

Coverage for **server.py**: 100%

95 statements 95 run 0 missing 0 excluded

« prev ^ index » next coverage.py v6.5.0, created at 2022-11-08 12:19 +0100

```
1 import json
2 import datetime
3 from flask import Flask,render_template,request,redirect,flash,url_for, abort
4
5
6 # ===== FLASK APP INITIALIZATION
7
8 app = Flask(__name__)
9 app.secret_key = 'something_special'
10
11 def page_not_found(e):
12     return render_template('404.html'), 404
13
14 app.register_error_handler(404, page_not_found)
15
16
17 # ===== CONSTANTS
18
19 MAX_PLACES = 12
20 PATH_JSON_CLUBS = 'clubs.json'
21 PATH_JSON_COMPETITIONS = 'competitions.json'
22
23 # ===== LOAD JSON DATA
24
25
26 def loadClubs():
27     try:
28         if PATH_JSON_COMPETITIONS:
29             with open(PATH_JSON_CLUBS) as c:
30                 listOfClubs = json.load(c)['clubs']
31                 if listOfClubs and len(listOfClubs) > 0:
32                     return listOfClubs
33             else :
34                 raise Exception("This file is empty, or key not found")
35
36     except Exception as error:
37         return str(error)
38
39
40 def loadCompetitions():
41     try:
42         if PATH_JSON_COMPETITIONS:
43             with open(PATH_JSON_COMPETITIONS) as c:
44                 listOfCompetitions = json.load(c)['competitions']
45                 if listOfCompetitions and len(listOfCompetitions) > 0:
46                     return listOfCompetitions
47             else :
48                 raise Exception("This file is empty, or key not found")
49
50     except Exception as error:
51         return str(error)
52
53 competitions = loadCompetitions()
54 clubs = loadClubs()
55 now = datetime.datetime.now()
56 current_date = now.strftime("%Y-%m-%d, %H:%M:%S")
57
58
59 # ===== INDEX ROUTE
60
61
62 @app.route('/')
63 @app.errorhandler(500)
64 def index():
65     """
66     Displays home page of the app.
67     if no data is available then return :
68         - status code 500 and the appropriate template (500.html)
69     """
70
71     if isinstance(clubs, list) and isinstance(competitions, list):
72         return render_template('index.html')
73     else:
74         return render_template('500.html'), 500
75
76
77 # ===== SUMMARY PAGE
78
79
80 @app.route('/showSummary', methods=['POST'])
81 def show_summary():
82     """
83     Displays the summary of all competitions
84     and points available of the connected club
85     if email is not found return error message
86     """
87
88     try:
89         club = [club for club in clubs if club['email'] == request.form['email']][0]
90
91         if club:
92             return render_template('welcome.html', club=club, competitions=competitions, current_date=current_date)
93
94     except IndexError:
95         flash("Sorry, this email wasn't found. Please try again with a correct email !!")
96         return redirect(url_for('index'))
97
98 # ===== ROUTE FOR BOOK PLACES
99
100
101 @app.route('/book<competition><club>')
102 def book(competition,club):
103     """
104     Displays :
105     Competition's name and places available
106     & the booking form or displays an error message.
107     """
108
109     try:
110         foundClub = [c for c in clubs if c['name'] == club][0]
111         foundCompetition = [c for c in competitions if c['name'] == competition][0]
112         return render_template('booking.html',club=foundClub,competition=foundCompetition)
113
114     except IndexError:
115         flash("Sorry, this club or competition wasn't found !!")
116         return render_template('welcome.html', club=club, competitions=competitions, current_date=current_date)
117
118 # ===== ROUTE FOR PURCHASE PLACES
119
120
121 cart = {
122     competition["name"]: {club["name"]: 0 for club in clubs}
123     for competition in competitions
124 }
125
126
127 @app.route('/purchasePlaces',methods=['POST'])
128 @app.errorhandler(400)
129 def purchase_places():
130     """
131     Act like a Controller of the booking form:
132     - By validating it,
133     - Or handling errors
134     """
135
136     competition = [c for c in competitions if c['name'] == request.form['competition']][0]
137     club = [c for c in clubs if c['name'] == request.form['club']][0]
138     club_points = int(club['points'])
139     placesRequired = int(request.form['places'])
140     current_cart = cart[competition["name"]][club["name"]]
141     current_places_available = int(competition['numberOfPlaces'])
142
143     if competition['date'] < current_date:
144         flash("Sorry, this competition is over !")
145         return render_template('welcome.html', club=club, competitions=competitions, current_date=current_date), 400
146
147     elif placesRequired > club_points:
148         flash("Sorry, your club doesn't have enough points !")
149         return render_template('booking.html', club=club, competition=competition, current_date=current_date), 400
150
151     elif placesRequired > MAX_PLACES :
152         flash(f"Sorry, you can't book more than {MAX_PLACES} places !")
153         return render_template('booking.html', club=club, competition=competition, current_date=current_date), 400
154
155     elif placesRequired + current_cart > MAX_PLACES :
156         flash(f""""Sorry, you have already booked places, for this competition
157         and now you have exceeded the limit of {MAX_PLACES} places !""")
158         return render_template('booking.html', club=club, competition=competition, current_date=current_date), 400
159
160     elif placesRequired > current_places_available:
161         flash("Sorry, you book more places than available !")
162         return render_template('booking.html', club=club, competition=competition, current_date=current_date), 400
163
164     else:
165         competition['numberOfPlaces'] = int(competition['numberOfPlaces'])-placesRequired
166
167         club['points'] = int(club['points'])- placesRequired
168
169         cart[competition["name"]][club["name"]] += placesRequired
170
171         flash(f"Great you have booked {placesRequired} places! for {competition['name']}")
172
173         return render_template('welcome.html', club=club, competitions=competitions, current_date=current_date, add_to_cart=cart[competition["name"]][club["name"]])
174 # ===== ROUTE FOR BOARD DISPLAY
175
176
177 @app.route('/board', methods=['GET'])
178 def show_board():
179     """
180     Displays the Public Board of all clubs with their points.
181     """
182     return render_template('board.html', all_clubs=clubs)
183
184
185 # ===== ROUTE FOR LOGOUT
186
187
188 @app.route('/logout')
189 def logout():
190     return redirect(url_for('index'))
```