QCC Credential Approval Procedure

Description

QCC's software development training program awards credentials to deserving students based on the creation of online portfolios, through which they may showcase their skills and knowledge in computer programming to potential employers.

The software development online portfolio contains two parts: (1) a GitHub profile, where students can store and showcase their code, and (2) a personal webpage that visually presents the student and highlights their projects.

Attaining the credential

Students are expected to work towards building their online portfolio throughout the duration of the program. This is implemented in several ways:

- Students are introduced to GitHub and create their GitHub profiles in their first workshop.
- Instructors continually encourage students to push new code to GitHub, as it is created in class, workshops and special events.
- Course instructors require students to build a personal webpage in their web development class.
- Instructors continually encourage students to add new projects to their webpage, as they are created in class, workshops and special events.
- Students complete portfolio check-ins with the Tech Success Manager at least twice per semester, where she provides feedback to ensure that portfolios progress in line with program standards.
- At the end of every semester, students participate in a live presentation of their portfolios in front of an audience of colleague, instructors, TechWorks staff and other stakeholders.

Awarding the credential

Students are expected to receive their credential at the completion of training #1. At this point, they must to meet the minimum requirements by providing links to: (1) a GitHub profile that belongs to them and contains code, and (2) a live webpage that belongs to them and showcases work that they have completed. The Program Director will determine whether a student has met these requirements and award the credential when appropriate. If a student has not made the necessary progress towards completing their credential at the end of training #1, they will work with the Tech Success Manager to meet this requirement before the end of their program.

In the 90 day period following completion of training #1, most students will continue to make improvements to their portfolio so that it will be attractive to employers. That is, they will go from creating an acceptable portfolio to an excellent one in the period following coursework completion. If a student participates in an internship, they will add work from their internship to their portfolio (e.g., adding new code on GitHub or links to projects on their webpage). QCC's guidelines for what constitutes an excellent portfolio appear in Appendix A, and are made available to students from the beginning of their program.

On the 91st day, students will be exited from the program.

Archiving the credential

Online portfolios are live documents that will change as a student makes updates, and at times may be unavailable for viewing if there are issues on the backend. To mitigate this, QCC will be archiving student portfolios using the QCC TechWorks GitHub account. This allows us to preserve students' hard code and live portfolio sites in one central location.

Additional, QCC is creating a PDF of the front page of every student's GitHub profile and webpage, which will be held in a digital folder.

Appendix A: Guidelines for Portfolio Excellence

Github: Github should look clean, neat, with several repositories/projects in it so that your contribution levels go up. Remember recruiters want to see how active your Github profile is.

Examples:

1. Jefferson Huisa: https://github.com/j3ffh95

2. Bernado Gomez:https://github.com/Bernardo11rex

3. Bryan Peralta:https://github.com/bpcodewriter13

Webpage: Your webpage can either be a single page html, or multiple pages. You may choose the languages for your webpage, if you'd like to build it out in Rails, React, Python, etc; however we require it to be at least with the simple three of HTML/CSS/Javascript.

Additional requirements:

1. Layout: Is there a good flow to your portfolio from a visual standpoint.

2. Code: Are there any incomplete pages, broken images, 404 errors, broken code, etc.

3. About Section: Is there a section that tells us about you, what you are looking for, this should be a few sentences about you, and why you want to be in tech and/or a developer.

4. Experience Section: Is there a section about your work experience, any competitions, hackathons attended, volunteer work, group activity should go here, similar to the experience section in your resume.

5. Education Section: This section should contain any education you've done similar to what you put in your resume. You can put that you are currently a student and/or graduate of QCC TechWorks depending on what cohort you are part of. Any certificates, awards, etc should go here as well.

6. Portfolio Section: The portfolio section is similar to that of an artist's portfolio, this is where you can display any freelance projects, class assignments (like the Javascript and/or React projects), or any passion projects you are working on can go here. You can either take screenshots of these projects and put it in a gallery form with a hyperlink taking you to the actual site, or Github code. But it's good to have at least 2-3 projects (or more) to show tech recruiters/hiring managers/clients visually what you have worked on.

7. Interests Section: This section is more than what we see on a resume, tell me what interests you have like do you like to dance/travel/attend meetups (which ones), volunteer with non-profits, hobbies and/or sports you're into, tell me something cool about you!

8. Contact Section: This section is extremely important. It must contain your email, Github link, and LinkedIn link...the other social media is not as necessary but up-to you to add, but please don't connect your Facebook if employers are going to see something that would not want make them not want to hire you.

Examples

1. Jefferson Huisa: https://www.jeffhuisa.com/

2. Bernado Gomez: https://bernardo11rex.github.io/Porfolio2/index.html

3. Bryan Peralta: http://bprocodesign.com/

Additional considerations for webpage excellence:

1. **The mission or purpose is clear:** The landing page should explain who you are and what you do - or are looking to do, in a single paragraph. You have a picture of yourself in the landing page, or in the about page. Employers like to see whose site they're viewing.
2. **Site design is driven towards your audience:** The design of reflects your work, and shares a similar look and feel to rest of the site. If you have more than one page, (example: about page, contact page), your site should have continuity between pages - background, fonts, colors, pictures, formats should be the same throughout.
3. **Your personal brand social media is integrated:** Any social media that is related to your personal brand should be clearly linked on your site. These links should include at least Github, LinkedIn, and any blogs they've personally written. Avoid linking personal twitter and facebook accounts that are not directly related to your "brand".
4. **Personal, succinct wording:** You should write to show your personality. Employers like to see that you have expressed a sense of yourself into the copy of the site. You should set a pleasant tone, expressed in a short, simple, and to-the-point manner.
5. **Consideration given to the user experience:** The site should navigate well. If you have multiple pages, then these pages should be easily accessed, and navigation on each page should be clear. Links should be correctly colored or categorized, and linked to the correct URL.
6. **Personal work is categorized correctly:** Employers look at prospect employees' work to see if they have a strong attention of detail. The more organization they show throughout their site, the better display of this quality. You should create categories for each type of project you create (example: Games, Personal sites, Company sites, music site, web apps, mobile applications ect ect..). These projects should also have correct

links to the deployed application or correct URL to the linked site, or a link to the Github repo.

7. **The website is responsive:** The website should respond to multiple screen sizes and correctly align its content automatically. This allows people viewing on their phones and computer alike to view your site.