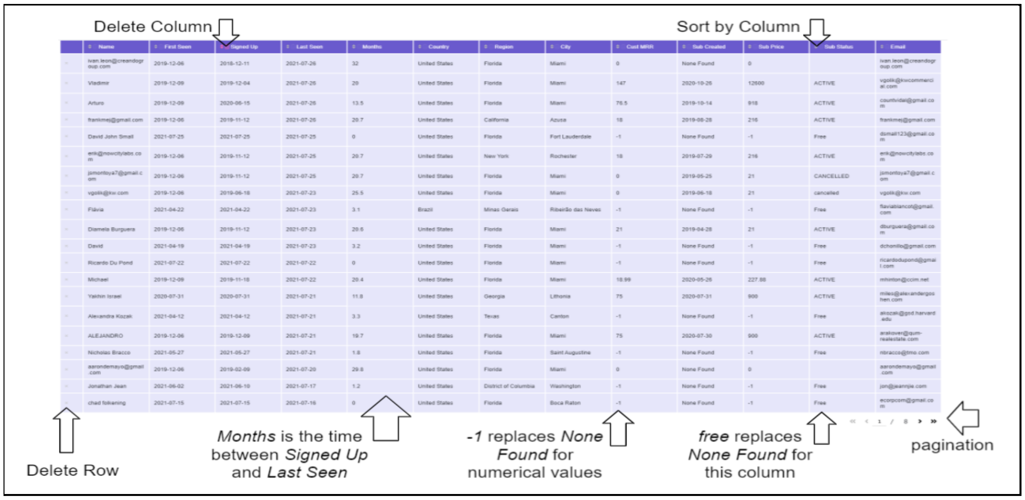
Deepblocks Dashboard Guide:

Jonah & Yuhan, Interns Summer 2021

Dash Link: https://deepblocks-dashboard.herokuapp.com/

Functionalities

Data Table


Delete Column
Sort by Column
Delete Row
Months is the time between *Signed Up* and *Last Seen*
-1 replaces *None Found* for numerical values
free replaces *None Found* for this column
pagination
« ‹ 1 / 8 › »

	Name	First Seen	Signed Up	Last Seen	Months	Country	Region	City	Cust MRR	Sub Created	Sub Price	Sub Status	Email
×	ivan.leon@creandogroup.com	2019-12-06	2018-12-11	2021-07-26	32	United States	Florida	Miami	0	None Found	0		ivan.leon@creandogroup.com
×	Vladimir	2019-12-09	2019-12-04	2021-07-26	20	United States	Florida	Miami	147	2020-10-26	12600	ACTIVE	vgolik@kwcommercial.com
×	Arturo	2019-12-09	2020-06-15	2021-07-26	13.5	United States	Florida	Miami	76.5	2019-10-14	918	ACTIVE	countvidal@gmail.com
×	frankmej@gmail.com	2019-12-06	2019-11-12	2021-07-26	20.7	United States	California	Azusa	18	2019-08-28	216	ACTIVE	frankmej@gmail.com
×	David John Small	2021-07-25	2021-07-25	2021-07-25	0	United States	Florida	Fort Lauderdale	-1	None Found	-1	Free	dsmall123@gmail.com
×	erik@nowcitylabs.com	2019-12-06	2019-11-12	2021-07-25	20.7	United States	New York	Rochester	18	2019-07-29	216	ACTIVE	erik@nowcitylabs.com
×	jsmontoya7@gmail.com	2019-12-06	2019-11-12	2021-07-25	20.7	United States	Florida	Miami	0	2019-05-25	21	CANCELLED	jsmontoya7@gmail.com
×	vgolik@kw.com	2019-12-06	2019-06-18	2021-07-23	25.5	United States	Florida	Miami	0	2019-06-18	21	cancelled	vgolik@kw.com
×	Flávia	2021-04-22	2021-04-22	2021-07-23	3.1	Brazil	Minas Gerais	Ribeirão das Neves	-1	None Found	-1	Free	flaviabiancot@gmail.com
×	Diamela Burguera	2019-12-06	2019-11-12	2021-07-23	20.6	United States	Florida	Miami	21	2019-04-28	21	ACTIVE	dburguera@gmail.com
×	David	2021-04-19	2021-04-19	2021-07-23	3.2	United States	Florida	Miami	-1	None Found	-1	Free	dchonillo@gmail.com
×	Ricardo Du Pond	2021-07-22	2021-07-22	2021-07-22	0	United States	Florida	Miami	-1	None Found	-1	Free	ricardodupond@gmail.com
×	Michael	2019-12-09	2019-11-18	2021-07-22	20.4	United States	Florida	Miami	18.99	2020-05-26	227.88	ACTIVE	mhinton@ccim.net
×	Yakhin Israel	2020-07-31	2020-07-31	2021-07-21	11.8	United States	Georgia	Lithonia	75	2020-07-31	900	ACTIVE	miles@alexandergoshen.com
×	Alexandra Kozak	2021-04-12	2021-04-12	2021-07-21	3.3	United States	Texas	Canton	-1	None Found	-1	Free	akozak@gsd.harvard.edu
×	ALEJANDRO	2019-12-06	2019-12-09	2021-07-21	19.7	United States	Florida	Miami	75	2020-07-30	900	ACTIVE	arakover@gum-realestate.com
×	Nicholas Bracco	2021-05-27	2021-05-27	2021-07-21	1.8	United States	Florida	Saint Augustine	-1	None Found	-1	Free	nbracco@tmo.com
×	aarondemayo@gmail.com	2019-12-06	2019-02-09	2021-07-20	29.8	United States	Florida	Miami	0	None Found	0		aarondemayo@gmail.com
×	Jonathan Jean	2021-06-02	2021-06-10	2021-07-17	1.2	United States	District of Columbia	Washington	-1	None Found	-1	Free	jon@jeannjie.com
×	chad folkening	2021-07-15	2021-07-15	2021-07-16	0	United States	Florida	Boca Raton	-1	None Found	-1	Free	ecorpcom@gmail.com

