Software Engineering Quiz Game – Final Report

Synergy Simulator

Andrea Lee Hamilton

Elizabeth Ponce

CS467 Summer ‘22

Introduction

Team Synergy Simulator will be developing a Software Programming Quiz that provides testing for incoming employment candidates. The quiz will be similar to SurveyMonkey. It is free and open-source for employers to use.

Description – User Perspective

The application will have two different perspectives, one from an employer and the other from the candidate.

The employer creates a quiz and then emails a unique key link that authorizes a person to take the quiz. The app will auto-email the key link to the candidate. After the candidates complete the quiz, employers are able to see the quiz respondents ranked by top score, and can view individual candidate results and stats.

For candidates, they will go to the site and register. They will later be emailed with their unique quiz key. The candidate provides the program with their quiz key which unlocks their unique timed quiz. The quiz questions are related to software engineering and they are all multiple choice. Once the candidate completes the quiz, the employer receives an email.

Client

- William Pfeil - No NDA required

Client Requirements (from project site)

- Can be done as an open source project (except for the questions and answers!)
- Accessible by web browser for interaction via a desktop computer.

- Responsive design to work on any device.
- Replace SurveyMonkey with this application.

Three main interfaces:

- For the employer, to CRUD an account and profile and login
- For the employer, to create a quiz and then email a unique key link that authorizes a person to take the quiz. The app should auto-email the key link to the candidate.
- The employer should be able to configure the amount of time to take the assigned quiz.
- For candidates, given the key link, allows them to take the timed quiz.
- For the employer, to see the quiz respondents ranked, and also individual results and stats.
- Variable number of questions, True/False, multiple choice, check all that apply, free-form answers.
- User has a decided amount of time to complete the quiz after clicking the key link and confirming the start of the quiz.
- An email should be sent to the employer after a candidate has completed a quiz.

Development Efforts vs Project Plan

We stayed incredibly close to our initial project plan, with Andrea taking on our front end and application routes work and Elizabeth standing up the backend with database and Heroku configurations.

Notable deviations are as follows:

Quiz questions went from 4 styles to 1

The initial plan was for us to create a quiz that had a mix of 4 styles of questions for the candidate: T/F, Multiple Choice, Free Text Box, and Select All That Apply. We had outlined using the Quiz API for the questions generation portion and Elizabeth set to work on building out the templates for the different style questions and answers. The API ended up only supporting a Multiple Choice question option and we reached out to our Client who clarified that we only needed to implement one style of questions. We chose multiple choice as that was the natural fit for our API of choice.

Quiz Key vs quiz key link

The requirements stated that we needed to build a quiz key link that was emailed to the candidate. Our project's implementation of this solution included emailing a key (unique number) for the candidate to log in and access their quiz. This was confirmed as an acceptable solution by our Client.

Structure of the Software

We are hosting our quiz as a web browser application, with a responsive design for inclusive computing practices. The code is hosted in a private, shared git repository, and a SQLite database is used to store the employer information, candidate information, scores, quiz questions, and quiz answers.

External Technologies Used

- Python
- Flask for web and mobile applications
- Open Trivia Database quiz api
- Heroku cloud hosting
- SQLite database
- HTML/CSS

Team Member Notable Accomplishments

1) Andrea:
    a) Frontend Development
    b) Master quiz maker
    c) API wizard
    d) Project Lead
    e) Videographer
    f) Slide queen

2) Elizabeth:
    a) Backend Development - databases
    b) Systems Architect
    c) Github troubleshooting

d) Heroku configurer
e) Tech Support
f) Paper writer

Future Directions

For the future directions we'd love to see the project upgraded with additional accessibility features – availability for screen readers and alternative text for images. We'd like to include React JS and add in Material UI components for clean icons and website features.

References and Citations

1) "Software Programming Quiz Project Charter". Accessed June 26, 2022. [Online]. Available: https://eecs.oregonstate.edu/capstone/submission/pages/viewSingleProject.php?id=WXCwYpWHmG3lGdYq
2) "Flask/Python User Mixin Module Discussion post on StackOverflow ". Accessed July 31, 2022. [Online]. Available: https://stackoverflow.com/questions/63231163/what-is-the-usermixin-in-flask
3) Kariya, Mahesh. "Flask Login Tutorial". Accessed July 31, 2022. [Online]. Available: https://www.youtube.com/watch?v=kidix0toV1g&ab_channel=MaheshKariya
4) "Open Trivia Database". Accessed June 27, 2022. [Online]. https://opentdb.com/
5) Blago, Antonio. "Building a mobile/web Flask application". Accessed June 27, 2022. [Online]. Available: https://antonioblago.medium.com/build-your-own-mobile-friendly-website-with-flask-6fe208f029d9