Test your Knowledge

Kthan Graham

Dr. Mubarak Banisakher

21/FA-COP-2939-55638 Computer Programming Capstone  
11/26/2021

SUMMARY OF PROPOSAL:

This project was aimed to create a program that would allow me to create a quiz/test that would have programming questions that could easily be modified and expanded on. I wanted to create a structure that I could further expand and make into a full fledged website/app that would be able to tell you how well you know your programming language.

OUTPUTS:

The project produces a lot of useful information. The most important is how the questions and answers are saved in a file and in an array so you can easily manipulate them to grab data how you see fit. For example if you review each Script.Js you will see that each will have its own UserScore (for example you can see the javascript has JSuserscore. This allows us to review how they did and we can actually store the answers they got wrong and push them to a results page at a later date. This also helped me to develop a sense of how HTML CSS and javascript interact as this was my first time implementing them together.

EVALUATION:

Sadly, I did say I wanted to originally make this in python but that was not possible in the long term as discussed via email so this was the closest I could make it to the original idea. Achieved: I achieved a good data structure, Sound data manipulation, ease of access for the end user as well as for the developers that want to improve this. Short comings: Its in HTML CSS and Javascript, I wanted to add an extra piece where you would be able to review the previous incorrect results by test on the results page but sadly, I did not have enough time to add it.

Preliminary design:

System Design

**2.1 Application Design**

Diagram

Description automatically generated

**2.2 Testing framework**

The testing framework that I will be using is going to be the keyword driven testing framework. Because this is such a small project, I believe this framework is the best way to go about testing my different variables and functions.

**2.3 development/build environment**

I will be using maven for dependency management as it is the most widely used resource. Git will be used for version control because I have just recently learned how to use this and would like to incorporate it into a program.

Testing matrix:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Title | | | | |
| Students name | | | | |
| ***ID*** | ***Requirement*** | ***How to Validate*** | ***PASS*** | ***FAIL*** |
| \_ | **Unambiguous description of a feature/function that your project will provide.** | **Detailed explanation of how that feature/function will be tested or verified during the demonstration.** |  |  |
| 1 | the system will present tests in different diciplens of programming. | How: the tests will be hard coded and stored in a test file in the app. Test: I will run through the tests to cmake sure they all pull correctly as well as have a counter at the top. |  |  |
| 2 | The system shall store previous test result. | How: the app will store your previous attempt in folder 'past' to see if you have improved. Test: Run through the test multipule times to confirm that the information is being saved correctly with a print at the end. |  |  |
| 3 | A user shall be able to see a statistical break down | How: at the end of the test the student will receive a score with a full break down on what they missed with useful links to 3wschool sources. Test: Run through the tests and intentionally miss questions to confirm that it is working. |  |  |
| 4 | A user shall be some sort of programmer (new, intermidiate, or even advanced) | The main audience I am aiming for are mainly for fresh out of college students to help gauge where they land. But it will be able to be used by anyone. The main aim of this project is just to help improve programming skills. |  |  |
| 5 | A user shall be able to get links to useful resources | How: hard coded so that when you miss a question it reviews your answer to see why you missed it. Test: miss questions intentionally and see end result to see if knew why I missed it. |  |  |
| 6 | Amount of data stored must be under 500mb | The amount of data stored will be minimal as the only part being stored would be previous test results. |  |  |
| 7 | System must be the size of 2gb | The size of system files will be minimal as I will not be storeing to much data besides the previous test results and the graphics will minimal. |  |  |
| 8 | All sensitive data: there will be no sensitive data. | N/A |  |  |
| 9 | System must display data in a table at the end of a test. | At the end of your test results will show at the end of the test in a table listing a statistical break down of how you did. |  |  |
| 10 | Credentials are not needed so this does not apply | N/A |  |  |
| 11 | User must have connection to the internet | to download the application. |  |  |

Requirements:

1.0 Acronyms and/or Definitions

PHP General purpose scripting language especially suited to web development

CSS3 Cascading Style Sheets; language used to describe the presentation of a document written in markup language, e.g., HTML

HTML Hypertext Markup Language, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

JAVASCRIPT an object-oriented computer programming language commonly used to create interactive effects within web browsers.

BOOTSTRAP a technique of loading a program into a computer by means of a few initial instructions which enable the introduction of the rest of the program from an input device.

TYKSG This working group will be responsible for the administrative oversight, operations, and support of Test your Knowledge.

API a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

1.1 Administrative Support and Oversight

The group in charge of the operations from a support perspective will be the TYKSG.