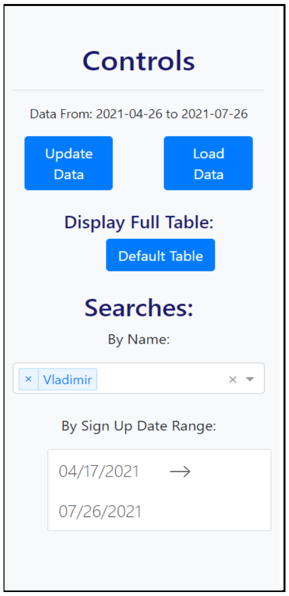
Recent addition: Export to CSV button (top left)

Controls Sidebar

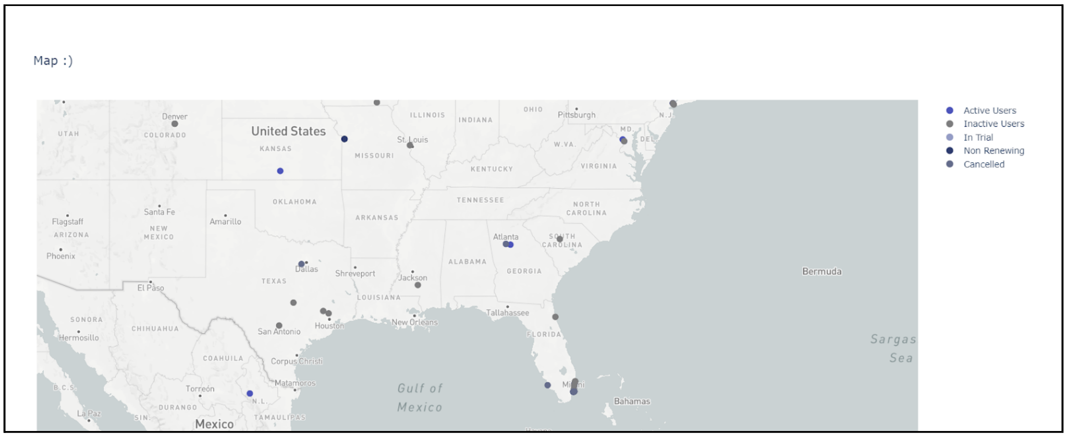
- All options in the sidebar manipulate the data table.
- To hide the sidebar and see the full table and graphs, use the horizontal scroll.

- By default, 150 users appear on the table upon loading.
  - Upon opening the dashboard, load all users in database by using *Load Data* button.
    - This button can only be pressed once.
  - Use *Default Table* button to show the table upon making any load or search changes.

