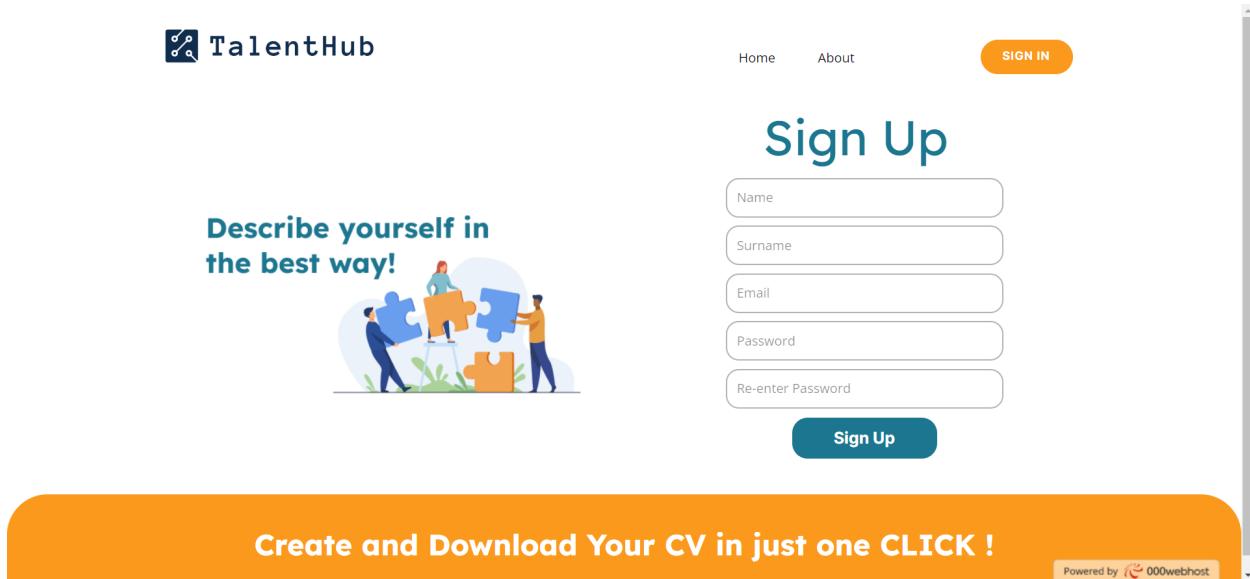


Total Pages of The Website

We have increased the total pages in the website from 5 to 6 main pages in our website, which are Sign in, Sign up, Home, My Profile, Template page and About as additional pages. After that we currently have another 1 page for the supporting subpages, which is the viewProfile page bind with the Home page when the user clicks one of the cards in the Home page.

Front end – Sign In and Sign Up Pages

The screenshot shows the TalentHub sign-in page. At the top, there is a navigation bar with the TalentHub logo, Home, About, and a yellow 'SIGN UP' button. Below the navigation bar, the page title 'Sign in' is displayed in large blue text. To the left of the form, there is a teal-colored call-to-action 'Describe yourself in the best way!' above an illustration of three people connecting puzzle pieces. The main sign-in form consists of two input fields: 'Email address' and 'Password', both with placeholder text. Below these fields are 'Remember Me' and 'Forgot password?' links. At the bottom of the form are two buttons: a teal 'Login' button and an orange 'As Guest' button. A large orange banner at the bottom of the page encourages users to 'Tell people more about your professional life'. In the bottom right corner, there is a small 'Powered by 000webhost' logo.



The Sign in and Sign up pages remained the purpose of design from the high fidelity prototype. The Sign in page is the page that allows users to sign in to the website by using registered email and password.

The last part of the modification is the hint of the names fields, we changed from first name and last name into name and surname. Depending on the cultural identification, the arrangement of names for a person might be totally different as everyone knows. The arrangement of the names for the mostly western country, first name is the given name and the last name is family name. But in the cultures of Chinese and Japanese, the names arrangement is totally different, in which the family name comes first and then the given name at the last[4]. To clear the confusion to certain types of users, we changed the hint to Name and Surname.

All the modifications are based on the comfortable level of the worldwide users, because we could not ensure the high rate of the visit rate but we can avoid the factors that will bring negative effects to the visit rate of the website.

Front end – Home Page

The screenshot shows the TalentHub home page. At the top, there is a navigation bar with the logo 'TalentHub' (featuring a stylized person icon), 'Home', 'About', and a prominent orange 'SIGN UP' button. Below the navigation bar, a search bar is centered with the placeholder 'Search'. A teal-colored header 'Newly Added' is displayed above two profile preview cards. The first card for 'Teh Kar Hock' (MY) shows a placeholder profile picture and the text 'nothing much'. The second card for 'Isabella Sanchez' (Sydney, Australia) shows a placeholder profile picture and a detailed bio: 'I have extensive experience in developing and implementing marketing strategies for various industries such as technology, healthcare, and education. I am a creative thinker with a passion for data-driven marketing.' In the bottom right corner of the page, there is a small watermark 'Powered by 000webhost'.

Home page view without sign in or sign up

This screenshot is identical to the one above, showing the TalentHub home page without sign in or sign up. The only difference is the presence of a 'LOG OUT' button in the top right corner of the navigation bar, indicating that a user is currently logged in.

Home page view after sign in

We still keep the design of the Home page to focus on our website's main purpose, which is to show the users' profile as listed at the middle of the Home page. Whatever visitors or registered users are able to click one of the listed profile preview cards, it will jump to another page we called `viewProfile`, to view the visible information of the owner's profile. To cooperate with the Guest mode, the visitors who did not sign up on

our website are not allowed to go to the Profile page, which is the page that allows users to enter their information. When visitors click on the Profile link on the navigation bar, it will redirect to the Sign in page to encourage visitors to register on the website.

Front end – Profile Page

Add New Profile

Job Position:

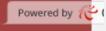
Location:

Contact No:

About Yourself:

About More:

[Add Profile](#)

Powered by 

Form for new user

Profile owner view

The Profile page is the core function of the website, we keep the mostly design from the high fidelity prototype and try to implement all of our ideas. Firstly, there will be a form named “Add New Profile”, for the new users who haven't created the profile after clicking the Profile button in the navigation bar in Home page.

The first part of the content of the Profile page is the basic information section which are stated name, job title, location, email, contact number and about myself. These fields match the top section in the resume and all of these are able to be edited by users by clicking the edit icon at the top right as we are using the “editable-heading” function implemented in the linked javascript file.

The following part of the content of Profile page is the collapsible contents and are collapsed as default view. Users are able to edit or add information after clicking the target field, inside including 2 icons, which are add and edit. As the shown icons, users are able to add new information or edit information in the target fields. From here, to implement the add and edit purpose, we are using 2 javascript files to reach our plan, which are addSkill.js and editSkill.js. The entered information will become cards in the field to achieve the purpose of data visualization.

Beside this, we implement a special function in the Profile page, which is that each of the information is able to set the status of visible. Users are able to edit the visible status of an information in the sub page of the target field after clicking the edit icon. At



the button part of the sub page, there is a checkbox able to tick, which is the place able to set the status visible of the information. If the checkbox is ticked, the eye icon on the information card will show as invisible; otherwise, the eye icon on the information card will show as visible. After that, if other users view the profile from the Home page, the information set as invisible will not be able to be viewed.

The purpose of making users able to set the visible status of the information cards is to implement the idea of protecting the privacy of users and to maintain the users confidence and comfort when using our website. We support users being able to hide their personal information from the view of others. From another perspective, users can enter all their information without thinking which parts should be visible to the public and which parts need to be invisible.

Next, as we planned at the high fidelity prototype, we will only expand the sub page in the Project field when users want to edit it. But now, every field in the second half part of the Profile page will extend the sub page when users edit it. It provides a more comfortable view to the users when doing the editing in the Profile page.

Unfortunately, the Profile page took more time to implement than we expected. We were unable to implement the header in the Profile page and the Resume button at the top right of the Profile page should be linked to the Template page which is able to convert user information into a resume and download it, but we are unable to link to the Template page due to time shortage.

The way to get out from the Profile page is using the backward arrow of the browser.

Front end – Template Page

From the high fidelity prototype, the Template page is the place where users can download their resume by selecting their preferred type provided from the website. But due to time shortage, we are unable to implement this feature to the users. However, we still create front end design from the high fidelity prototype without the back end program as we expected before. After that, before the feature is able to be implemented, it will show a message box that informs users when the user clicks the download button.

In this page, the header is still the same as in the Home page, which is able to link to the Home page, About page, Profile page or go to the Sign in page.

Due to time shortage, this page is unable to be implemented into our website, but the program files are still included in the attachment. There is still a way to go into this page by typing the link, but the normal user without that knowledge is confirmed couldn't access it.

