TOURISM

We want to gently show the list of places that are worth to see in Turkey for tourists. Here, we need your help to develop a mobile app with Flutter.

The app will interact with the pre-developed small-sized (76 lines) backend server by REST API --the source code is available in hello.py and a reference client code is available in client.py

Great Expectations

The app should have 4 distinct pages: **'login-page'**, **'cities-list-page'**, **'city-places-list-page'** and **'place-details-page'** which we describe below.

1) login-page: User inputs (username and password) should be passed to backend by calling "http://URL/rest/login"". Upon unsuccessful login response, user should be notified. Upon successful response, user should be routed to 'cities-list-page'

2) cities-list-page: App should fetch the list of cities that have places worth to see by calling "http://URL/rest/cities". Upon getting response, user should see the list of cities, and would be able to tap one of them to route 'city-places-list-page'

3) city-places-list-page: App should fetch list of places by calling "http://URL/rest/places" with city code as data. Upon getting response, user should see the list of places (only 'landing' image and 'intro' text). User should be able to tap one of them to route 'place-details-page'. User should also have the ability to return through 'cities-list-page'

4) place-details-page: App should show 'landing' image and 'alternative' images if any (preferably as sliding images, as in Instagram) with 'full' text. User should have the ability to route back to 'city-places-list' page.

Remarks

- Successful login response will contain json-web-token. This should be used for all further API calls for authentication.
- I don't recommend, but you're free to modify pre-developed backend code. Our single restriction is, all API calls (except /rest/login) have to use json-web-tokens.

Bonus Points

1) Handle disconnection: If the mobile device does not have an internet connection at login, or lost it after the login phase, just notify her as in today's modern mobile applications.

2) Develop auto-login and log-out: Once user enters the correct credentials, just auto-login her on her second application launching. Besides, develop log-out option, in which if user logs-out, route her to the login page on her second application launching --as in most modern applications that we've been using such as Instagram, Twitter, Gmail etc.

hint: https://pub.dev/packages/shared_preferences

3) Show distance: Get user's longitude and latitude information, calculate her distance to show how far she is to the corresponding place in 'place-details-page'.

hint: https://pub.dev/packages/geolocator

4) Add search filter: Let user to search cities and places in city by implementing a search-bar.

hint: https://pub.dev/packages/search_page

Installation

Prerequisites

- install python
- install virtualenv

Launch server

```
cd tourism

. bin/activate

export FLASK_APP=hello.py

flask run --host=YOUR_LAN_IP_ADDRESS --port=5000
```

Test server

```
cd tourism

. bin/activate

python3 client.py
```