The below tasks will be reviewed, update, maintained by the group;

* Develop new features to be implemented via feedback from the clients
* Review the already developed features and come up with new test cases as well as possible enhancements
* Provide updates to the application after the approval of new features
* Take in recommendations from clients to implement into the software
* Maintain current patches on the app

2.0 Current system requirements

This application will strictly be an android 11 application. There will be no log in requirement as the main users will be using it only a few times to check if they have improved or to see where they stand with that specific language. All data will be saved in their device to help reduce the strain on maintaining a server. This will be added in future updates and will store the answers in an SQL database then translate them to the app using PHP where it will then display a statistical break down of progress. The SQL database will be updated after each submit button and will only hold results from the past 365 days or the past 50 attempts to reduce lag time on the server.

3.0 Functional requirements

In order to get this application out in decent time frame I will be limiting it to having all the data housed on the Android device itself but will be implement a SQL server and housing the data in there and retrieving it with PHP which will be discussed below with a tag of (future update);

* **Data Accessibility**
  + Data will be managed on the device itself
  + Application will keep track of how often it is used
  + (future update) Data will be stored in a SQL database and be pulled using PHP and output into the application
  + (future update) Application data will be tied to a Facebook log in for ease of access
* **Profile Administration and user accounts**
  + As of now there will be no sign in feature for the application.
  + (future update) Clients must submit an account using the Facebook login API.
  + (future update) Once logged in the account will stay logged in on that device and require login credentials for any other devices.
  + (future update) any changes to account will be saved in an audit log for use of user confirmation.
  + (future update) once logged into by another device it will automatically kick you out of the previous device to help reduce confusion.
  + (future update) Users will be required to relog back in once every 6 months to reduce the amount of bots that might use the application.
* **System security**
  + The only security measures that will be taken would be to use hidden fields as this is just a pilot application but will be implementing PDO once the functionality of the application is confirmed.
* **Objective**
  + To review and describe where you stand in the ‘*real world*’ when it comes to your skill in creating, reviewing, and editing scripts/code.
* **End Result**
  + Test that very from different scripting and coding languages
    - Javascript
    - SQL
    - PHP
    - Java
    - SQL
    - C
    - C#
    - C++
* **Documentation**
  + The use cases would be as follows;
    - Individuals just starting programming to get a guide on where to state
    - Novice programmers to gauge how much they understand were they stand
    - Intermediate programmers, these are individuals that have finished or just finished their programming education at college and are looking to improve in ‘*real world*’ scenarios
    - Advanced programmers, these are individuals that have their ‘*real world*’ experience and looking to improve on a specific area.
* **Testing**
  + Testing will be done in the application to find bugs by designing the testing parameters first then creating the application
    - Examples: using integer vs using a string, bypassing questions by hitting the submit button with no answers, entering the answer using bootstrap vs Javascript, ect.

4.0 Non-FUNCTIONAL REQUIREMENTS

As seen below this will consist of Objective, end results, focus, documentation, and essentiality. Objective will describe how the product works. End result will define product properties. Focus will aim on the user expectations. Documentation will focus on the quality attributes. Essentiality will focus on the ‘nice to haves’.

* **Objective**
  + The product will work by providing the test and giving a statistical break down of the single test (what you missed) and providing documentation (W3 Schools) on what and how you can improve.
* **End result**
  + All properties of the application will be using HTML and JavaScript (bootstrap) to create a smooth clean GUI for the user to interact with.
* **Focus**
  + The user should expect the following from the program;
    - A clean interface
    - Ease of access to information
    - Tests on the various languages and scripts
    - Break down on what they missed
    - Documentation to help improve
* **Documentation**
  + The quality attributes will be done by testing the application (done by TYKSG) and confirming that the out put is correct.
    - This will consist of intentionally missing questions to gauge if the statistical break down is correct
  + Confirming if the provided documentation is correct if the question is missed
    - Intentionally missing questions to confirm the documentation
* **Essentiality**
  + Confirming that all bootstrap code is a clean and easy to use UI

*Test your Knowledge Pilot Phase 1*

* *Proposal Summary*

I propose to build an Android application that is designed to help all programming and scripting users improve their skills and give a generic gauge to help you understand where you fall on the spectrum of beginner, intermediate, and advanced.

* *Timeline*

1st phase will aim to be completed by the end of semester. This would involved have an alpha version of the application ready for use by end users.