Front end – About Page

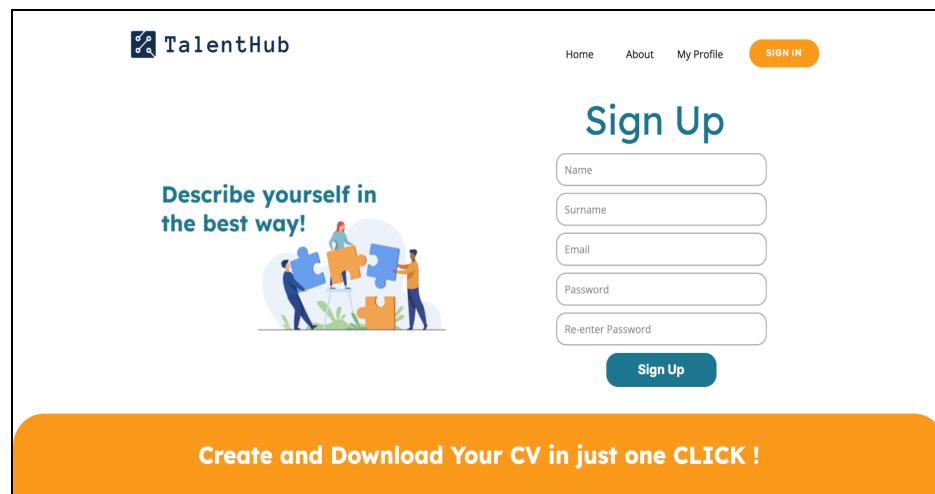
The screenshot shows the 'About' page of a website called 'TalentHub'. At the top, there is a navigation bar with links for 'Home', 'About', 'My Profile', and 'LOG OUT'. Below the navigation, a large heading reads 'Connect with others by share your profile'. Underneath this, a subtext states: 'Here is a comprehensive portfolio platform that helps freelancers, creative professionals, students, and educators create and host professional profiles, and connect with others in their industry.' A callout box below the subtext says 'Only 2 steps !'. Another callout box below it says 'HOW ?'. A third callout box below that says 'Step 1: Insert your information in My Profile'. A blue downward-pointing arrow icon is positioned next to the 'Step 1' text. In the bottom right corner of the page, there is a small watermark that says 'Powered by 000webhost'.

In the About page, have simple user guidelines for the users that show the easiest way to use our website.

Specifications

User Accounts

Users can create an account by providing their name, email address, and password.



Sign Up page of the Application

User Profiles

Each user can create and manage their own profile. They can add, edit, or delete details about their skills, education, career, awards, and more. They can also upload a profile picture and choose whether or not to make their profile public or private. Users can also select which sections of their profile they want to share publicly.

Profile page displaying all sections in a collapsed state

The profile of the user will consist of various sections(see Fig 4.2), such as a header which displays the user's profile picture, basic contact information and a brief introduction, and a primary section that contains the user's portfolio, awards, skills, and other pertinent information.

Jane Doe
Back-End Developer
New York

janedoe@gmail.com
+92234567890
Unpublish

About MySelf

Express

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

JavaScript

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Awards:

Projects:

Education:

User's profile page displayed in edit mode within their account.

Users will have the option to modify and append information that is exhibited on their profile within the application (refer to Fig 4.2 and 4.3). The privacy setting will be customizable by the user as they can choose to make their profile either public or private. Users will also be able to update their profile picture and revise the specifics of their skills, awards, and other relevant information.

Edit Skill

Skill:
Express

Summary:

This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

More Details:

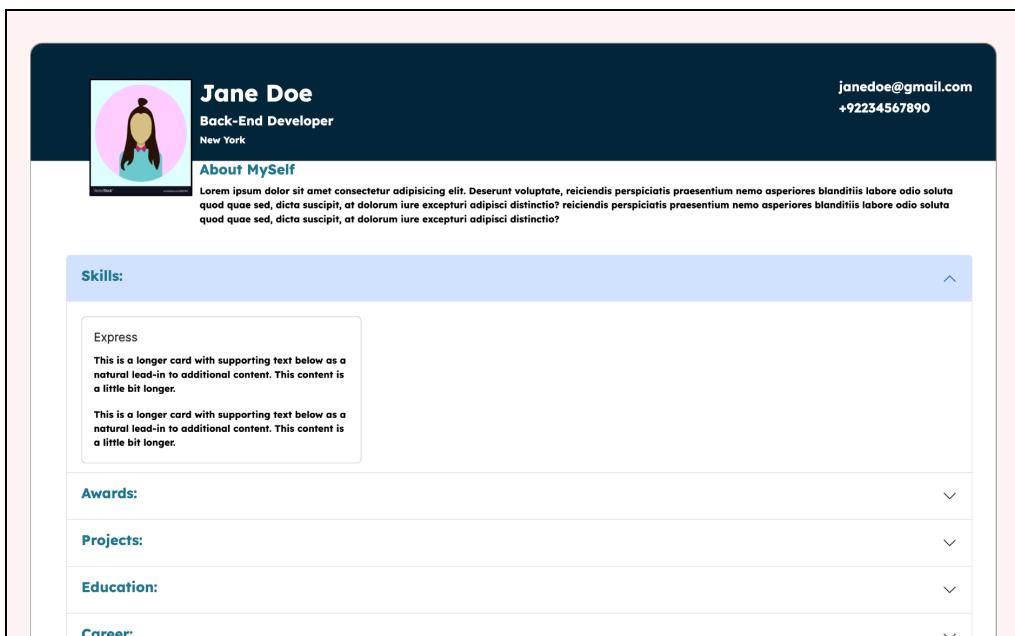
This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Make public

Update

Delete

Form for editing the information and visibility status of a user's skill



User's profile page when viewed by other users

The profile page of the user, visible to other users, is presented in Figure 4.5. It should be emphasized that only the skills that the user has chosen to make public will be viewable by others.

Home Page

The web application will have a user-friendly interface, with clear and intuitive navigation. The homepage will feature a search bar and a list of newly created profiles, and will provide easy access to the user's own profile and to the profile creation and customization tools.

The screenshot shows the TalentHub home page. At the top, there is a navigation bar with the logo 'TalentHub', 'Home', 'My Profile', and a 'Sign In' button. Below the navigation bar, the heading 'Newly added' is displayed above a grid of four profile cards. Each card contains a small profile picture, the name of the user, their location, and a brief description of their experience.

- Tyron Lannister**, from Salt Lake City, UT. Description: Highly Motivated And Detail-Oriented Individual With Over Five Years Of Customer Service Experience. Proven Ability To Build And Maintain Positive Relationships With Clients, With A Track Record Of Meeting And Exceeding Customer Expectations.
- Sarah Jones**, from New York, NY. Description: I Am A Creative And Detail-Oriented Graphic Designer With Over Seven Years Of Experience In The Industry. I Have A Strong Portfolio And A Track Record Of Producing High-Quality Work That Meets And Exceeds Client Expectations.
- John Smith**, from Los Angeles, CA. Description: I Am An Experienced Software Engineer With A Passion For Developing Innovative Solutions. I Have A Strong Background In Full-Stack Development And A Proven Track Record Of Delivering Successful Projects On Time And Within Budget.
- Mary Williams**, from Houston, TX. Description: I Am An Experienced Project Manager With A Background In Construction And Real Estate. I Have A Proven Track Record Of Leading Successful Projects From Start To Finish And Am Skilled At Managing Budgets, Schedules, And Team Members.

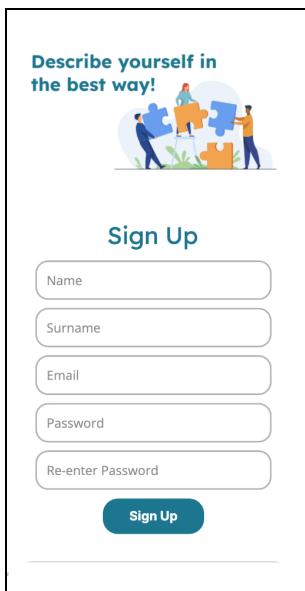
Home page of the Application

Responsive Design

The web application will be mobile-responsive and will support a range of modern browsers.

The screenshot shows the TalentHub SignUp page. At the top, there is a navigation bar with the logo 'TalentHub', 'Home', 'About', 'My Profile', and a 'SIGN IN' button. The main section features a heading 'Sign Up' and a form with five input fields: Name, Surname, Email, Password, and Re-enter Password. Below the form is a 'Sign Up' button. To the left of the form, there is a call-to-action 'Describe yourself in the best way!' with an illustration of two people assembling puzzle pieces. At the bottom, a large orange button contains the text 'Create and Download Your CV in just one CLICK !'.

Landscape view of the SignUp page



Portrait view of SignUp page

Approaches

Development Methodology

Agile Development

The platform will be built using an agile development methodology, with regular feedback and iteration cycles. The development team will work in two-week sprints, during which they will build and test new features, receive feedback from stakeholders and end-users, and make necessary adjustments to the product roadmap.

Regular Feedback and Iteration Cycles

To ensure that the platform meets the needs and expectations of its users, the development team will regularly solicit feedback from stakeholders and end-users. This feedback will be used to identify areas for improvement, prioritize new features and enhancements, and make necessary adjustments to the product roadmap.