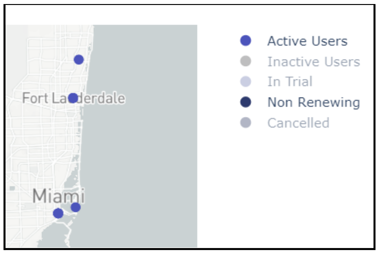
- The *Update Data* button will pull recent data from Intercom
- Searches, including name and date range, can be used to filter the table.
  - *Default table* button will reset the table after filtering. (May need to be pressed *multiple times* to load all pages)



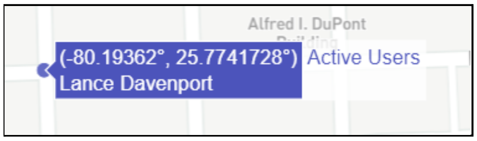
Map



- Map displays the 100 most recent users' subscription status.
    - User locations are based off of city/region coordinates from Intercom



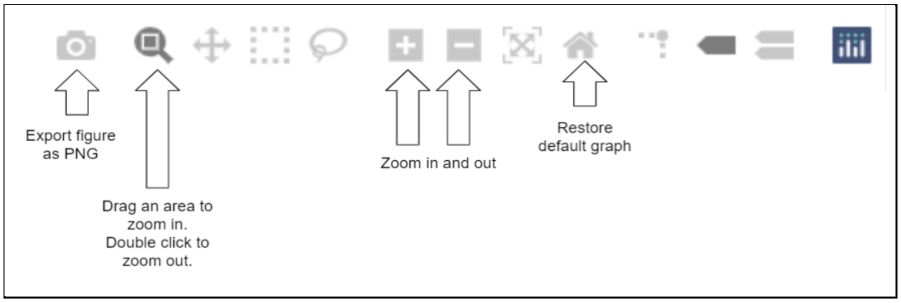
- When first opening the dashboard, use *load map* button to display the map.
    - Double click a dot on legend to isolate a single status.
        - Double click again to show all.


(-80.19362°, 25.7741728°) Active Users
Lance Davenport

- Hover over point to show subscription status.
- Export PNG option available in the toolbar at the upper left hand corner.

Plotly Built-in Toolbar

- Hover over graph to show on right hand corner


Export figure as PNG

Drag an area to zoom in. Double click to zoom out.

