Implementation Solution

I would like to propose the solution of using **Streamlit** to build a frontend dashboard for the stakeholders. It is a mature python package with built-in interactive UI components such as select box, slider, forms, etc. Engineers only needs to develop the main functions and codes, and then connect them to the UI components that will be presented to out stakeholders. The final web application can also be shared to anyone who wishes to play with it. With those advantages, it is expected to take less than 1 week's worth of work to design and deploy the application.

The way Streamlit works is straightforward. Once our stakeholders input some information they are interested in, for example in Question 2a of the Assessment they can simply select "Team2" and "T" in the select boxes provided in the dashboard, the information will be passed to the corresponding functions for processing, and the result will be returned on the dashboard within a very short amount of time. In this way, our stakeholder won't need to open terminals to run python scripts at all. Instead, all they need to do is to click some buttons or type some words in our app and wait for several seconds to obtain the information they need.

Here is a simple implementation code:

```
import streamlit as st


team = st.selectbox(
    'Please select the Team:',
    ['Team1', 'Team2'],)


side = st.selectbox(
    'Please select the Side:',
    ['T', 'CT'],)


# With the input information on team and side, we can extract corresponding rows
interest = data_df[(data_df['team'] == team) & (data_df['side'] == side)].copy()


# And then apply the check_boundary to each row and record result in a new column 'in_light_blue_zone'
boundary = [[-1735, 250], [-2024, 398], [-2806, 742], [-2472, 1233], [-1565, 580]]
interest['in_light_blue'] = interest.apply(lambda row: check_boundary(row, boundary), axis = 1)


# Other codes ......
```