In addition to soliciting feedback from stakeholders and end-users, the development team will also prioritize regular code reviews and testing to ensure that the platform is stable, secure, and scalable. This will involve using continuous integration and automated testing tools to catch and fix bugs early in the development process, as well as conducting manual testing and quality assurance checks to ensure that the platform meets high standards of quality and performance.

Technology Stack

The Technical Considerations

When selecting the tech stack for the application, the team considered various technical factors such as ease of use, scalability, reliability, and security. However, the team also took into account the individual skills and preferences of team members to ensure a balanced workload and leverage each member's strengths.

Team Members' Skills and Preferences

One member of the team had more experience with Express and felt more comfortable working with JavaScript-based frameworks, while another member had more experience with PHP and preferred working with that language. Therefore, the team decided to use both Express and PHP in the application to leverage the strengths of each team member.

Express API Server

For the user profile section, the team chose to use Express and ejs. Express is a Node.js framework that provides a set of features for building web applications, including routing, middleware, and template engines. The team decided to use ejs as the template engine for the user profile section because of its simplicity and flexibility. With ejs, the team can easily embed JavaScript code in HTML templates and generate dynamic content for the user profile pages.

PHP API Server

For the home page, sign-in, sign-up, and search functionality, the team decided to use HTML, CSS, and PHP. HTML and CSS will be used to design and style the front-end of the application, while PHP will be used to handle server-side logic, such as user authentication and database interactions.

SQLite3

The team chose SQLite3 as the database management system for the application because of its simplicity, reliability, and scalability. With SQLite3, the team could easily store and retrieve user data, search for users based on various criteria, and perform other essential database operations.

The team's decision to use both Express and PHP in the application allowed them to leverage each member's skills and preferences while also providing a balanced



workload. By using the appropriate tools and technologies, the team was confident that they would be able to develop a high-quality, scalable, and reliable application for users.

Project Timeline

The development of the Comprehensive Portfolio Platform will take place over a six-month period, with a detailed plan for each stage of the project. The timeline is broken down into the following stages:

Planning and Design

During the initial two-month phase, the team will work together to gather requirements, define the scope of the project, and create detailed plans for the development of the platform. This stage will include the following tasks:

- Defining the target audience and user personas
- Creating user stories and feature lists
- Creating wireframes and prototypes
- Defining the tech stack and infrastructure

User Profile Section

The first deliverable will be a functional prototype of the user profile section, to be completed by the end of the third month. This stage will include the following tasks:

- Building the back-end with Express and ejs
- Developing the front-end using HTML, CSS, and JavaScript
- Implementing user authentication and authorization
- Integrating database functionality with SQLite3

Home Page and Search Functionality

The second deliverable will be the home page and search functionality, to be completed by the end of the fifth month. This stage will include the following tasks:

- Designing the layout and style of the home page with HTML and CSS
- Creating the search functionality using PHP and SQLite3
- Implementing user registration and login
- Building the back-end for the home page with PHP

Testing and Final Polish

The final month of the project will be dedicated to testing, bug fixes, and final polish before release. This stage will include the following tasks:

- 
- Conducting functional testing to ensure all features work as expected
 - Identifying and fixing any bugs or issues
 - Finalizing the design and layout of the platform

System Development

Software Components

The software consists of two components: the web app, which is the front-end part of the system, and the back-end, which processes data from the database and delivers it to the front-end for display. The back-end acts as the intermediary between the database and the web app.

Commit	Author	Message	Time Ago
Create README.md	aishtomer	74b8684 10 hours ago	57 commits
removing sample images	jsServer	20 hours ago	
completed the back-end work for edit and view profile.	phpServer	20 hours ago	
completed the back-end work for edit and view profile.	.DS_Store	20 hours ago	
Create README.md	README.md	10 hours ago	
removing sample images	database.db	20 hours ago	
completed the back-end work for edit and view profile.	db_schema.sql	20 hours ago	

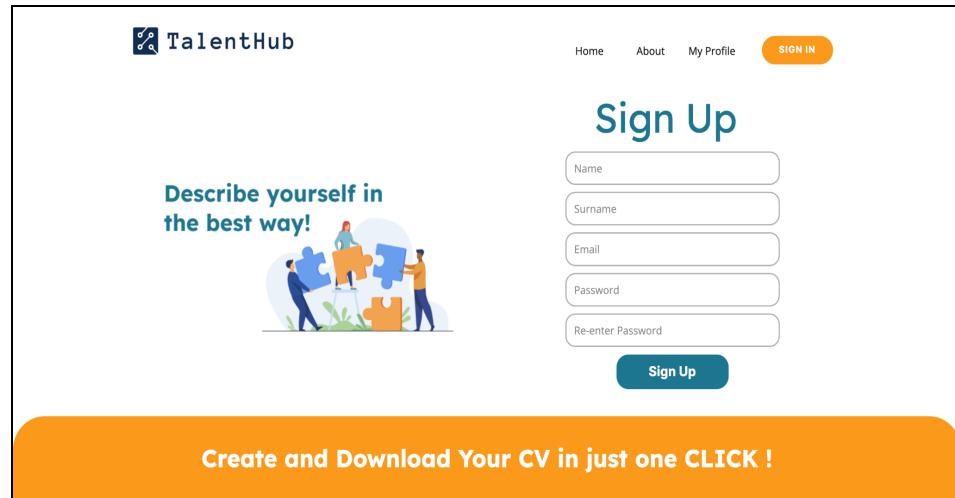
GitHub Repository for the Application

Presentation Layer

The web application is the main presentation layer of this project and runs on devices that support browsers such as Google Chrome, Microsoft Edge or Safari on their respective latest versions. They all need to have JavaScript enabled.

Express.js and PHP are both popular choices for back-end web development, while EJS is a templating language that can be used for both front-end and back-end development.

Using technologies like Express.js, PHP, and EJS can offer several advantages over traditional HTML, CSS, and JavaScript for building web applications that involve rendering data from a database. These technologies provide modularity, performance, scalability, flexibility, familiarity, compatibility, reusability, dynamic content, and expressiveness, which can save time and effort in development and improve the overall quality and functionality of the application.



EXPLORER

```

TALENTHUB
  > iServer
  > node_modules
  > public
    > images
    > scripts
    > uploads
    > routes
    > views
      > addAward.ejs
      > addCareer.ejs
      > addEducation.ejs
      > addProject.ejs
      > addSkill.ejs
      > editAward.ejs
      > editCareer.ejs
      > editEducation.ejs
      > editMoreAbout.ejs
      > editProfile.ejs
      > editProfileHeader.ejs
      > editProject.ejs
      > editSkills.ejs
      > viewProfile.ejs
    > index.js
  > package-lock.json
  > package.json
  > phpServer
  > images
  > scripts
  > styles
  > Absolt.html
  > editProfile.html
  > home.html
  > Landing-page-or-Login-page.html M
  > Sign-up.html M
  > Template.html
  > OUTLINE
  > TIMELINE

