Team Product Submission

Ctrl Alt Elite

University of Melbourne

COMP30022 IT Project, Semester 2, 2020

Repository: https://github.com/djatom17/e-portfolio

Deployed Product: https://e-port-folio.herokuapp.com/

Contents

Requirements

The product was developed and inspired around the requirements of our clients and throughout the semester, we at Team Ctrl-Alt-Elite managed to fulfill those requirements.

Our client team consists of four Master's students from the University of Melbourne:

1. *Shiyi Xu*
2. *Bokai YI*
3. *JiangChen ZHU*
4. *Yuchen Wu*

Date	Client Requirements	Implementation/ Deliverables
22 AUG 2020	Web application based e-portfolio system	The product is hosted and deployed on e-port-folio.herokuapp.com. The website is free and relatively simple to use.
	Individual profile page for each clients	Upon registration, a user is directed to their myprofile page
	Upload for image and media file	A user can select an image from their drive and upload for customise profile Image under 'profile image'. As well as upload any kind of media file on attribute showcase through 'edit' and 'upload'.
	Admin privileges for users to customise their specific/allocated portfolios	Upon successful login and given a valid session token, users would be shown an 'admin' version (with Settings and switching between 'View' or 'Edit' mode) of their own portfolio profile. In 'Settings': Change custom profile URL Change login email and password Change profile layout/ colour themes Change name In 'Edit mode': Upload/ Remove documents Change profile picture Modify descriptions Add/ Remove achievements, and skills Edit contact details Add/ Modify/ Remove education and career
	Link to external social media aspects	A user is able to add social media icons

	like facebook, linkedin, myspace and pictochat	under ‘edit’, and click to enable the wanted ‘social media icon’ and enter the link address for other users to view.
	Login page	Each user has been given their own login credentials with a default password for first-time login. The users were requested to change their password under ‘Settings’ upon first-time login. Each user credential is linked to his/her portfolio profile.
29 AUG 2020	Customisable portfolio (layout, colour theme)	When logged in, a user can change how their profile will be displayed via ‘Settings’. Users can choose between 4 profile layout templates and between 5 different colour themes, giving it a much more personalised feel to their own profiles.
13 SEP 2020	Able to browse other users’ portfolio	A ‘Browse’ page is linked from the header of the website. The ‘Browse’ page shows all the users that are currently “flexing” with us. A summary of user information, skills, career and education would be shown for each entry on “Browse”. When clicking on a user entry on the ‘Browse’ page, they would be redirected to that user’s portfolio profile (‘Admin’ mode if linked to the user's own profile).
	Lookup users based on tags/ skills under browse page	In the ‘Browse’ page, a search bar is implemented with an ‘Advanced Search’ feature. The search bar is able to lookup users based on name. With ‘Advanced Search’ enabled, users can lookup other users by the given skills tag query and/ or given user’s name query.
	Additional project and certificate showcase section under user profile	‘Projects’ and ‘Certificates’ tab has been added to each user profile.

	(with description and download link)	Users can upload/edit/remove projects and certificates via ‘Edit mode’. Visitors to the user’s profile can view/download the projects and certificates.
26 SEP 2020	More aesthetic/sleek user interface	Initial UI and website look has been re-hauled and improved by giving it a better colour scheme and design. More profile layout templates have been added and added a colour scheme feature to change profile colour theme.

Extra Project Deliverables

These are the deliverables created by our team independent from our client request in aims of polishing and in giving our users a better experience.

Date	**Description**	**Implementation/ Deliverables**
23 SEP 2020	About page	‘About’ and be found on the navibar. It contains a short introductory paragraph in aims of aiding our user.
25 SEP 2020	Registration	Registration can be found under ‘sign up’ on the navibar. Here a user can enter in their username, email address and password to create their account.
27 SEP 2020	Sign in with Google	In the ‘Login’ page, in addition to logging in with registered credentials, users can opt to login via Google. The authenticated Gmail used must be registered with Elite Flex. Google’s API was imported as a library and integrated into our backend.
2 OCT 2020	Copy email to clipboard feature	In ‘Browse’ under ‘user info card’, a visitor can click the ‘envelope’ icon to copy the user's designated contact information.
3 OCT 2020	In web profile image editor	After Login, under ‘myprofile’ page in ‘edit’, a user can select profile image icon. Upon selecting and uploading the

