React Task by ShoringCode / Eduwo

Scope

The purpose of this task is to evaluate your ReactJs skills, code quality, creativity and commitment.

The application should be done only by you, from scratch in ReactJS.

This task should be made between 8 to 16 hours.

Use your github account to commit and share the code.

Please do your best and good luck!

Objective

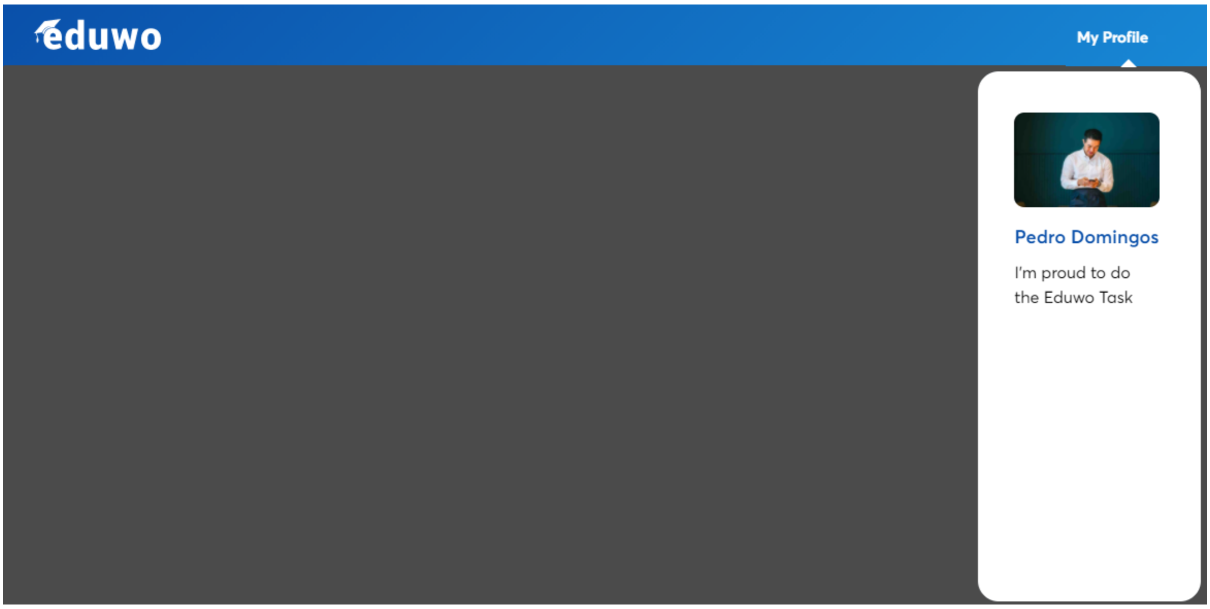
Create a ReactJs application to search cities around the world and show some details of the cities.

The React application must have a header, a footer and a body.

The header and footer design are described below

The search and the city details will be shown in the body. You are totally free to use your creativity.

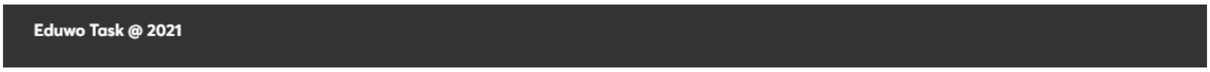
Header



- Background: Linear gradient from #0a50aa to #1888d5
- Height: 60px
- Width: 100%

- Left side: eduwo logo (the logo svg is at the end of this document)
- Right side: “My Profile” is a flyout. The flyout should open when the mouse hover.
  The box with a photo (can be yours or not), your name and a description.
  You are free to write whatever you want in the description.

Footer



Eduwo Task @ 2021

- Background: rgba(0,0,0,.8)
- Height: 60px
- Width: 100%
- Left side: White and bold text “Eduwo Tasks @ YYYY”, where YYYY is the current year.

Body

In the body is where you will implement the search screen and the city details. All the cities in the output must be clickable. By clicking on the city the user should be taken to the city details page.
To search the cities and get the city details we recommend you use the GeoCities API, but you can choose other APIs if they fit better to you. At the end of this document, you can find some useful links about GeoCities API.

For the search you must have by minimum:

- Filter by the name prefix
- Sort by name A-Z and Z-A
- Paginated results, show 10 per page.
- The cities on the output must be clickable. By clicking the user should be taken to the city details page.

NOTE: GeoCities API allows you to get the cities filtered, ordered and paginated. That’s the way that you should filter, sort and paginate the results. If you choose another API, this APIs should also allow that.