```

Sign-Up.html M

```

<!DOCTYPE html>
<html style="font-size: 10px;" lang="en"><head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta charset="utf-8">
<meta name="keywords" content="Sign Up">
<meta name="description" content="">
<title>Sign Up</title>
<link rel="stylesheet" href="/styles/nicepage.css" media="screen">
<script class="u-script" type="text/javascript" src="/scripts/jquery.js" defer=""></script>
<meta name="generator" content="Nicepage 5.5.0, nicepage.com">
<meta name="referrer" content="origin">
<link id="u-theme-google-font" rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:100,100i,200,300,400,400i,500,500i,700,700i,800,800i" type="text/css">
<link id="u-page-google-font" rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lexend:Deca:100,200,300,400,500,600,700,800" type="text/css">
<script type="application/ld+json">
  <@context: "http://schema.org" 
  <@type: "Organization" 
  <name: "" 
  <logo: "/images/websiteLogo.png" 
</script>
<meta name="theme-color" content="#478ac9">
<meta name="og:title" content="Sign Up">
<meta property="og:type" content="website">
<meta data-dpr="1" data-device="cd" data-layout="1" data-output="1" data-lang="en" data-image="1" data-animation="1" data-image-width="512" data-image-height="88" title="Home page">
<body class="u-body u-clearfix u-overlap" data-lang="en"><header class="u-clearfix u-header" id="sec-6a10" data-animation-name="" data-animation-duration="0.3s" data-animation-ease="ease-in-out">
<a href="Home-page.html" class="u-image u-logo u-image-1" data-image-width="512" data-image-height="88" title="Home page">

</a>
</header>
<div class="u-repeater-item" data-repeater-item="1" data-repeater-index="1">
<div class="u-repeater-item" data-repeater-item="2" data-repeater-index="2">
<div class="u-repeater-item" data-repeater-item="3" data-repeater-index="3">
<div class="u-repeater-item" data-repeater-item="4" data-repeater-index="4">
<div class="u-repeater-item" data-repeater-item="5" data-repeater-index="5">
<div class="u-repeater-item" data-repeater-item="6" data-repeater-index="6">
<div class="u-repeater-item" data-repeater-item="7" data-repeater-index="7">
<div class="u-repeater-item" data-repeater-item="8" data-repeater-index="8">
<div class="u-repeater-item" data-repeater-item="9" data-repeater-index="9">
<div class="u-repeater-item" data-repeater-item="10" data-repeater-index="10">
<div class="u-repeater-item" data-repeater-item="11" data-repeater-index="11">
<div class="u-repeater-item" data-repeater-item="12" data-repeater-index="12">
<div class="u-repeater-item" data-repeater-item="13" data-repeater-index="13">
<div class="u-repeater-item" data-repeater-item="14" data-repeater-index="14">
<div class="u-repeater-item" data-repeater-item="15" data-repeater-index="15">
<div class="u-repeater-item" data-repeater-item="16" data-repeater-index="16">
<div class="u-repeater-item" data-repeater-item="17" data-repeater-index="17">
<div class="u-repeater-item" data-repeater-item="18" data-repeater-index="18">
<div class="u-repeater-item" data-repeater-item="19" data-repeater-index="19">
<div class="u-repeater-item" data-repeater-item="20" data-repeater-index="20">
<div class="u-repeater-item" data-repeater-item="21" data-repeater-index="21">
<div class="u-repeater-item" data-repeater-item="22" data-repeater-index="22">
<div class="u-repeater-item" data-repeater-item="23" data-repeater-index="23">
<div class="u-repeater-item" data-repeater-item="24" data-repeater-index="24">
<div class="u-repeater-item" data-repeater-item="25" data-repeater-index="25">
<div class="u-repeater-item" data-repeater-item="26" data-repeater-index="26">
<div class="u-repeater-item" data-repeater-item="27" data-repeater-index="27">
<div class="u-repeater-item" data-repeater-item="28" data-repeater-index="28">
<div class="u-repeater-item" data-repeater-item="29" data-repeater-index="29">
<div class="u-repeater-item" data-repeater-item="30" data-repeater-index="30">
<div class="u-repeater-item" data-repeater-item="31" data-repeater-index="31">
<div class="u-repeater-item" data-repeater-item="32" data-repeater-index="32">
<div class="u-repeater-item" data-repeater-item="33" data-repeater-index="33">
<div class="u-repeater-item" data-repeater-item="34" data-repeater-index="34">
<div class="u-repeater-item" data-repeater-item="35" data-repeater-index="35">
<div class="u-repeater-item" data-repeater-item="36" data-repeater-index="36">
<div class="u-repeater-item" data-repeater-item="37" data-repeater-index="37">
<div class="u-repeater-item" data-repeater-item="38" data-repeater-index="38">
<div class="u-repeater-item" data-repeater-item="39" data-repeater-index="39">
<div class="u-repeater-item" data-repeater-item="40" data-repeater-index="40">
<div class="u-repeater-item" data-repeater-item="41" data-repeater-index="41">
<div class="u-repeater-item" data-repeater-item="42" data-repeater-index="42">
<div class="u-repeater-item" data-repeater-item="43" data-repeater-index="43">
<div class="u-repeater-item" data-repeater-item="44" data-repeater-index="44">
<div class="u-repeater-item" data-repeater-item="45" data-repeater-index="45">
<div class="u-repeater-item" data-repeater-item="46" data-repeater-index="46">

```

Web Application and its HTML Code

Structure and Implementation

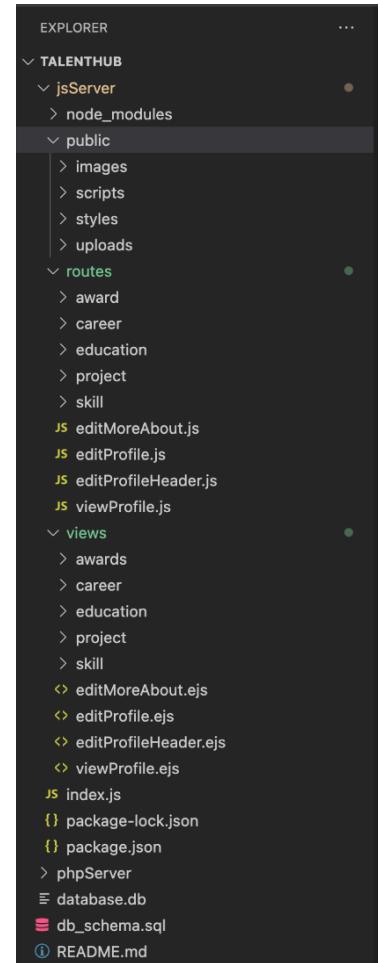
Express API Server

The Express.js part of the application is responsible for handling the server-side logic and rendering of the `viewProfile` and `editProfile` pages. This involves creating routes and defining middleware functions to handle incoming HTTP requests and generate responses.

In an Express.js application, there are several key files and folders that are used to organize and structure the code. These include:

- `index.js`: This is the main entry point for the application, where you initialize the Express.js app, set up middleware, and define routes.
- `routes/` folder: This folder contains files that define the routes for the application, mapping URLs to functions that handle HTTP requests.
- `views/` folder: This folder contains the EJS templates for rendering the HTML content of the view and edit profile pages.
- `public/` folder: This folder contains static assets such as images, CSS files, and client-side JavaScript files that are used by the front-end of the application.

When you visit the localhost in your browser, the first file that runs is `index.js`. This file initializes the Express.js app by creating an instance of the `express` module, setting up middleware functions, and defining routes.



Structure of jsServer

Below code defines a route for the URL /profile that handles HTTP GET requests by rendering the profile.ejs template and passing in some data to be displayed on the page.

When a user visits the /edit URL in their browser, the Express.js app will execute this route handler function and generate a response that includes the HTML content of the editProfile.ejs template.

The views/ folder contains the EJS templates for rendering the HTML content of the view and edit profile pages. These templates can include placeholders for dynamic data that is generated by the server-side code, such as the user's name and email.

```
const express = require("express"); // import express package
const bodyParser = require('body-parser'); // import body-parser package
const router = express.Router(); // create an instance of express router
const assert = require('assert'); // import assert package
const session = require('express-session'); // import express-session package

// listen to a get request on '/' route
router.get("/", (req, res) => {
    // Get profile details for the user with the specified user_id
    ...
});

// Define a route to handle the publication form submission
router.post('/', (req, res) => [
    var user_id = 1;
    // Extract the publication status from the request body
    const publication_status = req.body.publication_status;

    // Toggle the publication status in the database
    ...
]);

// This exports the router as a module so that it can be used in other parts of the application.
module.exports = router;
```

Code for Route defined for editProfile ejs template

```

<div class="header">
  <div class="img-container">
    
  </div>
  <div class="content-container">
    <div class="bio-data">
      <div>
        <h1 data-id="1"><%= profileDetailRows.first_name %> <%= profileDetailRows.last_name %></h1>
        <h3 data-id="1"><%= profileDetailRows.job_title %></h3>
        <h4 data-id="1"><%= profileDetailRows.current_location %></h4>
      </div>
      <div>
        <h3 data-id="1"><%= profileDetailRows.email %></h3>
        <h3 style="margin: 8px 0" data-id="1"><%= profileDetailRows.phone_no %></h3>
      </div>
      <div>
        <!-- <button type="button" class="btn btn-warning btn-sm">Resume</button> -->
        <button type="button" class="btn btn-warning btn-sm" style="display: none;">
          <form action="" method="POST">
            <input type="hidden" name="publication_status" value="<%= profileDetailRows.status %>">
            <button type="submit" class="btn btn-warning btn-sm"><%= profileDetailRows.status === 'public' ? 'Unpublish' : 'Publish' %></button>
          </form>
        </button>
      </div>
    </div>
    <div class="about">
      <h2 style="margin-top: 2rem;">About MySelf</h2>
      <p data-id="1"><%= profileDetailRows.about %></p>
    </div>
  </div>
</div>

```

Code snippet from editProfile.ejs template

The interaction between the Express.js application and the SQLite database is a crucial component of our project. We have used the SQLite3 driver to establish a connection to the database in our application code.

In our codebase, we have an *index.js* file that handles the database connection for *viewProfile* and *editProfile* pages and exports a function to query the database. This file is stored in the *jsServer* folder of our project.

```

// Open a connection to the database and store the connection in the global namespace
// creating a connection to the database
global.db = new sqlite3.Database('../database.db',function(err){
  if(err){
    console.error(err);
    // exiting the process if connection cannot be established
    process.exit(1);
  }else{
    // logging a message if database connection is successful
    console.log("Database connected");
    // setting foreign key constraints in SQLite
    global.db.run("PRAGMA foreign_keys=ON");
  }
});

```

Opening connection to SQLite3 database for the Express.js part of application

To interact with the database, we have written SQL queries to retrieve data from the database or to insert, update, or delete data from the database. We use the sqlite3 module in Node.js to execute SQL queries against the SQLite database.

```
// Get all project details for the user
global.db.all("SELECT * FROM projectDetails WHERE user_id = ?", [user_id], function (err, projectRows) {
  if (err) {
    next(err);
  } else {
    // Render the edit profile page with all the retrieved details
    res.render("editProfile.ejs", {profileDetailRows: profileDetails, skillDetailRows: skillRows, awardDetailRows : });
  }
});
```

Example of SQLite3 query from the editProfile Route

In the front-end of our application, we use the fetch API to make HTTP requests to our server-side code. Our server-side code then executes the SQL queries against the database and returns the results to the front-end as a JSON response.