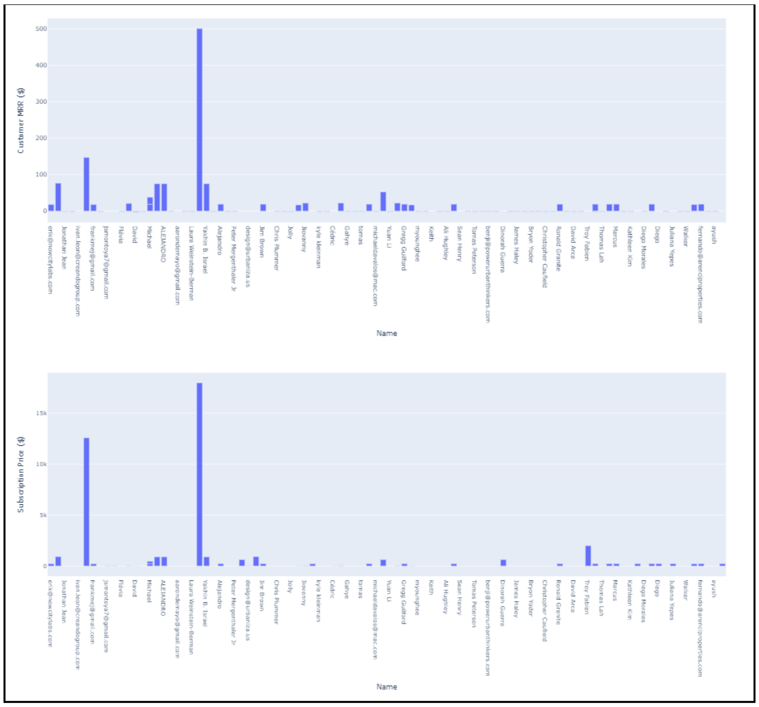
Zoom in and out

Restore default graph

MRR & Plan price Bar Graphs

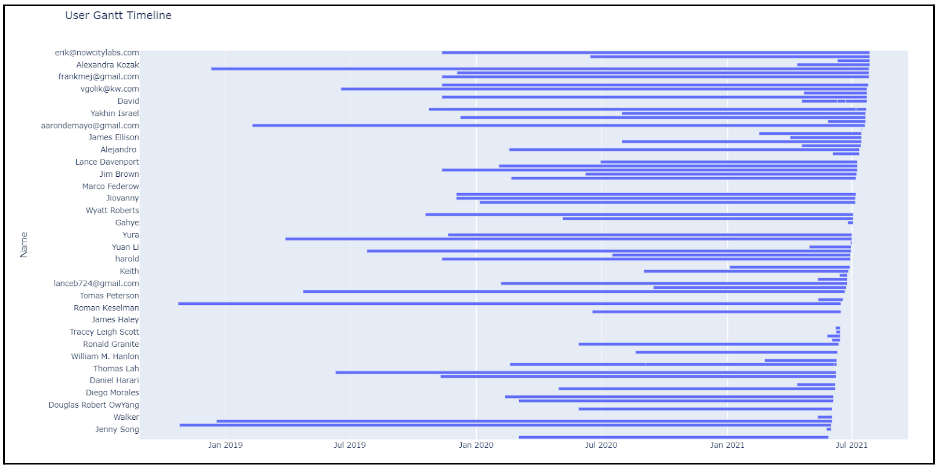
- Hover over bars to show respective values

- Because of spacing, the names on the x-axis may not correlate to the bars; this discrepancy can be clarified by zooming in.



User Gantt Timeline

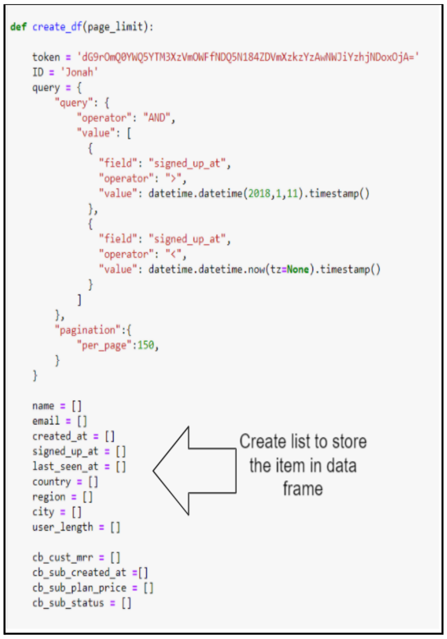
- Displays the duration of time from the users' sign-up date to the date in which they were last seen. Hover over the bar to show exact dates.
- The timeline is ordered by the most recent users.



Python: Dash and Plotly

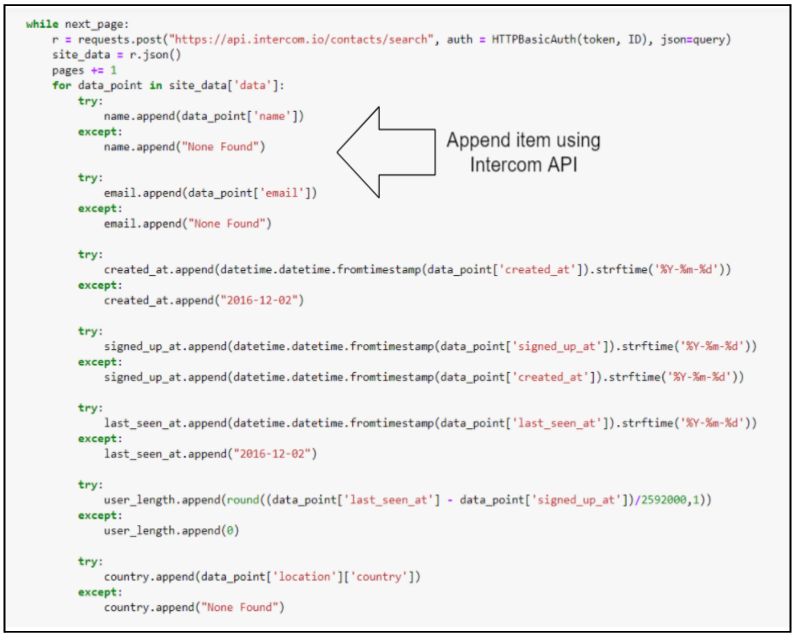
- The dashboard utilizes Plotly's Dash (including Bootstrap and Core components) as well as Pandas.

