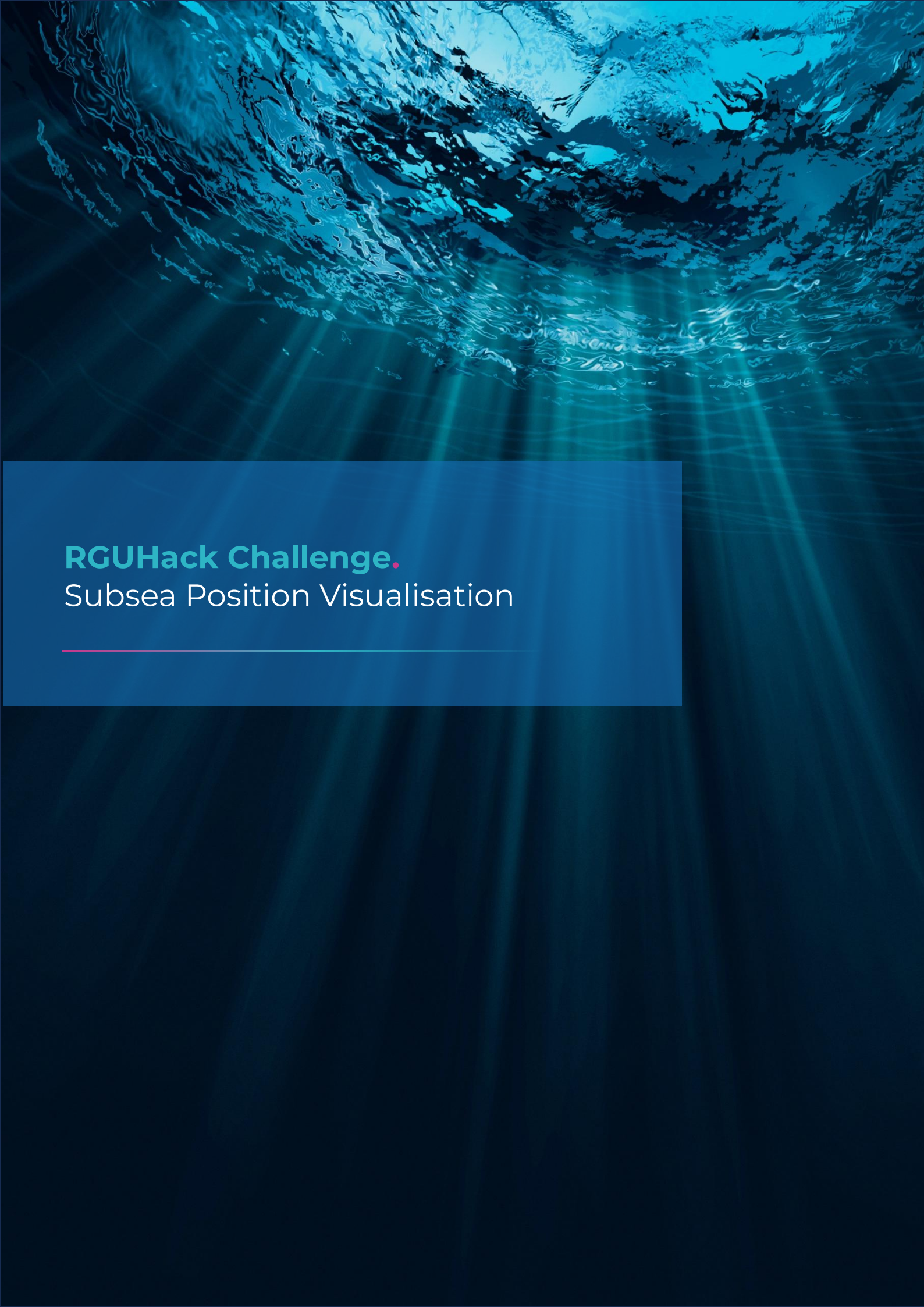

RGUHack Challenge.

Subsea Position Visualisation

1.0 Definition.

In this hackathon challenge, a fictional customer of Elementz has requested a visualisation that can show the geographical locations of their subsea assets and pipelines.

The customer has a "Google Earth"-style view in mind, showing a globe with the assets and pipelines plotted at their respective positions on planet Earth. A user should be able to navigate around the globe, zoom in on assets and see key information about them such as the asset name, its location and a count of open anomalies. This visualisation would serve as a “home page” in Elementz’s product, giving an “at a glance” summary of customer assets and pipelines.

