Busara

Fullstack Dev Task(Frontend)

Overview

The goal of this exercise is to quickly test your development aptitude. You will be expected to develop a web app based on provided APIs and at least with the items provided in the scope. You are free to determine any tech stack of your choice. At Busara we're looking for the best engineering talent so this test project is a way for you to prove that you're in the top 1% of developers in Kenya.

The expectation is that you can build this app in 7 days, if you finish before then even better. Feel free to ping us in advance, if you need more time to deliver this challenge.

If you have any questions along the way, either technical or otherwise, don't hesitate to get in touch with our hiring manager through the email chain.

Product Overview

Scope

Busara is implementing a data collection platform, which allows respondents to fill in surveys through a web/browser interface.
The purpose of this task is to create an application that will

1. Allow a user to login.
2. Allow data collection through a web/browser interface. We are looking for implementation of open ended (HTML text) and select/multiselect question types.
3. Submit data to a backend service.
4. Feel free to add your creativity into the data collection/handling or any other bit of the application

You will be provided with 4 APIs

1. Registration API
2. Login API
3. Form Data/Survey API

4. Data submission API

Additionally you will be required to:

- Host the code on github or other versioning system and optionally on publicly accessible servers (you can use https://www.heroku.com)
- Feel free to share any docs in advance for review.

What we are looking for

In this challenge we are looking for the structure and quality of code, your thought process in product development, user experience and proficiency of skills on the frontend stack. The backend service **is provided as indicated above**, so you don't have to code a custom backend!

Setup and technical requirements

The backend of the app is managed by a django backend using RESTful APIs and 0auth. All requests are signed with a token except for some publicly available endpoints.

1. **You will need to register yourself into the application.** Follow these steps below
    a. Access the URL <base url>/api/v1/users/registration/ and register yourself with the email in your CV.
    Note
        - The username is your email address
        - Enter a valid phone number ie +254721 000 000
        - Referral Code is not required.

    Sample Registration JSON

    ```
    {
            "username": "<Your Email>",
            "email": "<Your Email>",
            "password1": "<Your password>",
            "password2": "<Your password>",
            "referral_code": "",
            "phone_number": "<Your Phone Number>",
            "full_name": "<Yout Full Name >",
            "device_details": {"device": "Dummy"},
            "location": “Dummy”
    }
    ```

    b. Send us an email with your username, and will complete the signup process for you. A **client ID** and **secret** will be sent to you, which you'll use for other requests

2. Login to application via login endpoint on the application server (retrieve and save the session access token for future use i.e if you have a valid session token already there is no need to login again).
3. Each request to the backend will require you to pass a session token except for the API endpoints: login[POST] and user registration[POST].
4. Kindly see below for documentations on access and implementation
    a. https://www.djangoproject.com/ - the django project
    b. http://www.django-rest-framework.org/ - django based RESTful API implementation
    c. https://django-oauth-toolkit.readthedocs.io/en/latest/ - django based 0auth implementation

<u>Product Scope and Flow</u>

1. Login into the app
    a. This will provide a token that will be used for subsequent calls to the backend. This is provided by the API endpoint
    b. Sample request:
    ```
    curl -X POST -d
    "grant_type=password&username=<username>&password=<password>"
    -u"<client ID>:<client secret>" <base url>/api/v1/oauth/token/
    ```
    c. Sample response:
    ```
    {  "token_type": "Bearer",
        "scope": "groups read write",
         "expires_in": 36000,
        "refresh_token": "<your_refresh_token>",
        "permissions": [<A list of user  permissions>],
        "access_token": "<your_access_token>"
    }
    ```

2. Get details of currently logged in user. You can display logged in user details here e.g. first/last name, location, etc (*your creativity*)
    a. <base url>/api/v1/users/current-user
    Sample Response
    ```
    {
      "id": 1,
      "email": "<Your Email>",
      "first_name": "<First Name>",
      "last_name": "<Last Name>",
      "phone_number": "<Phone Number>",
      "groups": [1],
    ```

```
    "device_details": {},
    "is_active": true,
    "language": "English",
    "user_timezone": "Africa/Nairobi",
    "universe": 1,
    "access_token": "<Your Access Token>",
    "is_subject": true,
    "permissions": [],
    "referral_code": "<Your referral code>",
    "name": "<Your Full Name>",
    "project": null,
    "approver_level": "Initiator",
    "universe_name": "<Your Universe>",
    "subject": "<A subject ID>"
   }
```

3. Call the Survey API to get Survey/Form Data JSON **.** This is after a successful login
    a. <base url>/api/v1/recruitment/forms/?node_type=Both

    NB. Remember to attach to the headers the auth token you got at login to access this API

    This will be used to create the survey data collection view. It will have a tree with forms, pages and sections and questions .

    The form will contain the pages, pages contain sections(we can have more than one section per page) and sections contain one or more questions.

    Sample JSON Structure(Without content)

```
{
 "forms": [
        {
        "pages": [
                {
        "sections": [
                {
                "questions": [
                        {
                                }
                        ]
                }]
               }]
              }
         ]
}
```

4. Submit the answers to the backend
   <base url>/api/v1/recruitment/answers/submit/
   Sample JSON

```
[ {"ans": [ {
          "column_match": "first_name",
          "q_ans": "Respondents First Name",
          "q_id": "8290"
          },{
          "column_match": "gender",
          "q_ans": 25871,
          "q_id": "8286"
          },{
          "column_match": "date_of_birth",
          "q_ans": "1996-02-27",
          "q_id": "8287"
          }],
          "end_time": "2021-02-03 11:35:16.649 +0300",
          "local_id": 0,
          "location": {
          "accuracy": 0,
          "lat": 0,
          "lon": 0
          },
          "start_time": "2021-02-03 11:27:37.739 +0300",
          "survey_id": <Survey ID - Retrieve this from the FORM Data>
  }
]
```

NB

- Each question has a column_match, this is needed to submit answer
- Start time and end time are the times the participant started and completed the survey
- Local ID and Location Data can be sent as is shown on the sample json
- If question is select/multi-select type, send the option id as the answer

<u>API Access Details</u>

Base url: http://104.248.0.49/