Documentation of Assignment 1

Formula1 Statistics

Author: Ali Ramezanian Nik

main.py:

- **Libraries**
  - google.oauth2.id_token
    - Parse and verify an ID Token issued by Google OAuth 2.0 authorization server
  - google.cloud
    - Access datastore module from App engine
  - flask
    - Access Flask Framework which is a simple handler of HTTP methods
  - google.auth.transport
    - This library simplifies using Google various server-to-server authentication mechanisms to access Google APIs
- **Variables and data structures**
  - att_list
    - type: List[Dic]
    - description: A dictionary that is useful to access kinds by keys and their properties are stored as a list and it is used to easily iterate on all properties by the key.
- **Functions**
  - flash_redirect(message, path)
    - type: String, String → Redirect
    - description: This function not just simply forward to another page and instead it also carry a message to page redirected.
  - get_session_info()
    - type: None → Option(Redirect, Dic, None)

    - description: Based on the token retrieved from cookies it either retrieve data of that token from firestore, fail to get the token an return silently(None) or fail to communicate with firestore to fetch the information of the token and redirect to error page (503).
- retrieve_row(kind, name)
    - type: String, String → Option(None, google.cloud.datastore.entity.Entity)
    - description: fetch the information of the specified row and return a copy of that data. As the fetch information has type of google.cloud.datastore.entity.Entity but we prefer to return Dic.
- update_row(kind, name)
    - type: String, String → None
    - description: update the row specified by (kind, name) based on the information retrieved from the form that sent to server via the POST method.
- create_row
    - type: String, String → None
    - description: create a row specified by (kind, name) based on the information retrieved from the form that sent to server via the POST method.
- delete_row
    - type: String, String → None
    - description: delete a row specified by (kind, name)
- query_result(key, comp, val, kind)
    - type: String, String, String, String → List(Option(None, google.cloud.datastore.entity.Entity)
    - description: query over the “kind” database with the filter provided: “key, comp, val” . If filters or kind does not provided or if there is no row with those filters fetched, then List(None) will be returned. Otherwise it return the result as List of google.cloud.datastore.entity.Entity
- priority_taging(retreived, kind):
    - type: List(google.cloud.datastore.entity.Entity), String → List(google.cloud.datastore.entity.Entity)
    - description: the “retrieved” parameter is a List with only two elements in it (but it can be expandable to more elements, though the implementation of this function should change correspondingly.) and each element is a map of properties with their values. The function compare element1 and element2 values for every property and replace the

value of each property with new value with following format: (value, comparison_tag). Good to note that the comparison_tag is only a value that indicating the result of comparison (>, < or =)

- tag_returner(a, b, col)
  - type: String, String, String → String, String
  - description: this function return the result of comparison of two index of database in the shape of two Strings . The comparison policies is based on the assignment rules. Also it taken care of non comparable values and change them to comparable values
- projection_on(kind, prop):
  - type: String, String → List(String)
  - description: query on the specified table(kind) but limiting the returned properties and return only one of the column values. The returned object is a list of values recorded for specified property.
- add(kind):
  - type: String → Option(rendered_template, redirect)
  - description: This function provide an html page with a form for every GET uest that invokes it. If a POST request provide the values of form field for this function, first the function verify that the key value provided by form is not recorded in table. If its not been added already, it create a new row in database. If the function can not verify the user authentication it won't server the request and instead redirect user to homepage with helpful flash message.
- query()   :
  - type: None → Option(rendered_template, redirect)
  - description: First provide a html page that has a form to query on all fields of tables and the provided page has a button that fire a POST request and sends the form data to this function (query) and then fetches the information from the datastore and finally return another html page that shows those information
- compare()
  - type: None → Option(rendered_template, redirect)
  - description: The first aim of this function is to response a html page to make it possible to input the compare data first, and second is to provide a html page to see the result of comparison.
- update(kind, name):

        - type: String, String → Option(rendered_template, redirect)
        - description: update function also has two duty: first to provide a html page that show a form that has been filled with fetched data for (kind, name) and these (kind, name) is provided by Flask dynamic paths. The second duty is to update the (kind, name) with the new data written in form and posted to this function. If the function can not verify the user authentication it won’t server the request and instead redirect user to /query with helpful flash message.
    - delete(kind, name)
        - type: String, String → redirect
        - description: This function is almost pass the variables (kind and name) that provided by Flask to delete_row function and redirect to appropriate link. If the function can not verify the user authentication it won’t server the request and instead redirect user to /query with helpful flash message.
    - root()
        - type: None → rendered_template
        - description: This function just show the homepage with or without user fetched information provided by get_session_info() .