		desired image, a user is optionally introduced to our custom image editor where users can choose to zoom or rotate their uploaded profile image.
3 OCT 2020	Customisable URL	Under navbar ‘user’ -> ‘settings’ after login, user can navigate to ‘Profile settings’ -> ‘Custom URL’ inorder to set custom URL to their liking
3 OCT 2020	Change Password	Under navbar ‘user’ -> ‘settings’ after login, user can navigate to ‘User settings’ -> change account password

Testing

We have tested our product using various methods, to ensure that certain aspects of the product are consistently working as intended. As we have used automated testing frameworks (outlined below), we are able to continuously test the product on every deployment (next section).

Backend

Framework

The testing of the backend Express server for our app is based on two popular testing libraries/modules, Mocha and Chai. Mocha is generally used for asynchronous testing, and it is a JavaScript framework running on Node.js. Chai is simply an assertion library that can be paired with any testing framework, to allow tests to feel more like a “natural” language.

Tests

Backend tests involve asynchronous requests to the Express server, which are then fulfilled by responses and status codes. Any and every request made to the server must have each of those elements to be considered “complete”.

As most of our product’s backend is heavily dependent on database tasks, we mostly tested routes pertaining to our MongoDB Atlas instance. Testing of the database is only performed on a separate testing database, so that the production database is not affected by any CRUD operations during testing. The other tests were regarding authorisation, and validity of the auth-tokens being used to perform actions on the personal profiles. All these tests are considered to be comprehensive unit tests.

Frontend

Framework

We used the popular Jest library along with Enzyme for testing our Frontend. Jest is a unit test framework whereas Enzyme allows these tests to be written using regular assertions (Similar to Chai described above).

Testing

All our smaller components were tested to see whether they are rendered as intended. This was done via mock functions. Mock functions essentially isolate these components and allow test-time configuration of return values.This allows us to set up scenarios and provide dummy values to these functions/components and test how they work or how they are rendered in such scenarios.

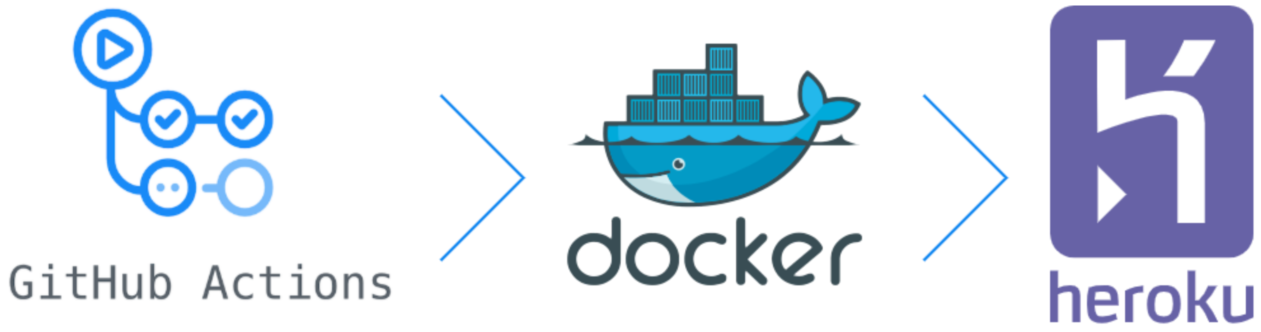
We also used snapshot testing to test one of our layouts. Snapshot tests are a very useful tool whenever you want to make sure your UI does not change unexpectedly. A typical snapshot test case renders a UI component, takes a snapshot, then compares it to a reference snapshot file stored alongside the test.This helped us test whether the changes we made during development were intentional and helped catch errors.

End-to-end

Throughout the development process, we would continuously perform end-to-end testing of the features that we implemented - such as editing portfolio information, changing password, uploading a file, etc. This was done so that the experience of performing those tasks replicated generic user flow - it was usually tested out by a developer who had not worked on that specific feature.