We have used the EJS templating engine to render the views for our front-end pages. The data stored in the SQLite database is retrieved using SQL queries and passed to the EJS templates for rendering.

```
<% skillDetailRows.forEach(function(row) { %>
<div class="col">
  <div class="card h-100">
    <div class="card-body">
      <div class="icons">
        <a href="/edit-skill?skill_id=<%= row.skill_id %>"><i class="fa-solid fa-pencil" style="margin-right:<br/><i class="fa fa-eye<% if (row.visibility !== 'public') { %> visibility-off<% } %>" aria-hidden="true"></i></a>
      </div>
      <h5 class="card-title " data-id="1"><%= row.skill_name %></h5>
      <p class="card-text " data-id="1"><%= row.summary %></p>
      <p class="card-text " data-id="1"><%= row.more_detail %></p>
    </div>
  </div>
</div>
<% }); %>
```

Fig 5.8: Extracting data from JSON response using ejs syntax in editProfile.ejs

In summary, the interaction between the Express.js application and the SQLite database involves establishing a connection to the database using the SQLite3 driver, writing SQL queries to interact with the data in the database, and sending and receiving data between the front-end and the back-end of the application using HTTP requests and responses. This approach has enabled us to build a robust and scalable platform that meets the needs of our target audience.

Modules

This list shows the requirements to build and run Express API Server. The installation has been tested under macOS and Windows 10 / 11(most recent builds).

Package Name	Description
console	Built-in module to clear the console before logging info messages and requests
express	Minimal web framework that provides APIs for HTTP utility methods
body-parser	Middleware that parses incoming requests and information available under req.body attribute
multer	Middleware for handling multipart/form-data, which is primarily used for uploading files.
path	Built-in Node.js module that provides utilities for working with file and directory paths.
fs	Built-in Node.js module that provides utilities for interacting with the file system.
express-session	Provides middleware for managing session data in Express.js applications.
sqlite3	Provides a lightweight implementation of the SQL database engine, SQLite, in Node.js.
assert	Built-in Node.js module that provides a simple set of assertion functions for verifying the correctness of values and behaviors in code.

Table 5.1: Modules needed to run the ExpressJS API Server

Endpoints for Express API Server

API Endpoint	Method	Description	Arguments	API Key
/edit	GET	Renders editProfile.ejs template with	—	True

		data fetched from database		
/edit	POST	Handles the route of publication status of the user's profile and updates the info in the database	Publication_status (String)	True
/view	GET	Renders editProfile ejs template with data fetched from database	-	True
/edit/edit-header	GET	Renders editProfile Header form autofill with data fetched from database	-	True
/edit/edit-header	POST	Handles the process of updating the database with edited information	Image(Octet-Stream) First Name(String) Last Name(String) Job Title(String) Location(String)	True
/edit/edit-about	GET	Renders editMoreAbout form autofill with data fetched from database	-	True
/edit/edit-about	POST	Handles the process of updating the database with edited information	more_about(String)	True



/edit/edit-skill	GET	Renders editSkill form autofill with data fetched from database	skill_id(JSON number)	True
/edit/edit-skill	POST	Handles the process of updating the database with edited information	skill_id(JSON number) skill_name(String) summary(String) more_detail(String) make_public(String)	True
/edit/add-skill	GET	Renders the addSkill form	–	True
/edit/add-skill	POST	Adds newly created skill data into the database	skill_name(String) summary(String) more_detail(String) make_public(String)	True
/edit/delete-skill	GET	Removes the corresponding skill data from the database	skill_id(JSON number)	True

Endpoints of Express API Server

PHP API Server

About the php part we have chosen to use an online web host which is called 000webhost. 000webhost is a free web hosting service provider that offers users a platform to host their website for free. It offers various features like website builders, FTP access, and MySQL databases.

One of the primary reasons to use 000webhost is that it is free of cost, which makes it an excellent option for us who want to develop our website. Additionally, it offers unlimited bandwidth and disk space, which means we can upload and store as much content as we want. Connection to SQL

Connection to phpmyadmin

To connect PHP to MySQL, we use mysqli. This provides a set of functions or methods that allow us to establish a connection to a MySQL database, execute queries, and fetch results.

Here's an example of how we using the mysqli extension to connect to our MySQL database:

```
<?php
$servername = "localhost";
$username = "id20320564_talenthub_user";
$password = "hym8*uGw5n<3Xb-N";
$dbname = "id20320564_talenthub";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// echo "Connected successfully";
?>
```

Connection to sql

In this code, we define the configuration variables for the database, including the host name, username, password, and database name. Then we create a new mysqli object using these variables. If the connection is successful, we can start executing queries against the database.

We use conn.php file (short for "connection.php") to establish a connection to a database. This file contains configuration variables that define the connection details, such as the hostname, database name, username, and password.

Why we use conn.php:

1. **Centralization:** By placing all of the database connection details in a single file, it becomes easier to manage and modify the connection settings.
2. **Reusability:** Once the conn.php file is created, it can be included in any PHP script that needs to connect to the database. This eliminates the need to repeat the connection code in every file that needs to connect to the database.
3. **Security:** By keeping the connection details in a separate file, we can prevent unauthorized access to the database. We can set permissions on the conn.php file to restrict access only to those who need it.

Overall, the conn.php file is an important part of our website as it allows for a centralized and reusable way to establish a connection to a database. By using this file, we can improve the security and maintainability of our PHP code.

Index.php

```
<?php
if (isset($_COOKIE['user'])) {
    header('Location: /phpServer/Home-page.php');
    exit;
}

require "conn.php";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $password = $_POST["password"];

    $checkLoginQuery = $conn->query("SELECT * FROM loginDetails WHERE email = '$email'");
    $numRow = mysqli_num_rows($checkLoginQuery);

    if ($numRow == 1) {
        $emailError = false;
        $user = mysqli_fetch_assoc($checkLoginQuery);
        if (password_verify($password, $user["password"])) {
            $passwordError = false;
            setcookie('user', $user["user_id"], time() + 86400 * 30, '/');
            header("Location: Home-page.php");
            exit;
        } else {
            $passwordError = true;
            $passwordErrorMessage = "Password Incorrect";
        }
    } else {
        $emailError = true;
        $emailErrorMessage = "Email does not exist.";
    }
}
?>
```

Partial logic of how php handle log in

index.php is a file that serves as the default landing page for many websites and the same in 000webhost. It is typically the first page that users see when they navigate to a website's URL. In the case of a website with a login feature, the content of index.php is a login form that allows users to enter their login credentials and access our website's features.

The login form on the index.php page includes fields for the user's email and password. Once the user enters their login information and clicks the "submit" button, the form data is sent to a server-side script that verifies the user's credentials and grants access to our website if they are valid.

Overall, the content of the index.php file plays a crucial role in the user experience of our website, as it is the first page that users encounter.

Cookies

Cookies are small text files that are stored on a user's device by a website when the user visits it. Cookies contain information that allows our website to remember our user and their preferences, making it easier to provide a personalized experience.

In our website, cookies can be used to keep a user logged in until they explicitly choose to log out. When a user logs in, the website can create a cookie that stores a unique identifier (user_id) that is associated with the user's account. This cookie is then sent back to the website on subsequent requests, allowing our website to recognize the user and keep them logged in.

```
<?php  
    // Set the cookie expiration time to a past date  
    setcookie('user', '', time() - 3600, '/');  
  
    // Redirect to the login page  
    header('Location: Landing-page-or-Login-page.php');  
    exit;  
?>
```

delete cookies to log out

To log a user out of our website, we simply delete the cookie that was created when the user logged in. This will prevent our website from recognizing the user on subsequent requests, effectively logging them out. In PHP, this can be accomplished using the "setcookie()" function with an expiration time set in the past, which will cause the cookie to be deleted.



Browser Support

Browser	Version
IE(Internet Explorer)	Not Supported
Edge(Microsoft Edge)	79 - 110
Firefox	67 - 110
Chrome	67 - 111
Safari	11.1 - 14.1
Opera	50 - 95

Supported browsers for this application

Recommended

The web application has been tested in different browsers using popular Operating Systems.