Adding Columns to Data Table from Intercom



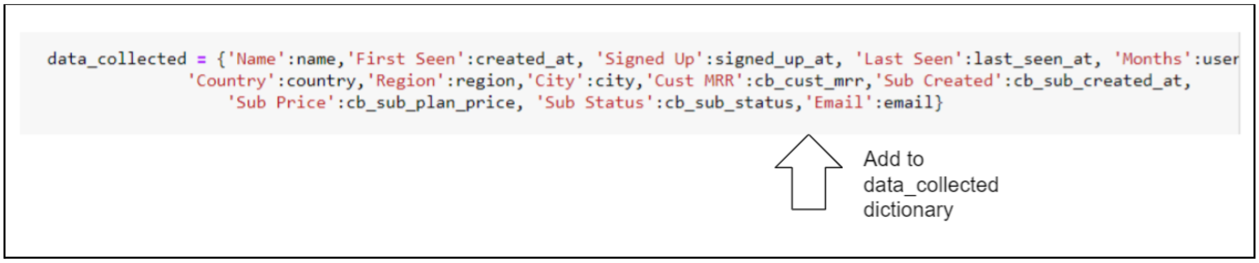
```python
def create_df(page_limit):

    token = 'dG9rOmQ0YWQ5YTM3XzVmOWFfNDQ5N184ZDVmXzkzYzAwNWJiYzhjNDoxOjA='
    ID = 'Jonah'
    query = {
        "query": {
            "operator": "AND",
            "value": [
              {
                "field": "signed_up_at",
                "operator": ">",
                "value": datetime.datetime(2018,1,11).timestamp()
              },
              {
                "field": "signed_up_at",
                "operator": "<",
                "value": datetime.datetime.now(tz=None).timestamp()
              }
            ]
        },
        "pagination":{
            "per_page":150,
        }
    }

    name = []
    email = []
    created_at = []
    signed_up_at = []
    last_seen_at = []
    country = []
    region = []
    city = []
    user_length = []

    cb_cust_mrr = []
    cb_sub_created_at =[]
    cb_sub_plan_price = []
    cb_sub_status = []
```

Create list to store the item in data frame



```python
while next_page:
    r = requests.post("https://api.intercom.io/contacts/search", auth = HTTPBasicAuth(token, ID), json=query)
    site_data = r.json()
    pages += 1
    for data_point in site_data['data']:
        try:
            name.append(data_point['name'])
        except:
            name.append("None Found")

        try:
            email.append(data_point['email'])
        except:
            email.append("None Found")

        try:
            created_at.append(datetime.datetime.fromtimestamp(data_point['created_at']).strftime('%Y-%m-%d'))
        except:
            created_at.append("2016-12-02")

        try:
            signed_up_at.append(datetime.datetime.fromtimestamp(data_point['signed_up_at']).strftime('%Y-%m-%d'))
        except:
            signed_up_at.append(datetime.datetime.fromtimestamp(data_point['created_at']).strftime('%Y-%m-%d'))

        try:
            last_seen_at.append(datetime.datetime.fromtimestamp(data_point['last_seen_at']).strftime('%Y-%m-%d'))
        except:
            last_seen_at.append("2016-12-02")

        try:
            user_length.append(round((data_point['last_seen_at'] - data_point['signed_up_at'])/2592000,1))
        except:
            user_length.append(0)

        try:
            country.append(data_point['location']['country'])
        except:
            country.append("None Found")
```

Append item using Intercom API

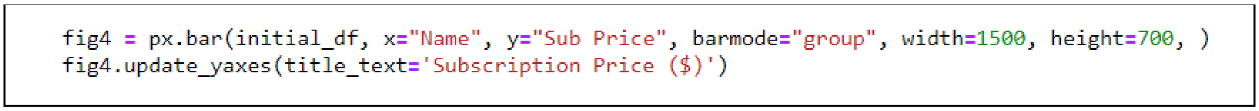


```python
data_collected = {'Name':name,'First Seen':created_at, 'Signed Up':signed_up_at, 'Last Seen':last_seen_at, 'Months':user
                  'Country':country,'Region':region,'City':city,'Cust MRR':cb_cust_mrr,'Sub Created':cb_sub_created_at,
                      'Sub Price':cb_sub_plan_price, 'Sub Status':cb_sub_status,'Email':email}
```