Deployment

The deployment process for the product has been streamlined as much as possible to keep the development environment as comfortable as possible.



Active Deployment

We have deployed our product using Heroku's free services at the following link: https://e-port-folio.herokuapp.com/. Note that the site will need to "restore" its state after 30 minutes of inactivity due to the free limits, therefore the first load of the page will take longer than the subsequent ones - and is thus, unavoidable.

Pipeline Steps

The deployment pipeline that is in place is straightforward - it tests the code, and on success, it creates a Docker container of the product, and deploys that to Heroku. This pipeline is facilitated by GitHub Actions' workflows, found in the **/.github/workflows/node.js.yml** file on the repository. The steps in the pipeline are defined as follows:

- Running the tests on Node versions 12 and 14, to ensure compatibility across supported versions.
    - Retrieving the cache of the Node modules/packages used by the app - mainly done to save fresh install times. This cache is renewed in the next step only if the **/package-lock.json** or **/server/package-lock.json** files are modified.
    - Running the `npm install` command - in both the root folder and the **/server/** directory, to make sure all the modules required are present.
    - Testing the React app, and the Express server using the written tests - the tests are written in **/src/App.test.js** and **/server/test/main.js** respectively.
- Once the tests pass, the next step in the pipeline is started, which is the deployment phase.
    - Setting up the Docker environment by initiating Docker Buildx - this is required to facilitate subsequent steps involving Docker actions.
    - Caching the Docker container layers - Docker images are created as layers. What this means is that each "layer" of an application (server, frontend, etc.) can be contained within the same Docker image. This is done so that in the rare case

that the app or the server has not changed on master (updated tests, new README, etc.), then the “new” deployment is just the previous one.
    - Logging in to the Heroku Container Registry - Done using the email used to sign up for the application, and the API key that is provided. This API key is obtained using the Heroku CLI with the command `heroku authorizations:create -a <app name>`. This can be performed after logging into the CLI using the same credentials as the one used for creating the application.
    - Building the Docker container and pushing it to Heroku - This step is conveniently performed in one go, thanks to the official GitHub Action docker/build-push-action@v2. It uses **/Dockerfile** to build the image, and then tags that image with `registry.heroku.com/<app name>/web`. This tag is created so that when the image is pushed to the Container Registry, Heroku knows where the image is supposed to go. “Web” just defines the service that Heroku should use - since we are on the free quota of Heroku, we only have access to the “web” service.
    - Releasing the app/image on Heroku - This command tells Heroku that the deployed app should now use the new container that has been uploaded with the required tag. This is done using `heroku container:release -a <app name> web`.

Pipeline Settings

It is run on two scenarios:

- When a pull request is made - where it only tests the code.
- When a pull request is merged to the master branch, or a commit is made directly to master - this tests the source code, and if successful, it deploys that version.

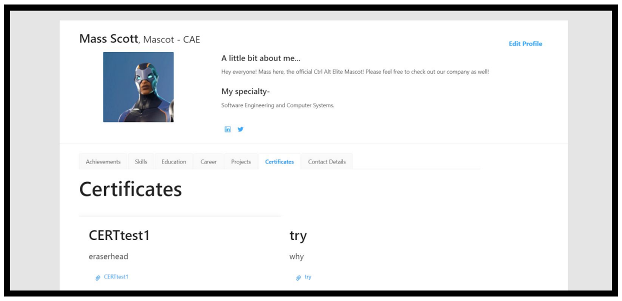
Our code involves using APIs for external online tools like AWS S3, MongoDB Atlas, etc., which means that our code requires secret API keys. Since these should never be stored on a public repository, these have been entered as environment variables on GitHub (as Secrets) and Heroku (as Config Vars). GitHub requires more information as it also requires access to the testing database, and the API key to Heroku itself for deployment.

Elite “Flex” Page

For our team, boasting is quintessential as our product is called Elite Flex e-Portfolio. In this section we will be showcasing the features that we worked hard on and are proud of!