Browsers	Operating Systems	Versions
Edge(Microsoft Edge)	macOS, Windows 10/11	79 - 110
Firefox	macOS, Windows 10/11	67 - 110
Chrome	macOS, iOS & iPadOS, Windows 10/11	67 - 111
Safari	macOS, iOS & iPadOS	11.1 - 14.1

Recommended Browsers for this Application

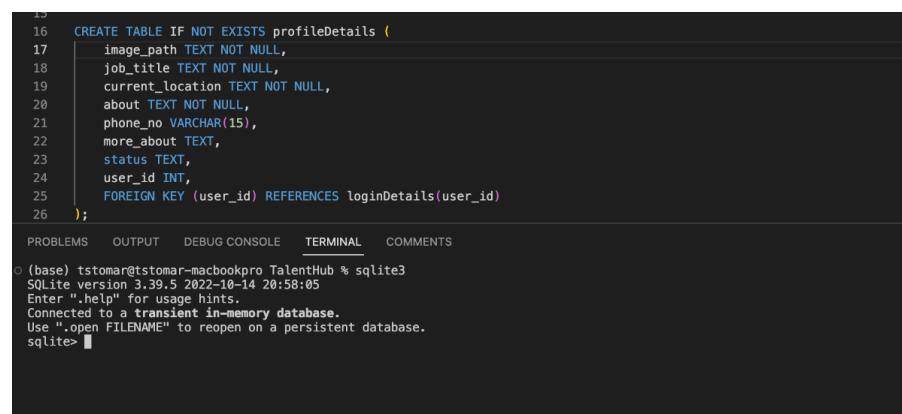


SQLite3

SQLite3 is a free, open-source, lightweight relational database management system (RDBMS) that is widely used in many applications. As a team, we recently used SQLite3 for our project's database management needs and found it to be an excellent choice. Here are some of the advantages we experienced:

- **Lightweight:** SQLite3 is a lightweight database management system that is easy to use and doesn't require any special setup or administration. This made it a great choice for our project, as we needed something that was simple and efficient.
- **Portable:** SQLite3 is also highly portable, meaning that it can be used on a wide range of devices and operating systems. This was important for our team, as we needed to ensure that our project could be accessed and used by all team members, regardless of their devices or platforms.
- **Low Maintenance:** SQLite3's self-contained nature means that it doesn't require any maintenance or administration. This made it a low-maintenance solution for our project, allowing us to focus our time and resources on other areas of development.
- **Fast and Efficient:** SQLite3 is known for its speed and efficiency, allowing for quick database access and retrieval. This was important for our project, as we needed to be able to retrieve data quickly and efficiently.
- **Free and Open-Source:** Finally, SQLite3 is a free and open-source software, meaning that it can be used and distributed without any licensing fees. This made it a cost-effective solution for our project and allowed us to allocate our resources to other areas of development.

Using SQLite3 for our project's database management needs was a great choice. Its lightweight, portable, low-maintenance, fast, and free and open-source nature made it a highly efficient and cost-effective solution for our team. We would definitely use it again in the future for similar projects.



```
15
16 CREATE TABLE IF NOT EXISTS profileDetails (
17     image_path TEXT NOT NULL,
18     job_title TEXT NOT NULL,
19     current_location TEXT NOT NULL,
20     about TEXT NOT NULL,
21     phone_no VARCHAR(15),
22     more_about TEXT,
23     status TEXT,
24     user_id INT,
25     FOREIGN KEY (user_id) REFERENCES loginDetails(user_id)
26 );
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    COMMENTS
○ (base) tstomar@tstomar-macbookpro:~/TalentHub % sqlite3
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> 
```

SQLite3 Database Schema

BootStrap



Bootstrap is a free and open-source front-end development framework that allows developers to quickly build responsive and mobile-first web pages. It is based on HTML, CSS, and JavaScript and includes pre-built components, such as buttons, forms, modals, navigation bars, and more.

As a team, we used Bootstrap for this project and found it to be extremely beneficial. Some of the advantages we experienced:

- **Easy to Use:** Bootstrap was easy to use, even for team members who didn't have a lot of experience with front-end development. Its intuitive syntax and pre-built components allowed us to quickly create complex layouts and designs.
- **Responsive Design:** Bootstrap's responsive design capabilities were particularly useful for our project, as we needed to ensure that our website was accessible and functional on a variety of devices.
- **Consistent Design:** Bootstrap's consistent design language helped us maintain a cohesive visual identity throughout our website. This was especially important as we had several team members working on different sections of the website.
- **Customizable:** Despite its pre-built components, Bootstrap was also highly customizable. We were able to modify the CSS and JavaScript to create a unique look and feel that fit our project's specific needs.
- **Large Community:** The large and active community of developers using Bootstrap provided us with a wealth of resources and support. We were able to quickly find solutions to any issues we encountered and benefited from the expertise of other developers.

Using Bootstrap for our project helped us save time and ensured that our website had a professional and consistent design. Its ease of use and responsive design capabilities made it a great choice for our team, and we would definitely use it again in the future.

```
<div align="center">Your Dream Hol  
    <p class="intro">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis  
    <table class="btn btn-primary btn-lg btn-pill" border="0" cellpadding="0" cells  
        <tbody>  
            <tr>  
                <td class="btn-content"><a href="#">book now</a></td>  
            </tr>  
        </tbody>  
    </table>  
</div>
```

Code Snippet showing Bootstrap classes to style the elements



Express

ExpressJS

As a team, we used ExpressJS for our web development project and found it to be a great choice.

Overview

ExpressJS is a popular web framework for Node.js that simplifies the process of building web applications. It provides a range of features that are commonly used in web development, such as routing, middleware, templating engines, and more. ExpressJS is built on top of Node.js, which means it inherits all of Node.js's features and modules.

Starting with ExpressJS

To get started with ExpressJS, we needed to install Node.js on our development machines. We then installed ExpressJS using npm (Node Package Manager), which is a package manager for Node.js. Once we had ExpressJS installed, we were able to create a new project and start building our application.

```
mkdir TalentHub  
cd TalentHub  
npm init
```

Setting up ExpressJS API Server

This created a new package.json file in the project directory, which contains information about the project and its dependencies.

Next, we installed Express.js as a dependency of the project using the following command.

```
npm install express --save
```

Installing ExpressJS package using npm

```
JS app.js > ...
1  const express = require('express');
2
3  const app = express();
4
5  app.get('/', (req, res) => {
6    |   res.send('Hello World!');
7  });
8
9  app.listen(3000, () => {
10  |   console.log('Server started on port 3000');
11});
12|
```

Initializing a new application in ExpressJS

This code creates a new Express.js application and defines a route for the root URL (""). When a GET request is made to this URL, the server responds with the message "Hello World!". Finally, the server starts listening on port 3000.

Advantages

- **Easy to Use:** ExpressJS is easy to use and provides a simple and intuitive syntax for building web applications. It allowed our team members, even those who were new to web development, to quickly get up to speed and start contributing to the project.
- **Flexible Routing:** ExpressJS provides a flexible routing system that allowed us to easily define routes and handle HTTP requests. This was particularly useful as our project had a lot of complex routing requirements.
- **Middleware:** ExpressJS's middleware feature allowed us to easily add functionality to our web application, such as logging, error handling, and authentication. This saved us a lot of time and made our code more modular and maintainable.
- **Large Community:** ExpressJS has a large and active community of developers, which meant that we had access to a wealth of resources and support. We were able to quickly find solutions to any issues we encountered and benefited from the expertise of other developers.
- **Template Engines:** ExpressJS supports a variety of template engines, such as Handlebars and EJS, which allows us to easily render dynamic content on our web pages.

Using ExpressJS for our web development project helped us save time and ensured that our web application was scalable and maintainable. Its easy-to-use syntax, flexible

routing, middleware, large community, and template engine support made it a great choice for our team, and we would definitely use it again in the future.

```
// including modules in the app
const express = require('express'); // importing express module for creating the server
const bodyParser = require('body-parser'); // importing body-parser middleware for parsing incoming request bodies
const port = 3000; // defining the port number on which the server listens
const multer = require('multer'); // importing multer middleware for handling file uploads
const path = require('path'); // importing path module for working with file paths
const fs = require('fs'); // importing file system module for working with files
const session = require('express-session'); // importing express-session middleware for creating secure session
const sqlite3 = require('sqlite3').verbose(); // importing sqlite3 module for database connectivity

// creating an instance of express application
const app = express();

// using body-parser middleware to parse URL-encoded request bodies
app.use(bodyParser.urlencoded({ extended: true }));

// Serve static files from the 'public' directory
app.use(express.static('public'));

// setting the views directory for rendering ejs templates
app.set('views', [ __dirname + '/views']);

/**----- */
/* VIEW PROFILE PAGE */

// using viewProfileRoute for rendering the view profile page
const viewProfileRoute= require('./routes/viewProfile');
app.use('/view', viewProfileRoute);
```

Code Snippet from index.js file of Express API Server

PHP: Hypertext Preprocessor



Overview of PHP

PHP is a popular server-side scripting language that is widely used for developing dynamic and interactive web applications. It is an open-source language that is free to use and modify. Our team used PHP for developing our web application, and we found it to be an excellent choice for our project. During the development of our web application, we found several advantages of using PHP.