* + Application downloading
  + Tests for the user to use
  + Break down of the tests

External learning resource

Proposal:

The purpose of this application for my capstone is to help gauge where you fall as far as your knowledge goes for different programming languages (both logically and for syntax. With the languages of ; C, C#, C++, CSS, Java, Javascript, PHP, SQL, and HTML as these are the only languages I have experience with) and give you a statistical break down on what you scored and why, it will also provide W3school links to resources to help you improve .The problem that follows entry level program specialists is that they have a hard time gauging their knowledge level and often fall victim to ‘imposter syndrome’. Imposter syndrome is defined as ‘refers to an internal experience of believing that you are not as competent as others perceive you to be’ ([reference](https://www.verywellmind.com/imposter-syndrome-and-social-anxiety-disorder-4156469)).

Please let me know if this does not meet your requirements as I spent a few days brainstorming and landed on a few different ideas (Name: new Music much? - a music app that is similar to Spotify but will not replay previously played songs to help you expand your music library, Name: It’s just facts - a visualizing programming puzzle game to help teach younger individuals to program, or Name: Psych - stock bot that will give the approximate prices of stock in the future based off of news articles and world events)

Platform:

I really wanted to test my self and my ability to learn new processes and make an Android application as I have never even thought of trying to develop one before. I am hoping to gain a good understanding of UX and UI because most of my programing projects have focused on the back end of the software and not much of the front end. I will be using adobe Xd to help me decide on a clean and easy use interface.

Intended audience:

This app would be aimed to hit young adults that want to improve their programming as well and individuals trying to break into the industry to help them gauge if they are considered novice, intermediate, or a savant and point them to resources to improve.

The inspiration for my idea comes from my self and my friends that I help to program (Aden and Alicia). Aden is a brilliant individual and I consider a savant, but he is always downplaying what he can do when it comes to programming and Alicia is always saying that I am 1000x better and programming logic even though she solves problems I didn’t even realize existed.

Limitations:

Sadly, this is going to be a very time-consuming project because I have never developed an app and have next to no experience with front end design. On top of that I will be needing to reach out to developers at my work and in linked in to get a gauge on what working professionals consider to be novice, intermediate, or a savant and why then I will need to translate that into the test somehow.

Schedule:

I am aiming to have 2 week sprints where I gauge where I am at and what I need to complete the next step of the program.

* By Sep 19 – Have a prototype with a few of the language tests flushed out with a statistical break down.
* By Sep 26 – Have at least one test completely flushed out with links pointing to resources.
* By Oct 1 – have a Case study defined completely with feedback from professional programmers.
* By Oct 31 – Fully document my program and hopefully be done with all of it and begin bug testing
* By Nov 15 – Have a fully fleshed out App that would be ready to launch (this should give be about 10 days to add min more features and implement some nice-to-have ideas that pop in my head)

User training and/or user documentation:

Please forgive my ignorance but I believe this is asking for what the user manual would be right? If that is the case, then I when you first download the app or launch it for the first time I am thinking of making a quick 10 second video explain what the app is and what it does. After that it should be pretty straight forward for the user to begin using the tests.

A delivery/installation plan if:

This one will be the tricky issue because I am not sure what the requirements are to publish an app on the Play store but if I cannot get my app up in time then I will have a packet that can be installed onto your phone directly that will let you launch

Detail design:

1. Overview:

This section of the document will describe the main file design of the application. It is aimed to make it as streamlined and user friendly as possible.

1.1 Basic structure:

Diagram

Description automatically generated

1.2 Structure Overview

1.2.1: Main.py

Will be the main file to start the application

1.2.2: Index.py

This file will contain the data that will be displayed in the Display.py and be the directory you use to choose which test you would like to run.

1.2.3: C#.py

Contains all the testing material for C#

1.2.4: java.py

Contains all the testing material for javascript

1.2.5: python.py

Contains all the testing material for python

1.2.6: C.py

Contains all the testing material for C

1.2.7: C++.py

Contains all the testing material for C++

1.2.8: results.py

This is where all the results will be stored and calculated. It will also contain the directory that houses all of the ‘useful links’.

1.2.9: readme.txt

Contains basic usage instructions

1.2.10: GUI.py

This file houses all of the basic layouts that will be used across the document

1.3 pseudocode

* Starting the test
  + SET (classes) to a new class that has the visible
  + CALL showQuestions
    - Query from questions.js
      * LET class (option) be filled in by questions.js
    - OnClick attribute
* Selecting options
  + SET timer (0)
    - IF userAns == correctAns
      * SET class correct
    - ELSE SET class incorrect
* Score
  + IF userAns == correctAns
    - userScore ++
* showing results
  + SET classes for all of the test to their hidden class
    - IF userScore > 3
      * LET scoreTag = GOOD STUFF
    - IF userScore > 1
      * LET scoreTag = nice
    - ELSE
      * Let scoreTag = Oh no!

2. Detailed User Interface:

Graphical user interface, application

Description automatically generated