Layouts

This semester, as per our clients’ requests, we created 4 distinct profile layouts. This ended up being almost 4 times the front-end throughput we expected. In addition to that, we added the functionality to seamlessly switch between these layouts. Our efforts definitely paid off as we received overwhelmingly positive feedback from our clients regarding the layout options!


Mass Scott, Mascot - CAE

A little bit about me...

Hey everyone! Mass here, the official Ctrl Alt Elite Mascot! Please feel free to check out our company as well!

My specialty-

Software Engineering and Computer Systems.

Certificates

CERTtest1

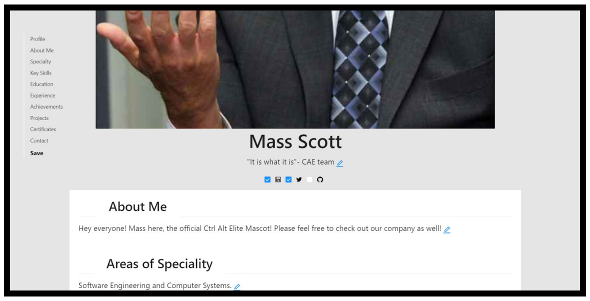
eraserhead

CERTtest1

try

why

try


Mass Scott

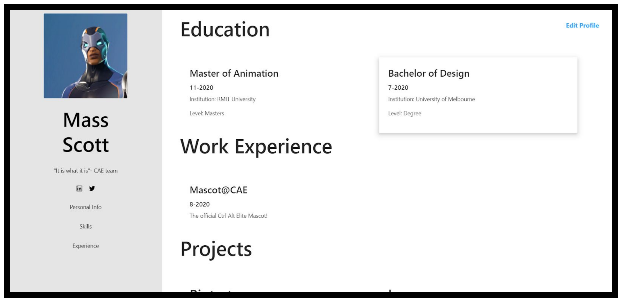
"It is what it is"- CAE team

About Me

Hey everyone! Mass here, the official Ctrl Alt Elite Mascot! Please feel free to check out our company as well!

Areas of Speciality

Software Engineering and Computer Systems.


Mass Scott

"It is what it is"- CAE team

Education

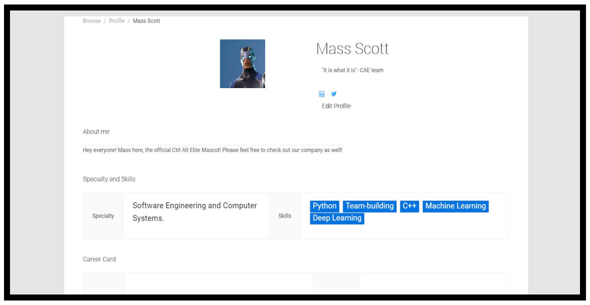
Master of Animation
11-2020
Institution: RMIT University
Level: Masters

Bachelor of Design
7-2020
Institution: University of Melbourne
Level: Degree

Work Experience

Mascot@CAE
8-2020
The official Ctrl Alt Elite Mascot!

Projects


Browse / Profile / Mass Scott

Mass Scott

"It is what it is"- CAE team

About me

Hey everyone! Mass here, the official Ctrl Alt Elite Mascot! Please feel free to check out our company as well!

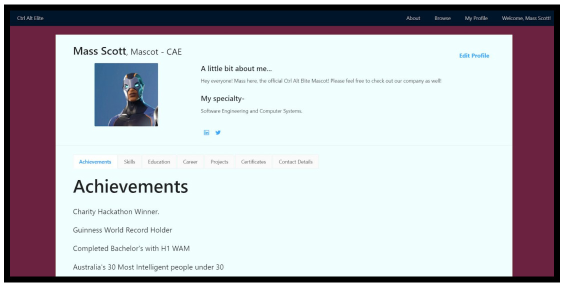
Specialty and Skills

Specialty	Software Engineering and Computer Systems.	Skills	Python Team-building C++ Machine Learning Deep Learning

Career Card

Customisability

For a better user experience, we have put in extra effort to give the users more power and control of their profiles. Users have the option to select a colour scheme from our 5 unique presets. They also have the options to change their account details, including- email, password and their customised URL. The customisable URL is a feature that we decided to add so that the users could proudly share a custom link to their e-portfolio. All the profile data is fully editable granting the user complete control. This way, users can create a portfolio that is truly theirs!