Advantages of PHP

- **Easy to learn and use:** PHP has a simple and easy-to-learn syntax that makes it easy for our team members to get started with web development using PHP.
- **Platform-independent:** PHP is platform-independent, which means that it can run on a variety of operating systems. This helped us ensure that our application would be accessible to users regardless of the platform they were using.
- **Large and active community:** PHP has a large and active community of developers and users who provide support and resources for developers of all skill levels. We found this community to be an invaluable resource for finding solutions to problems and getting advice on best practices.
- **Wide range of frameworks and tools:** PHP has a wide range of frameworks and tools available for web development. We used the Laravel framework for our project, which provided us with a robust set of tools for building our application quickly and efficiently.
- **High performance:** PHP is a high-performance language that can handle large volumes of traffic and requests. This made it suitable for building a scalable web application that could handle a large number of users and requests.
- **Built-in security features:** PHP has built-in security features that helped us ensure the security of our application. For example, PHP has built-in functions for sanitizing user input, preventing SQL injection attacks, and protecting against cross-site scripting (XSS) attacks.



We found PHP to be an excellent choice for our project. Its ease of use, platform independence, large community, wide range of frameworks and tools, high performance, and built-in security features made it an ideal language for building our web application.

SQL

Overview of SQL

Structured Query Language, or SQL, is a programming language used for managing and manipulating data in relational databases. SQL is used to perform tasks such as inserting, updating, deleting, and retrieving data from a database. It is widely used in various industries such as finance, healthcare, education, and e-commerce, among others.

Advantages of SQL

- **Simplicity and ease of use:** SQL has a simple syntax that is easy to understand and learn, making it accessible to users with little to no programming experience. Additionally, SQL is a standardized language, which means that it is consistent across different database management systems.
- **Ability to handle large amounts of data efficiently:** SQL databases are designed to handle large datasets and perform operations quickly and accurately, making it an ideal choice for businesses and organizations that deal with large amounts of data.
- **Offers strong security features to protect sensitive data:** It allows for secure access to data by limiting user access based on user roles and permissions. This makes SQL databases more secure than traditional file-based systems, which are more vulnerable to data breaches.
- **Provides a high degree of flexibility and scalability:** It allows users to add, modify, or remove data without affecting the entire database. Additionally, SQL databases can be easily scaled to handle growing amounts of data, making it a flexible solution for businesses and organizations of all sizes.

In summary, SQL is a powerful and versatile programming language that offers many advantages for managing and manipulating data. Its simplicity, efficiency, security, and scalability make it an ideal choice for businesses and organizations that deal with large amounts of data.



Phpmyadmin

Overview of phpmyadmin

phpMyAdmin is a popular open-source tool for managing MySQL databases. It is a web-based graphical user interface that allows users to perform a variety of database management tasks through a web browser.

Advantages of phpmyadmin

- **Easy installation:** phpMyAdmin is easy to install and configure, making it accessible to a wide range of users.
- **User-friendly interface:** The web-based interface of phpMyAdmin is user-friendly, making it easy to navigate and use for both novice and advanced users.
- **Manage database structure:** phpMyAdmin allows users to create, modify, and delete database tables, as well as manage the relationships between them.
- **Import and export data:** Users can easily import and export data from the database in a variety of formats, including CSV, SQL, and XML.
- **Execute SQL queries:** phpMyAdmin allows users to execute SQL queries directly from the interface, making it easy to modify and retrieve data from the database.
- **Manage user permissions:** Users can manage user permissions and control access to the database, allowing for better security and control.
- **Multi-language support:** phpMyAdmin supports multiple languages, making it accessible to a global audience.

Overall, phpMyAdmin is a powerful and flexible tool for managing MySQL databases that offers a variety of features and advantages. Its user-friendly interface, ease of installation, and robust functionality make it a popular choice for many users.

000webhost



000webhost is a free web hosting service that offers a range of features and advantages for users. Advantages of using 000webhost:

- **Free hosting:** 000webhost offers free hosting, making it accessible to users who don't want to pay for a hosting service.
- **Easy to use:** 000webhost has a user-friendly interface that makes it easy to create, manage, and publish websites.
- **Reliable uptime:** 000webhost guarantees a 99.9% uptime for users' websites.
- **Support for PHP and MySQL:** 000webhost supports PHP and MySQL, making it easy to develop dynamic websites and web applications.
- **Automatic backups:** 000webhost provides automatic backups of users' websites, ensuring that their data is always safe and secure.
- **Security features:** 000webhost offers a range of security features, including SSL certificates and firewall protection, to keep users' websites safe from hackers and malware.

Overall, 000webhost is a reliable and feature-rich hosting service that offers a range of benefits for users who are looking for a free hosting solution. Its user-friendly interface, one-click installer, and reliable uptime make it a popular choice for many users.

Analysis

TalentHub application stands out from other platforms in its comprehensive features and its focus on skills and project showcasing. It provides a customizable and user-friendly interface that enables users to showcase their creativity, making it an ideal platform for freelancers, artists, designers, and anyone else who wants to showcase their work in a professional manner.

Features

The TalentHub application provides an all-in-one platform for users to showcase their skills, experience, and projects in a user-friendly manner. The platform has a range of features that make it stand out from other platforms. It provides a highly customizable interface that allows users to add and organize different sections. The platform also allows users to showcase their skills and projects by adding multimedia content like images, videos, and links.

Furthermore, the platform provides search functionality that enables users to search for other users based on their skills, location, and other criteria. The easy-to-use dashboard also helps users to track their profile views, clicks, and other metrics.

Comparison to other platforms

When compared to LinkedIn, the TalentHub provides a more comprehensive platform for users to showcase their skills and projects. LinkedIn mainly focuses on work experience and professional connections, and while it allows users to add some multimedia content to their profiles, it does not provide as much customization as TalentHub. Additionally, TalentHub provides a search function that enables users to find other users based on specific skills, which LinkedIn does not have[6].

GlassDoor, on the other hand, is mainly focused on job search and company reviews and does not provide a platform for users to showcase their skills and projects. It is a useful platform for people looking for jobs, but it does not have the same comprehensive features as TalentHub[7].



SquareSpace is a website builder platform that provides a highly customizable interface, but it mainly focuses on website creation rather than providing a comprehensive platform for users to showcase their skills and projects. While users can add multimedia content and create custom designs, SquareSpace does not have the same focus on skills and project showcasing as TalentHub[8].

Evaluation

Process Evaluation

Agile Development Methodology

- The team employed an agile development methodology to ensure the TalentHub application was developed iteratively with continuous feedback from stakeholders.
- The iterative approach allowed the team to incorporate changes and updates as necessary and ensured that the final application met the needs of its intended users.
- The team adopted daily stand-up meetings, sprint reviews, and retrospectives to ensure that the application was on track and met the requirements of the stakeholders.

Challenges with Communication and Coordination

- There were some challenges with communication and coordination between team members, particularly when it came to integrating different technologies.
- Some team members were more comfortable with certain technologies than others, and this led to delays in some parts of the project.
- The team members had to dedicate time to learning the necessary technologies and tools to work collaboratively to mitigate these challenges.

Project Management Tools

- The team used Trello and GitHub to track progress and manage tasks.
- Trello was used for task tracking and organization, while GitHub was used for version control and collaboration.
- These tools allowed the team members to monitor their tasks, deadlines, and progress in real-time.
- However, there were some instances where tasks were not well-defined or assigned to the appropriate team member, which led to delays.

Technical Details

- The team utilized Git for version control, which allowed for easy collaboration and the ability to revert to previous versions of the codebase if necessary.

- 
- The team also used a continuous integration and deployment (CI/CD) pipeline, which enabled them to automate testing and deployment processes.
 - This helped to ensure that the code was always in a deployable state and that any issues were caught early in the development process.

Overall Evaluation

- The team worked collaboratively and effectively to deliver a high-quality application within the given timeframe.
- However, there are areas for improvement, such as enhancing communication and coordination between team members, and ensuring that tasks are well-defined and assigned appropriately to avoid delays in the future.
- Additionally, the team could explore using more automated testing and development tools to further streamline the development process.

Product Evaluation

Advantages

- **Customizable Interface:** One of the most significant advantages of the TalentHub application is its customizable interface. Users can add and organize different sections, showcase their skills and projects with multimedia content, and use the search functionality to find other users based on specific criteria. This level of customization makes the application highly user-friendly and allows users to present themselves in a professional and creative manner.
- **Comprehensive Features:** Another significant advantage of the TalentHub application is its comprehensive features. Unlike other platforms like LinkedIn and GlassDoor, the application focuses on showcasing skills and projects rather than just work experience or job searching. The platform's range of features enables users to have a more holistic representation of their abilities, which can be useful for freelancers, artists, designers, and anyone else who wants to showcase their work.
- **Easy-to-Use Dashboard:** The application's easy-to-use dashboard is another advantage that helps users to track their profile views, clicks, and other metrics. This feature can be helpful in evaluating the impact of their profile and identifying areas for improvement.



Disadvantages

- **Security Measures:** While the application has some security measures in place, such as password protection, it could benefit from more robust security measures such as two-factor authentication and data encryption. These additional measures would help protect user data and reduce the risk of unauthorized access to user accounts.
- **Personalization:** The application could be enhanced with more personalized recommendations and connections between users. This would improve the user experience by providing users with more relevant content and connections to other users based on their skills and interests.
- **Testing and Optimization:** The application could benefit from additional testing and optimization to ensure that it performs well across different devices and platforms. This would improve the user experience by making the application more accessible and easier to use.

