**راهنمای راه اندازی سامانه مدیریتی با فریم ورک دش (Dash)**

**دکتر فریدی – مهندس عظیمی**

**ساخت محیط مجازی و نصب کتابخانه های مورد نیاز**

در مسیر دلخواه وارد خط فرمان شوید و دستور العمل زیر را اجرا کنید.

ساخت محیط مجاری:

Python -m venv DASHenv

Cd DASHenv\Scripts\Activate

نصب کتابخانه ها:

Pip install dash

pip install dash\_bootstrap\_components

**ساختن پروژه**

وارد مسیر دلخواه شده و یک پوشه برای پروژه خود بسازید. در این مستند ما از نام Dash Dashboard Adminlte برای نام اصلی پروژه استفاهد می‌کنیم.

**ساختار پروژه**

یک پوشه برای قرار دادن فایل های استاتیک مثل css, js, images, … در مسیر root پروژه می‌سازیم و یک پوشه دیگر برای قرار دادن tab‌ها جهت ماژولار کردن ساخته می‌شود.

Dash Dashboard Adminlte/

-------/assets/

-----------------/css/

-------------------------/webfonts/ # download webfonts folder from icons

-------------------------/bootstrap.rtl.min.css

-------------------------/all.min.css # fonts

----------------/custom.css

-------/pages/

---------------/tab\_01.py

-------/app.py

**نمونه Tab**

برای هر tab باید یک MODULE\_NAME تعریف شود دقیقا به نام همان tab یعنی اگر نام فایل پایتونی tab\_01 باشد، باید متغیر MODULE\_NAME به شکل زیر تعریف شود.

MODULE\_NAME = "tab\_01"

و هر قسمتی که قرار است id استفاده شود باید به صورت زیر اعمال شود.

id=ns('layout')

Pages/tab\_01.py

from dash import html, dcc, Input, Output

import dash\_admin\_components as dac

MODULE\_NAME = "tab\_01"

def ns(id\_):

    return f"{MODULE\_NAME}\_{id\_}"

layout = dac.TabItem(id=ns('layout'),

    children=[

        html.H4('نمونه استفاده از کارت DAC'),

        html.Div([

            dac.ValueBox(

                value="بخش 1",

                subtitle="زیر بخش 1",

                color = "primary",

                icon = "shopping-cart",

                href = "#"

            )

        ], className='row'),

    ]

)

جهت استفاده از دکوراتور @callback باید در همان tab مربوطه از قظعه کد زیر به شکل دلخواه استفاده کنید.

به عنوان نمونه یک تب دیگر ایجاد می‌کنیم و در آن از callback استفاده می‌کنیم.

Pages/tab\_02.py

from dash import html, dcc, Input, Output

import dash\_admin\_components as dac

MODULE\_NAME = "tab\_02"

def ns(id\_):

    return f"{MODULE\_NAME}\_{id\_}"

layout = html.Div([

    dcc.Input(id=ns('input'), value=f'Enter a number: {ns('input')}', type='text'),

    html.Div(id=ns('output'))

])

def register\_callbacks(app):

    @app.callback(

        Output(ns('output'), 'children'),

        Input(ns('input'), 'value')

    )

    def update\_output(input\_data):

        try:

            return str(float(input\_data) \*\* 2)

        except:

            return "Error, the input is not a number"

**فایل اجرایی برنامه**

فایل اجرایی برنامه همان app.py می‌باشدو اعمال زیر مهمترین وظایف این فایل است.

* تمام فایل‌های درون پوشه Pages معرفی می‌شوند.
* آدرس فایل‌های استایک قرار می‌گیرد.
* منوبار و منو کناری ساخته می‌شوند.
* برنامه با دستور run اجرا می‌شود.

App.py

import dash

from dash import html, dcc, Input, Output

import dash\_bootstrap\_components as dbc

import importlib

import os

import types

# Dash

app = dash.Dash(

    \_\_name\_\_,

    suppress\_callback\_exceptions=True,

    external\_stylesheets=[

        "/assets/css/bootstrap.rtl.min.css",

        "/assets/css/all.min.css",

        "/assets/custom.css"

    ]

)

app.title = "داشبورد "

# load pages (import) and call callback decorator for any app

PAGES\_DIR = "pages"

pages = {}

for file in os.listdir(PAGES\_DIR):

    if file.endswith(".py") and file != "\_\_init\_\_.py":

        module\_name = file[:-3]  # فقط نام فایل بدون پسوند

        full\_module\_name = f"{PAGES\_DIR}.{module\_name}"

        module = importlib.import\_module(full\_module\_name)

        # فرض بر این است که layout در ماژول تعریف شده

        pages["/" + module\_name] = module.layout

        # اگر تابع register\_callbacks وجود دارد، آن را صدا بزن

        if hasattr(module, "register\_callbacks"):

            module.register\_callbacks(app)

# Sidebar items

sidebar\_items = [

    {"id": "tab\_01", "label": "تب اول", "icon": "fa fa-home"},

    {"id": "tab\_02", "label": "تب دوم", "icon": "fa fa-chart-line"},

]

# Layout

app.layout = html.Div([

    dcc.Location(id="url", refresh=False),

    # Navbar

    dbc.Navbar(

        children=[

            dbc.Button(

                html.I(className="fa fa-bars"),

                id="btn\_sidebar",

                color="secondary",

                className="m-2"

            ),

            dbc.NavbarBrand("داشبورد آزمایشگاه علم داده ", className="me-auto"),

            dbc.Nav(

                [

                    dbc.NavItem(html.I(className="fa fa-user mt-3")),

                    dbc.NavItem(dbc.NavLink("جهت استفاده لوکال", href="#")),

                ],

                className="m-2",

                navbar=True

            )

        ],

        color="dark",

        fixed="top"

    ),

    # Sidebar

    html.Div(

        id="sidebar",

        children=[

            html.Div(

                [html.I(className=item["icon"]), html.Span(item["label"], className="ms-2")],

                id=item["id"],

                className="sidebar-item"

            )

            for item in sidebar\_items

        ]

    ),

    # Page content

    html.Div(id="page-content",)

])

# Toggle Sidebar

@app.callback(

    Output("sidebar", "className"),

    Output("page-content", "style"),

    Input("btn\_sidebar", "n\_clicks"),

    [dash.dependencies.State("sidebar", "className")]

)

def toggle\_sidebar(n, current\_class):

    if not current\_class:

        current\_class = ""

    collapsed = "collapsed" not in current\_class if n else "collapsed" in current\_class

    class\_name = "collapsed" if collapsed else ""

    style = {"marginRight": "80px" if collapsed else "250px", "transition": "margin 0.3s", "paddingTop": "70px"}

    return class\_name, style

@app.callback(

    Output("page-content", "children"),

    [Input(item["id"], "n\_clicks") for item in sidebar\_items]

)

def display\_page(\*args):

    ctx = dash.callback\_context

    if not ctx.triggered:

        return pages.get("/tab\_01")

    button\_id = ctx.triggered[0]["prop\_id"].split(".")[0]

    return {

        "tab\_01": pages.get("/tab\_01"),

        "tab\_02": pages.get("/tab\_02"),

    }.get(button\_id, pages.get("/tab\_01"))

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

**اجرای برنامه**

پروژه را اجرا کنید

Python app.py

پروژه رو یک مرورگر با آدرس <http://127.0.0.1:8050> در دسترس است.

