front-end-coding

December 12, 2022

1 Front-end code

It might not work in colab, we upload it for making our project as a whole.

[ ]:

```
# anvil code of form1
from ._anvil_designer import Form1Template
from anvil import *
import anvil.server
import plotly.graph_objects as go
import time

class Form1(Form1Template):
  def __init__(self, **properties):
    # Set Form properties and Data Bindings.
    self.init_components(**properties)
    self.timer_1.interval = 10 * 3 # 60*3s
    self.refresh()
#     anvil.server.call('getBackground_task')



  def timer_1_tick(self, **event_args):
    print('---- refresh ----')
    self.refresh()

  def refresh(self):
    yCode = self.getYesterday()
    tCode = self.getToday()

  def getToday(self):
    tData, tCode = anvil.server.call('getT_Information')
    if tCode == 200:
      self.repeating_panel_2.items = tData.values()
    else:
      if tData != None and len(tData) > 0:
        self.repeating_panel_2.items = tData.values()
      else:
        print('error today')
    return tCode

  def getYesterday(self):
      yData, yCode = anvil.server.call('getY_Information')
      if yCode == 200:
        self.repeating_panel_1.items = yData.values()
      else:
        if yData != None and len(yData) > 0:
          self.repeating_panel_1.items = yData.values()
        else:
          print('error yesterday')
      return yCode
```

[ ]:

```
# panel1
from ._anvil_designer import ItemTemplate1Template
from anvil import *
import anvil.server

class ItemTemplate1(ItemTemplate1Template):
  def __init__(self, **properties):
    # Set Form properties and Data Bindings.
    self.init_components(**properties)
    pic_away = self.item['awayLogo']
    pic_home = self.item['homeLogo']
    name_away = self.item['fullAwayName']
    name_home = self.item['fullHomeName']
    self.pict_away1.source = pic_away
    self.pict_away2.source = pic_away
    self.pict_home1.source = pic_home
    self.pict_home2.source = pic_home
    self.awayname_label1.text = name_away
    self.awayname_label2.text = name_away
    self.homename_label1.text = name_home
    self.homename_label2.text = name_home
    self.away_predict1.text = self.item['pred_away_yesterday']
    self.home_predict1.text = self.item['pred_home_yesterday']
    self.away_real2.text = self.item['awayScore']
    self.home_real2.text = self.item['homeScore']
    self.result_yesterday.text = self.item['result_yesterday']
    # Any code you write here will run before the form opens.
```

[ ]:

```
# panel2
from ._anvil_designer import ItemTemplate2Template
from anvil import *
import anvil.server

class ItemTemplate2(ItemTemplate2Template):
  def __init__(self, **properties):
    # Set Form properties and Data Bindings.
    self.init_components(**properties)
    today_awayLogo = self.item['today_awayLogo']
    today_homeLogo = self.item['today_homeLogo']
    today_awayName = self.item['today_fullHomeName']
    today_homeName = self.item['today_fullAwayName']
    self.image_away1.source = today_awayLogo
    self.image_away2.source = today_awayLogo
    self.image_home1.source = today_homeLogo
    self.image_home2.source = today_homeLogo
    self.name_away1.text = today_awayName
    self.name_away2.text = today_awayName
    self.name_home1.text = today_homeName
    self.name_home2.text = today_homeName
    self.goal_away2.text = self.item['today_awayScore']
    self.goal_home2.text = self.item['today_homeScore']
    self.forecast_away1.text = self.item['pred_away_today']
    self.forecast_home1.text = self.item['pred_home_today']
    self.result_today.text = self.item['result_today']
    # Any code you write here will run before the form opens.
```