Technical Evaluation

Using Express and PHP together in the application

The decision to use both Express and PHP in the application has its advantages and disadvantages. Using both technologies allows the team to leverage the strengths of each team member and distribute the workload more evenly. This can lead to faster development and better quality code as each team member works on what they are best at. Furthermore, using a combination of server-side languages and frameworks can make the application more flexible and adaptable to changing requirements.

However, using multiple technologies can also lead to compatibility issues and increase the complexity of the application. The team may need to ensure that the different technologies work together seamlessly and that there are no conflicts or issues that arise from using both Express and PHP. This could lead to more testing and development time to ensure the application performs as expected. Additionally, using multiple technologies can increase the learning curve for new team members who may not be familiar with both Express and PHP. This could lead to slower onboarding times and potentially slower development cycles. Furthermore, it can also make maintenance and bug fixing more challenging as developers need to be proficient in both technologies.

Using SQLite3 as the database management system

The decision to use SQLite3 as the database management system has its own set of advantages and disadvantages. SQLite3 is a lightweight, serverless, and self-contained database management system that is easy to use and maintain. It requires minimal setup and administration and is ideal for small to medium-sized applications. Additionally, it is compatible with most programming languages and operating systems, making it a versatile choice.

However, SQLite3 has limitations in terms of scalability and performance. It may not be suitable for applications with large volumes of data or high concurrency requirements. Furthermore, it may not be suitable for applications that require advanced features such as replication, clustering, or high availability. Finally, using SQLite3 may limit the application's ability to integrate with other databases or services that require a different database management system.

Further Development

There are several ideas for further development of the TalentHub application:

1. **Improve Security:** Enhancing the security of the application should be a top priority. The team could add more robust security measures such as two-factor authentication, data encryption, and other security features to protect user data from potential threats.
2. **Personalization:** The application could benefit from more personalized recommendations and connections between users. This can be achieved through machine learning algorithms that analyze user data and provide personalized recommendations based on their interests and skills.
3. **Optimization:** To ensure that the application performs well across different devices and platforms, the team could conduct more testing and optimization. This could involve streamlining the code, optimizing database queries, and improving page load times to provide a smoother user experience.
4. **Social Media Integration:** Integrating the application with social media platforms such as LinkedIn, Twitter, and Facebook can help users to connect with more people and expand their professional network.

- 
5. **Gamification:** The team could add gamification features to the application to make it more engaging and fun for users. For example, users could earn badges, points, or rewards for completing certain tasks or achieving specific milestones.
 6. **Collaboration:** To encourage collaboration between users, the team could add features such as project management tools, group messaging, and document sharing.
 7. **Mobile App:** Developing a mobile application version of TalentHub can enable users to access the platform from their smartphones, which could increase user engagement and make it easier to showcase their skills and experience.
 8. **Analytics:** Adding analytics features to the application can help users track their progress, monitor their engagement, and measure the impact of their projects. This can provide valuable insights for users looking to improve their skills and showcase their achievements to potential employers.
 9. **Coming Features:** Implement the features that are unable to apply currently. Included the Remember me and Forgot password features in the Sign in page, and the download feature in the Template page.
 10. **More Complete Profile Page:** We are unable to complete the Profile page as we expected, comparing the complete version of Profile page, the current version is unable to implement the header and the Resume button is able to link to the Template page.

User Guide

1. **Install XAMPP:** First, you need to download and install XAMPP on your system. XAMPP is a free and open-source cross-platform web server solution stack package that includes Apache HTTP server, MySQL database, and PHP programming language. You can download XAMPP from the official website and follow the installation process.
2. **Start Apache and MySQL:** Once XAMPP is installed, open the XAMPP control panel and start the Apache and MySQL services. You can do this by clicking the "Start" button next to each service.
3. **Download phpMyAdmin:** Next, you need to download the latest version of phpMyAdmin from the official website. Make sure to download the correct version that matches your PHP version.
4. **Extract phpMyAdmin:** After downloading the phpMyAdmin zip file, extract it to the "htdocs" folder in your XAMPP installation directory. This folder is usually located at "C:\xampp\htdocs" on Windows or "/Applications/XAMPP/htdocs/" on Mac.
5. **Rename phpMyAdmin folder:** Rename the extracted folder from "phpMyAdmin-x.x.x-all-languages" to "phpMyAdmin".
6. **Configure phpMyAdmin:** Open the "config.inc.php" file located in the "phpMyAdmin" folder with any text editor of your choice. Then, find the following line:


```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```

 and change it to:


```
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = '';
```
7. **Import a database:** Open your web browser and navigate to "<http://localhost/phpmyadmin/>". This will open the phpMyAdmin login page. Login

with the username "root" and leave the password field blank. Once you are logged in, click on the "Databases" tab and import our database into it.

8. **Move our file to htdocs:** Move our entire folder into the "htdocs" folder.
9. **Run our website:** Open your web browser and navigate to "http://localhost/". This will open our landing page which is the login page.

If you are not able to run our website here is the version that we put on webhost kindly refer to it : <https://talenthub-930.000webhostapp.com/>

Conclusion

The TalentHub application has successfully provided a platform for individuals to showcase their skills and experience in a user-friendly manner. By leveraging both Express and PHP, the team was able to balance the workload and distribute tasks according to each team member's strengths. This resulted in faster development and higher quality code. Furthermore, the use of SQLite3 as the database management system provided simplicity, reliability, and scalability.

The TalentHub application has the potential to impact a wide range of stakeholders, including job seekers, recruiters, and employers. By providing a customizable platform for individuals to showcase their skills and experience, the application can help job seekers stand out in a competitive job market. Recruiters and employers can benefit from the search functionality and personalized recommendations, which can help them identify and connect with top talent.

Overall, the TalentHub application has the potential to make a significant impact on the job market by providing a platform for individuals to showcase their skills and experience in a more effective and efficient manner. Future development could focus on enhancing security, personalization, and optimization to further enhance the application's value to its users.

Individual Reflection

During our project, I was responsible for the backend development of our website, which included the login and registration functions, cookies log out, and the home page. As a team member, I worked collaboratively with my peers to ensure that our project was completed on time and to the best of our abilities.

In terms of my role, I worked diligently to ensure that my portion of the project was completed to a high standard. I was able to leverage my skills and experience in backend development to create a robust and secure login and registration system, as well as a user-friendly home page that included a search function. I also made sure to communicate regularly with my teammates to ensure that our work was coordinated and that everyone was aware of any updates or changes to the project.

Working in a team was both challenging and rewarding. One of the main challenges we faced was coordinating our efforts, as everyone had different schedules and working styles. However, we were able to overcome this by setting up regular meetings and communicating regularly through online platforms. This helped us to stay on track and ensure that everyone was working towards the same goal.

Another challenge we faced was managing our workload effectively. As we were all working on different parts of the project, it was important to make sure that everyone was making progress at the same pace. We were able to overcome this challenge by setting deadlines and milestones, which helped us to stay focused and ensure that we were making progress at a consistent rate.

In conclusion, working on this project was a great learning experience for me. I was able to contribute my skills and expertise in backend development to the project, while also learning from my teammates and gaining valuable experience working collaboratively. While we faced some challenges along the way, we were able to overcome them through effective communication and coordination. Overall, I am proud of the work we were able to accomplish as a team, and I look forward to future opportunities to collaborate and grow as a developer.

Reference

1. MarketsandMarkets™ , (Jul 2021), *Digital Identity Solutions Market - Global Forecast to 2026*,
<https://f.hubspotusercontent00.net/hubfs/4018359/GrowthNatives/Mar2021/PDF/Digital%20Identity%20Solutions%20Market%20-%20Global%20Forecast%20to%202026.pdf>
2. Nicepage, (2023), <https://nicepage.com/>
3. K. Takahshi and K. Watanabe, (2015) "Effects of image blur on visual perception and affective response," 2015 7th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 2015, pp. 169-172, doi: 10.1109/KST.2015.7051480, <https://ieeexplore.ieee.org/document/7051480>
4. Koshal, (August 5, 2011), *Difference Between First Name and Last Name*,
<https://www.differencebetween.com/difference-between-first-name-and-vs-last-name/>
5. Ricoh, (July 20, 2022), *8 Benefits of Data Protection and Privacy Services to a Business*,
<https://www.ricoh.com.my/blogs/how-a-business-can-benefit-from-data-protection-and-privacy-services#:~:text=Prevent%20data%20breaches%20that%20cause.transferring%2C%20sharing%2C%20and%20disposal.>
6. Wikipedia, (2023), [Online], LinkedIn, [LinkedIn - Wikipedia](#)
7. Wikipedia, (2023), [Online], Squarespace, [Squarespace - Wikipedia](#)
8. Wikipedia, (2023), [Online], Glassdoor, [Glassdoor - Wikipedia](#)
9. UsabilityGeek, Nathon Thomas
<https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>
10. WeareColorBlind,
<https://wearecolorblind.com/resources/color-oracle-color-blindness-simulator/#:~:text=Color%20Oracle%20takes%20the%20guesswork,of%20the%20software%20in%20use.>