You are completely free to add other filters. The design and displayed fields are totally up to you.
When the user clicks on the city, to see the city details, you need to call the city details API and show the returned information from the API. The design and display fields are totally up to you.

Extra points

- Update the application url:
  - / -> Root page with the search
  - /{city id or name} -> City detail page
- Use redux
- Proper commit history
- In the city details show city images from some API.
- Mobile responsive
- Anything that will impress us (be creative)

Additional Information

GeoCities API urls:

Page: http://geodb-cities-api.wirefreethought.com
API documentation: http://geodb-cities-api.wirefreethought.com/docs/api
Find Cities Api: http://geodb-cities-api.wirefreethought.com/docs/api/find-cities
Get City Details API: http://geodb-cities-api.wirefreethought.com/docs/api/get-city-details

Logo svg code:

```
<svg width="123px" height="30px" viewBox="0 0 123 30" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <title>Logos/Eduwo-S-White</title>
  <g id="Logos/Eduwo-S-White" stroke="none" stroke-width="1" fill="none" fill-rule="evenodd">
    <path d="M110.136067,19.4228934 C110.136067,17.8076629 110.393225,16.5073346
110.907782,15.5209211 C111.421613,14.5347544 112.283405,14.0416711 113.49316,14.0416711
C114.714295,14.0416711 115.58529,14.5317929 116.105658,15.5117899 C116.625301,16.4917869
116.885364,17.7955703 116.885364,19.4228934 L116.885364,19.8108449 C116.885364,21.4751862
116.625301,22.7883476 116.105658,23.7498355 C115.58529,24.7115701 114.726403,25.1920672
113.528997,25.1920672 C112.295028,25.1920672 111.421613,24.7142848 110.907782,23.7589666
C110.393225,22.8036485 110.136067,21.4877725 110.136067,19.8108449 L110.136067,19.4228934 Z
M104.003512,19.8108449 C104.003512,22.8317824 104.838427,25.2846129 106.507772,27.1708171
C108.176875,29.0567745 110.517445,30 113.528997,30 C116.504713,30 118.827364,29.0567745
120.496467,27.1708171 C122.165812,25.2846129 123,22.8317824 123,19.8108449 L123,19.4228934
C123,16.4150357 122.165812,13.9649198 120.496467,12.0722991 C118.827364,10.180172
116.492363,9.23373829 113.49316,9.23373829 C110.505095,9.23373829 108.176875,10.180172
106.507772,12.0722991 C104.838427,13.9649198 104.003512,16.4150357 104.003512,19.4228934
L104.003512,19.8108449 Z M89.8741315,9.6034274 L86.0097454,9.6034274 L82.8163422,21.2900949
L82.7076192,21.2900949 L80.5481735,9.6034274 L74.8875587,9.6034274 L79.5139738,29.611555
L84.6849722,29.611555 L87.8967785,18.9971833 L88.0055014,18.9971833 L91.1986625,29.611555
L96.3878218,29.611555 L100.996076,9.6034274 L95.3354613,9.6034274 L93.2125794,21.1980428
L93.1038564,21.1980428 L89.8741315,9.6034274 Z M65.8928545,29.611555 L71.2454614,29.611555
L71.2454614,9.6034274 L65.1129067,9.6034274 L65.1129067,24.0829999 C64.7615549,24.4529358
64.3382861,24.7298324 63.842616,24.9149238 C63.3467039,25.0995216 62.7723368,25.1920672
62.1187885,25.1920672 C61.3448943,25.1920672 60.7489763,24.9149238 60.3320033,24.3603901
C59.914546,23.8053629 59.7060595,22.9670225 59.7060595,21.8448754 L59.7060595,9.6034274
L53.5914235,9.6034274 L53.5914235,21.8078571 C53.5914235,24.6071786 54.1689384,26.6713173
55.3242104,28.0032346 C56.4792402,29.3344115 58.0730361,30 60.1051138,30 C61.25409,30
62.2846575,29.7623427 63.1985113,29.2882621 C64.1116387,28.8139347 64.8826272,28.1323049
65.5114768,27.2448536 L65.8928545,29.611555 Z M36.5410521,19.6445095 C36.5410521,17.955736
36.7827124,16.5998802 37.2667595,15.5764485 C37.7503223,14.5535103 38.5365659,14.0416711
```

```