The Elementz developers have started making an API that can return the required data. The API is available for you to use, but as this is still in a prototype stage, the endpoints return latitude and longitude coordinates. These will need to be converted to be displayed on a 3D globe as latitude and longitude are a 2D coordinate system. The logic required for this conversion is part of the hackathon challenge for you to implement.

The visualisation should be presented in a browser, but you are free to decide which technologies to use, for example, React, JavaScript, Three.js, Babylon.js.

2.0 Using The API

RGU are hosting the Elementz API at https://elementz.rguhack.uk.

If the API is not publicly available by RGU, you can use the Docker image to use the API locally.
You will need to download and install Docker Desktop if you don't have it already. A download link can be found here: https://docs.docker.com/get-started/get-docker/

When installation is complete, run this command to get the image:

```
docker pull edshearerelz/rguhack-elementz-api:latest
```

To run the API in the image, run this command:

```
docker run -p 8088:8088 edshearerelz/rguhack-elementz-api:latest
```

Note that "8088:8088" maps port 8088 in your device to port 8088 in the image. Please ensure that you don't have any other applications using port 8088 in your device or change the mapping to an available port.

3.0 API Endpoints.

The API provided by Elementz has 4 endpoints that will need to be called to make the visualization. If you are using the API hosted by RGU, the base URL is https://elementz.rguhack.uk, e.g. https://elementz.rguhack.uk/pointsOfInterest. If you are running the Docker image, the base URL will be http://localhost:8088, e.g. http://localhost:8088/pointsOfInterest. Change the port number in the URL if you had to change the port mapping.

GET /pointsOfInterest - returns an array of points of interest. Displaying these in your final solution is not essential but may be useful in the early stages of development. A couple of familiar locations are included to help you ensure your 2D coordinates to 3D coordinates conversation is correct.

GET /surfVessels - returns an array of subsea vessels with the latitude and longitude of each vessel. The heading is also required for displaying a vessel: a heading of zero degrees means the vessel is pointing at true north. 90 degrees is east, 180 degrees is south, and so on. The rest of the data in this response is for you to decide how to display.

GET /subseaAssets - returns an array of subsea assets with the latitude, longitude and depth of each asset.

GET /subseaPipelines - returns an array of subsea pipelines with the start and end latitude and longitude of each pipeline. Pipelines can be visualized as a line from start to end.

There are no other endpoints. As this API is a work in progress by Elementz developers, the data returned from them is static.

4.0 Visualisation Requirements.

- You will need to decide how to visualize points of interest, subsea vessels, subsea assets and subsea pipelines. A 2D image overlayed on the globe would suffice. Alternatively, if you have a 3D graphics artist in your team, you could make simple 3D models to represent each object.
- Images of subsea assets and vessels are easily found online. Subsea assets are complex structures of pipework, but there's no need to model this in detail. Basic geometry is fine!
- The objects should be coloured by type: for example, subsea assets should be yellow, vessels should be red, and so on. Feel free to change these colours as you see fit.
- Pipelines are not all the same length, so you will need to stretch or repeat a 3D model across the full length. You will also need to consider the curvature of the Earth. Please don't create a separate 3D model for each pipeline!

5.0 Optional Requirements.

- Add filters so a user can choose to see assets and pipelines based on their health, which can be either Healthy, Degraded, Critical, Offline, or Unknown.
- Add a filter so a user can choose to see only assets, only pipelines, or both.
- Add a toggle to see points of interest or not.
- Add a filter so that assets or pipelines that exceed a given number of anomalies are shown.
- Subsea assets and pipelines have depths. The depth is not essential for visualization - you can choose whether to visualize depth or show all subsea assets at sea level.
- Use Elementz brand colours for the globe and any popup windows when viewing asset and pipeline details.
- Animated 3D models. Don't go overboard here! For example, a subtle water ripple around vessels.
- Add floating Icons above vessels and subsea assets and pipelines that show their health.
- Effects like a glow around the globe may enhance the overall aesthetic of the visualization but should not be too distracting.
- Find a public weather API and display any weather alerts at their respective locations on the globe. Weather Is particularly important to a subsea inspector as severe weather can delay inspections.