Capital Quiz Proposal

**Overview**

Create a system with capital quizzes from each continent/region. A user will be able to pick which continent they would like a quiz from. The quizzes will be 10 questions long which will include the 10 most populous countries from the continent/region. After the quiz, the user will receive their result and given the option to return back to menu to select another quiz if they’d like. Once completed, create a GUI for the quizzes making them multiple choice. The following list is the included regions and countries.

| Africa | Asia | Europe | North and Central America | Oceania | South America |
| --- | --- | --- | --- | --- | --- |
| Nigeria  Abuja | China  Beijing | Russia  Moscow | United States  Washington D.C | Australia  Sydney | Brazil  Brasilia |
| Ethiopia  Addis Ababa | India  New Delhi | Turkey  Ankara | Mexico  Mexico City | Papua New Guinea  Port Moresby | Colombia  Bogota |
| Egypt  Cairo | Indonesia  Jakarta | Germany  Berlin | Canada  Ottawa | New Zealand  Wellington | Argentina  Buenos Aires |
| DR Congo  Kinshasa | Pakistan  Islamabad | France  Paris | Guatemala  Guatemala City | Fiji  Suva | Peru  Lima |
| South Africa  Cape Town  (Legislative) | Bangladesh  Dhaka | United Kingdom  London | Cuba  Havana | Solomon Islands  Honiara | Venezuela  Caracas |
| Tanzania  Dodoma | Russia  Moscow | Italy  Rome | Haiti  Port-au-Prince | Micronesia  Palikir | Ecuador  Quito |
| Kenya  Nairobi | Japan  Tokyo | Spain  Madrid | Domincan Republic  Santa Domingo | Vanuatu  Port Vila | Chile  Santiago |
| Uganda  Kampala | Philippines  Manila | Ukraine  Kyiv | Honduras  Tegucigalpa | Samoa  Apia | Bolivia  Sucre |
| Algeria  Algiers | Vietnam  Hanoi | Poland  Warsaw | Nicaragua  Managua | Kiribati  South Tarawa | Paraguay  Asuncion |
| Sudan  Khartoum | Iran  Tehran | Romania  Bucharest | El Salvador  San Salvador | Tonga  Nuku’alofa | Uruguay  Montevideo |

**Technical Frame Work**

This will start as a command line but then be turned into a web application. The backend model will be a normalized SQLite database consisting of a table for the regions, countries, capitals and answers. Through the database a user would be able to perform CRUD operations as well as generating a quiz. The front end will be a series of web pages presented through the bottle app. The sites will be in HTML and CSS with bottle enhancements. The control layer will be a bottle app and a number of bottle view templates writing in python served by the bottle framework. The templates would include the main page with the option of selecting a quiz, a template for each quiz, and a template for the result of each quiz. The users will access the application through a web browser. The user will have limited access to the system.

**Date Design**

Majority of this application will be databases written in SQLite. The main goal in this project is to be able generate quizzes by region. In order to do this a database including the region, country, capital and a few other cities (options) would be appropriate for the large amount of data being used (60 entried in total). Another option would have been to make two separate tables one with region and country and another with capital and cities linked by the IDs. This way one could easily add more options to the questions. However, I decided to go with the first method. The database would include the following table

*Quiz*

* ID
* Region
* Country
* Capital
* Answer 1
* Answer 2
* Answer 3

The table will have an autonumbered primary key ID for every entry. By sorting, the countries by region and using a for loop. A quiz can be generated through the command line quite easily with this database. Similarly, the quiz will be generated on a GUI. With each quiz having their own page.

**User Flow and Visual Layout of GUI**

A user will begin on a home page with the following available flows. 

The main page will display the 6 regions as hyperlinks and when clicked will direct the user to the appropriate quiz. The main page layout will have a background image of the world and the hyperlinks will be listed and centered. Each quiz page will have their respective generated quiz with shuffled answer choices and a submit button. Unlike the command link, on the GUI the options will formatted with radio buttons. These radio buttons will be able to take the information of the page and be able to post it on the quiz result pages. Similarly to the main page layout, the quizzes will be centered and in an ordered list. They will contain a transparent background with a circular radiant background. On the quiz result pages, using the request bottle import the information from the radio buttons will be used to generate how many correct answers a user received on the quiz. On all the quiz result pages there will be a “back to menu” button which will redirect someone back to the main page.

