**PharmGenius Documentation #2**

**Setting Up the Testing Environment: (on Windows)**

(!!! Run your terminal (cmd, powershell, etc.) as Administrator for this setup process !!!)

1. Google Cloud SDK
   1. Go to <https://cloud.google.com/sdk/docs/quickstart-windows>
   2. Click on “Google Cloud SDK Installer”
   3. Run the installer
   4. When selecting components, make sure “Bundled Python” is selected
   5. On the last page:
      1. Make sure to “Run ‘gcloud init’ to configure the Cloud SDK”
      2. Make sure to “Start Cloud SDK Shell”
   6. gcloud init
      1. Log in to gmail.
      2. DO NOT configure Google Compute Engine
      3. Select a default project (pharm-quiz)[should happen automatically]
2. Python Environment Variable
   1. Press the windows key
   2. Search: “environment”
   3. Select: “Edit the system environment variables”
   4. Click the “Environment Variables” button
   5. Under “System Variables”, scroll until you find Path → double click
   6. Click “New” and add the path of the bundledpython folder
   7. If you kept the default: C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\platform\bundledpython\
3. Google App Engine
   1. $ gcloud components install app-engine-python

**Running the App on Localhost:**

1. cd [application directory]
2. $ python [Full path to dev\_appserver.py] . ← (there is a space-period there)
   1. Default path: “C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin\dev\_appserver.py”
   2. This runs the app on localhost:8080
3. If that doesn’t work (it didn’t for me), you can try a different port (which did work)
   1. $ python [Full path to dev\_appserver.py] --port [portNum] . I used 8084

**Deploying the App**

1. RUN cmd/PowerShell AS ADMINISTRATOR
2. Make sure your configuration only allows one file upload process at once
   1. $ gcloud config set app/num\_file\_upload\_processes 1
   2. This should only need to be done once
3. From the application folder:
   1. $ gcloud config set project aecs1980qg
   2. $ gcloud app deploy
4. To open the site:
   1. $ gcloud app browse

**Languages / Frameworks**

* Backend: Python 2.7, Django, Google Cloud API
  + <https://cloud.google.com/python/>
* Frontend: Javascript with some jQuery
  + <https://jquery.com/>
  + <https://datatables.net/>
* Interface: HTML, CSS with Bootstrap
  + <https://getboostrap.com/>
  + <https://getbootstrap.com/examples/grid/>

**TODO**

**Bug Fixes**

|  |
| --- |
| **Description** |
| Align the NavBar drop-downs |
| Logging out of PharmGenius logs you out of Google entirely |
| Sometimes the broken image icon appears instead of a blank string when the user doesn’t have a profile picture. This only happened when we were testing it with Ravi; I couldn’t recreate it. |
| Add-A-Question: Some categories are added twice (implement ignoreCase and don’t consider whitespace) |
| Leaderboard: Selecting a specific category or subcategory doesn’t alter the table. (only on the live version; it works on my local environment) |
| Quiz: Percentages of who picked what answer aren’t calculated |
| Quiz: Can’t un-upvote or un-downvote - you can switch sides, but there is no going back to neutral. |
| After an admin accepts a question, it isn't removed from the table immediately when you return to the Questions Under Review page, but only after you refresh the page. This isn’t currently a problem because users can automatically add questions. |

**Features**

|  |
| --- |
| **Description** |
| Let users decide how many people to view on the leaderboard  And have it cut off at top 10 by default |
| Make it so that when the user clicks on a leaderboard entry, they are redirected to that user’s profile page |
| Allow each question to be tagged with multiple of each category |
| Allow users to take quizzes with multiple categories |
| Add a Badge / Achievement system |
| Allow users to “add” their friends, and filter the leaderboard by friends |
| Allow users to suggest an explanation to questions without one  (this should have upvote downvote capabilities separate from question upvote downvote) |
| Permission Levels: Admin, Professor, Student |
| Accept non-gmail email (with confirmation emails) |
| Accept Pitt Passport login |

**Notes:**

* General
  + All CSS and JavaScript should be [minified](http://www.minifier.org/) for release
  + Bootstrap is built upon mobile-first design; you should do the same
  + A lot of the Django to separate admin and user privileges has been commented out to allow beta testers to have more control.
    - For example, they can add a question automatically (without other users having to vote for them)
    - The bracket-notation was removed so that it could be commented out with HTML comments. So, basically, <!-- {% if admin %} --> doesn’t actually comment out the Django.
  + Currently, subcategories are implemented by creating separate categories that have the format: category:subcategory, and they are split before being used. It would be better to add the subcategories to the models; what we did was to create a minimum viable product for user testing.
* NavBar
  + \_base.html
  + The “Profile” drop-down is the only one that is aligned with the navbar - Don’t know why
  + There were some bugs that only appeared when Ravi was testing. One example was that the profile image (in the navbar) used the “broken image” image when the user didn’t have a profile picture instead of no image, as I have specified by alt=””. I haven’t been able to recreate that.
  + Ravi discussed considering moving “hamburger menu” (see mobile version) to a different corner of the screen. It would be more accessible, for example, on the bottom-right corner, where your thumb naturally rests.
  + IMPORTANT: We do not have permission to use this image yet, but this is a Pitt Pharmacy app, and this is the Pitt Pharmacy logo. Permission should be obtained to avoid legal problems. I have included the Adobe (CS6) Illustrator and Photoshop files I used to create it in /Documentation/Team 2/Media.
* Home Page
  + index.html
  + This page used to inherit from \_baseHome.html. We switched it to \_base.html so that it looked more consistent with the rest of the app.
  + It can be accessed from “/” or from “/play”
    - “/play” is where the “Quiz Me!” NavBar button links to
    - It will begin on the take a quiz form if accessed from “/play”
* Leaderboard
  + leaderboard.html
  + The leaderboard table isn’t centered in a sm viewport (see Bootstrap)
  + The drop-downs aren’t doing anything on the live version of the site, but they work on my local version.
  + Some categories are added twice - This, I think, has to do with how categories are added. For instance, BioChemistry and Biochemistry are two different categories - so when checking to see if a category already exists, it should be done by ignoring case (this is not implemented). Some have the same capitalization, but different whitespace. This also needs to be fixed. (on the )
* Questions
  + viewDatabase.html, reviewNewQuestions.html
  + Under the NavBar buttons, “Your Questions”, “View Question Database”, and “Review New Questions”
  + These use DataTables, a plugin for jQuery <https://datatables.net/>
  + The “Your Questions” and “View Question Database” both use viewDatabase.html
* Quiz
  + quiz.html submitQ.js
  + For the Single-Question Results, the percent of people who answered each answer isn’t being calculated.
  + Answers are randomly placed and colored, so that users don’t just memorize the location/color of the correct answer
  + I kept the previous code for calculating score - we changed it to a flat +100 per correct answer

I wish you all luck in continuing to work on this project. Our team wasn’t especially familiar with the frameworks/codebase, and that made this project a little difficult. I highly suggest you review the code and the documentation available at the links provided above before you jump into making changes. Otherwise, you may find yourself overwhelmed by the legacy code later in the semester - we sure did.

One thing you may want to consider is to migrate the code from Google Cloud API to something that could work on Pitt’s Unix servers - It may be in your customer’s best interest to not be stuck using GC API, which forces the application to be hosted on Google’s App Engine.