Mass Scott, Mascot - CAE

A little bit about me...

Hey everyone! Mass here, the official Ctrl Alt Elite Mascot! Please feel free to check out our company as well!

My specialty-

Software Engineering and Computer Systems.

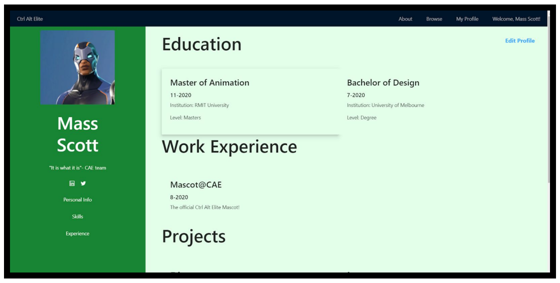
Achievements

Charity Hackathon Winner.

Guinness World Record Holder

Completed Bachelor's with H1 WAM

Australia's 30 Most Intelligent people under 30



Mass Scott

"It is what it is"- CAE team

Education

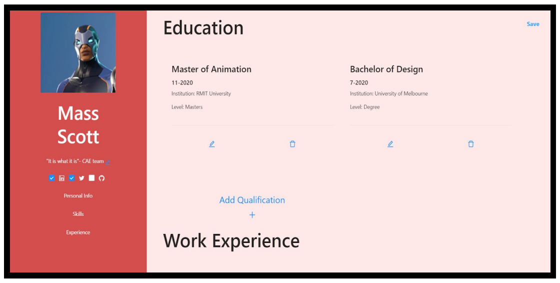
Master of Animation
11-2020
Institution: RMIT University
Level: Masters

Bachelor of Design
7-2020
Institution: University of Melbourne
Level: Degree

Work Experience

Mascot@CAE
8-2020
The official Ctrl Alt Elite Mascot!

Projects



Mass Scott

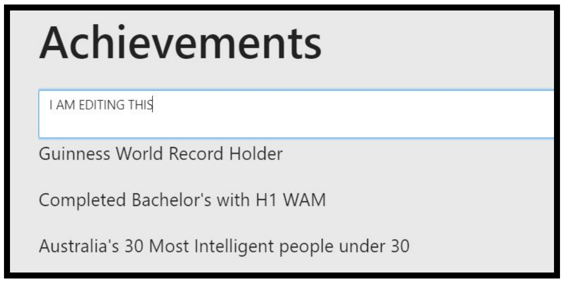
"It is what it is"- CAE team

Education

Master of Animation
11-2020
Institution: RMIT University
Level: Masters

Bachelor of Design
7-2020
Institution: University of Melbourne
Level: Degree

Work Experience



Achievements

I AM EDITING THIS

Guinness World Record Holder

Completed Bachelor's with H1 WAM

Australia's 30 Most Intelligent people under 30

Search

Another client requirement that we fulfilled successfully is the ability to search for a user. Our browse page features a search bar that by default allows the user to look up a person by name. Additionally, if the user wishes to look up a professional on our app by skills, they can do so by clicking on the advanced search button. With advanced search, the user can customise their search by name and skill.

The best part about this feature is that it is beautifully integrated with the rest of the page. The user can click on any skill tag on any profile card and it will trigger a search for the skill. This is one feature that is subtle but adds charm to our project.

Google Sign In

A feature we decided to add for convenience was the handy Google sign in button. With this feature, the user can securely log in with just a click of a button, provided they have signed into their Google account beforehand. Another added bonus of this feature is that the user does not have to recall their password to login in through this option.

Bonus: Register

This feature was not originally part of the client requirements or our sprint plans. However, our clients expressed interest in this feature after our original requirements were finalised. Therefore, we allocated a few work hours to add this bonus feature to impress our clients. We are proud of it as we managed to complete and include it in our project within a limited timeframe.