Add to data_collected dictionary

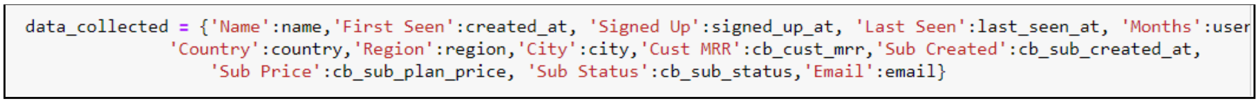
Adding Figures to the Dashboard

Fig 1.



```
fig4 = px.bar(initial_df, x="Name", y="Sub Price", barmode="group", width=1500, height=700, )
fig4.update_yaxes(title_text='Subscription Price ($)')
```

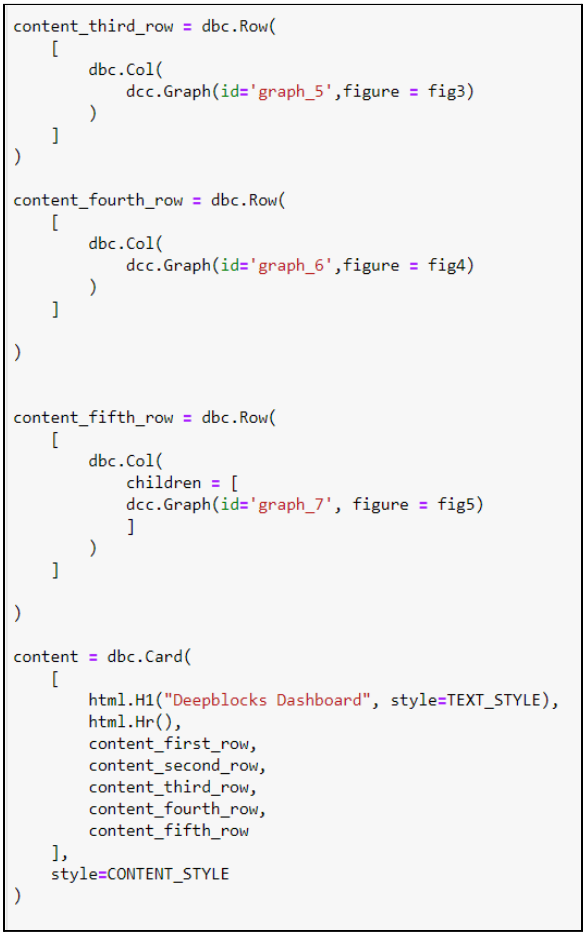
Fig 2.



```
data_collected = {'Name':name,'First Seen':created_at, 'Signed Up':signed_up_at, 'Last Seen':last_seen_at, 'Months':user
                  'Country':country,'Region':region,'City':city,'Cust MRR':cb_cust_mrr,'Sub Created':cb_sub_created_at,
                       'Sub Price':cb_sub_plan_price, 'Sub Status':cb_sub_status,'Email':email}
```

- Dashboard figures are numbered and created using various Plotly graph objects. (Fig 1.)
- Basic plotly chart implementations can be found here: https://plotly.com/python/basic-charts/
- To append data to a graph, use list names in the *data_collected* dictionary. (Fig 2.)
- Dashboard contents are separated into rows using Bootstrap. (Fig 3.)
- The content card holds all of the rows of figures. (Fig 3.)

Fig 3.



```
content_third_row = dbc.Row(
    [
        dbc.Col(
            dcc.Graph(id='graph_5',figure = fig3)
        )
    ]
)

content_fourth_row = dbc.Row(
    [
        dbc.Col(
            dcc.Graph(id='graph_6',figure = fig4)
        )
    ]

)


content_fifth_row = dbc.Row(
    [
        dbc.Col(
            children = [
            dcc.Graph(id='graph_7', figure = fig5)
            ]
        )
    ]

)

content = dbc.Card(
    [
        html.H1("Deepblocks Dashboard", style=TEXT_STYLE),
        html.Hr(),
        content_first_row,
        content_second_row,
        content_third_row,
        content_fourth_row,
        content_fifth_row
    ],
    style=CONTENT_STYLE
)
```