39.6254903,14.0416711 C40.2540977,14.0416711 40.8107882,14.1838212 41.2945931,14.4666408
C41.778156,14.7506943 42.1774524,15.1576485 42.4919983,15.6872565 L42.4919983,23.7680978
C42.1774524,24.2372427 41.7752502,24.5908905 41.2856338,24.8315093 C40.7957752,25.072128
40.2301254,25.1920672 39.5891686,25.1920672 C38.4883791,25.1920672 37.7047991,24.7389636
37.2393972,23.8327564 C36.7737531,22.9270428 36.5410521,21.6602776 36.5410521,20.0332013
L36.5410521,19.6445095 Z M48.6247951,0.764698721 L48.6247951,29.611555 L43.30851,29.611555
L42.8731339,27.152308 C42.2682567,28.0767775 41.5633737,28.7830861 40.7594535,29.2697529
C39.9545648,29.756173 39.0448275,30 38.0287886,30 C35.6092796,30 33.7343538,29.0876231
32.4042534,27.2633628 C31.1367992,25.525737 30.4731143,23.2569284 30.4127803,20.4571502
L30.4082552,20.0332013 L30.4082552,19.6445095 C30.4082552,16.5260905 31.0765744,14.0108225
32.4132127,12.0996927 C33.7493667,10.1890564 35.6337362,9.23373829 38.0651103,9.23373829
C38.9721841,9.23373829 39.7916015,9.44622315 40.5236047,9.87168645 C41.1738437,10.2498761
41.7645809,10.7717758 42.2956463,11.4370389 L42.4919983,11.6932321 L42.4919983,0.764698721
L48.6247951,0.764698721 Z M18.4741333,8.96797287 C21.1353027,8.96797287 23.2218629,9.78150995
24.7340561,11.4088331 C26.1770509,12.9619514 26.9316746,15.0434978 26.9972954,17.6526138
L27.0019826,18.0289335 L27.0019826,21.301842 L15.7889916,21.301842 L15.7526699,21.41265
C15.8856073,22.435835 16.297011,23.2776304 16.9866389,23.9368024 C17.6760247,24.5969615
18.5952057,24.9264241 19.744424,24.9264241 C20.9055075,24.9264241 21.83389,24.8338785
22.5293293,24.6487871 C23.1477129,24.4847002 23.9116941,24.212782 24.8214431,23.8342457
L25.1694321,23.6872993 L26.620847,27.4964792 C25.76196,28.1376356 24.6522112,28.6706986
23.2916006,29.0964087 C21.9307478,29.5213785 20.4701315,29.7341101 18.9097515,29.7341101
C16.0187869,29.7341101 13.7116329,28.8096405 11.9878054,26.959961 C10.3391584,25.1914106
9.47899483,22.994301 9.40731453,20.3684163 L9.40242723,20.0076833 L9.40242723,19.3048298
C9.40242723,16.2599538 10.1855229,13.7730665 11.7519565,11.8431807 C13.3181479,9.91428214
15.5591964,8.95550898 18.4741333,8.96797287 Z M26.1668743,0 L19.9309238,7.2279404
L20.4491133,8.45670014 C10.7991658,7.10972873 9.27438116,15.3914559 9.27438116,15.3914559
L9.27438116,15.3914559 L8.19247899,13.4591022 C11.0047466,7.3113549 16.8740901,6.97893085
16.8740901,6.97893085 C10.5565369,6.7296745 7.70334685,13.2098459 7.70334685,13.2098459
L7.70334685,13.2098459 L2.81250978,13.8329867 C2.81250978,13.8329867 2.73090704,14.912933
2.6899846,15.2038966 C2.64930431,15.4946134 3.17911674,15.9099584 3.66824888,16.8238161
C4.15738101,17.7379205 4.03509798,18.5688573 4.03509798,18.5688573 C5.50249438,22.7642611
2.6899846,23.4288624 2.6899846,23.4288624 C2.85319008,23.2210665 2.85319008,22.5564652
2.44566068,21.4348117 C2.03788914,20.3131581 2.93455067,18.6103177 3.30164192,18.069851
C3.66824888,17.5306182 2.97547311,16.7821088 2.3231355,15.8270375 C1.67104003,14.8714726
2.07881158,13.9576149 2.07881158,13.9576149 L2.07881158,13.9576149 L0,14.0824899
L10.0264824,2.24306031 L26.1668743,0 Z M18.4741333,13.7755344 C17.5910319,13.7755344
16.9290085,14.0963594 16.4875789,14.7375158 C16.0900743,15.3143345 15.8324154,16.0656647
15.7144259,16.9924059 L15.6802687,17.3078176 L15.7345091,17.4003633 L21.0144726,17.4003633
L21.0144726,16.9193726 C21.0144726,15.908527 20.8086496,15.1321305 20.3974881,14.5891959
C19.9863265,14.0472485 19.3451275,13.7755344 18.4741333,13.7755344 Z" id="Fill-1" fill="#FFFFFF"/>
  </g>
</svg>
```

Good Luck!