**Technical Plan**

Data Creation

The first step in this project would be to create the database. It will be written through python and its sqlite3 module. It will have normal CRUD features along with the feature to generate a quiz.

Quiz Table

Since this is our only table it will not depend on any other table. It will be filled with 6 regions, with the 10 most populated countries, their capital, and 3 other cities in that country. One would be able to create, record, update, delete, search by country or region, and generate a quiz. Through different functions built using the database. For the GUI, it will be used only for users due to limited time so the function mainly needed is the generate quiz function to fill out the quiz pages in the interface.

GUI

After building all of the functions for the database the creation of the GUI through the bottle app will begin. Using HTML and CSS the page will be formatted as stated earlier. There will be forms used to record the answer a user submits and they will be requested on the result page to display the correct answers of a user.

Admin GUI (If possible)

Creation of GUI for admin purposes contianing a password and having the ability to display and use CRUDS

**Algorithm**

Import random module and sqlite3 module

Create a table with the following columns (ID, Region, Country, Capital, Ans1, Ans2, Ans3)

Fill in table with desired Regions,Country, Capital, and Cities

Create a function called CreateRecord

Ask the user Region and store in variable Region

Repeat for rest of the columns (not including ID)

Execute the input to the Quiz table

Create a function called ReportAll

Create a variable called result assigned to the quiz table

Create a loop that iterates through every record and prints it

Create a function called updateRecord

Create a variable called rn assigned to getRecordID function

If rn is equal to 0, output the message “Not a legal ID”

Else create a variable called result assigned to the quiz table

Then begin with sentry row that passes through the quiz table at the given input for rn

Ask the user for Region and store it in NewRegion

Whatever was in Region is converted to what is stored in New Region

Ask the user for Country and store it in NewCountry

Whatever was in Country is converted to what is stored in NewCountry

(Repeat for the rest of the columns)

Execute new inputs to Quiz table

Create a function called deleteRecord

Create a variable called rn assigned to getRecordID function

If rn is equal to 0, output the message “Not a legal ID”

Else create a variable called result assigned to the quiz table

Then begin with sentry row that passes through the quiz table at the given input for rn

Output the columns with their values of the given rn

Ask the user if they are sure if they want to delete record (Y/N) and store the result in confirmation variable

Capitalize confirmation variable

If confirmation variable starts with Y, delete record

Output message saying Record Deleted

Create a function called searchCountry

Ask the user what country to search for and store the result in searchCountry

Create a variable called result assigned to the quiz table where the country is column is equal to the searchCountry variable

Then begin with sentry row that passes through the quiz table at the result variable

Create a variable called row as a list of all the columns

Output the Region, Capital, Country with their values

Create a function called searchRegion

Ask the user what country to search for and store the result in searchRegion

Create a variable called result assigned to the quiz table where the country is column is equal to the searchRegion variable

Then begin with sentry row that passes through the quiz table at the result variable

Create a variable called row as a list of all the columns

Output the Region, Capital, Country with their values

Create a function called genQuiz

Ask the user what region and store in searchRegion variable

Capitalize input

Create a variable called result assigned to the quiz table where the country is column is equal to the searchRegion variable

Create a variable called score with value 0

Begin with sentry row that passes through the quiz table at the result variable

Create a variable called row as a list of all the columns

Create a list called answers with the indexes for Capital, Ans1, Ans2, Ans3 from row

Shuffle answers list

Ask the user a question What is the capital of a country with the capital and answers shuffled and store in variable called UserAnswer

Capitalize user answer

If useranswer is equal to capital add 1 to score variable and output correct

Else out incorrect

Output the final score

Create function called getRecordID

Create a variable called Result assigned to the Quiz table

Create a list of empty IDs called legalIDS

Begin with sentry row that passes through the quiz table at the result variable

Create a variable row assigned to the id and region columns

Output those columns with their values

Add current IDs to list of IDs

Ask user which id number and store in returnVal

If returnVal is not in the current IDs or a digit return 0

Else return the ID number