



Sumit Mehta

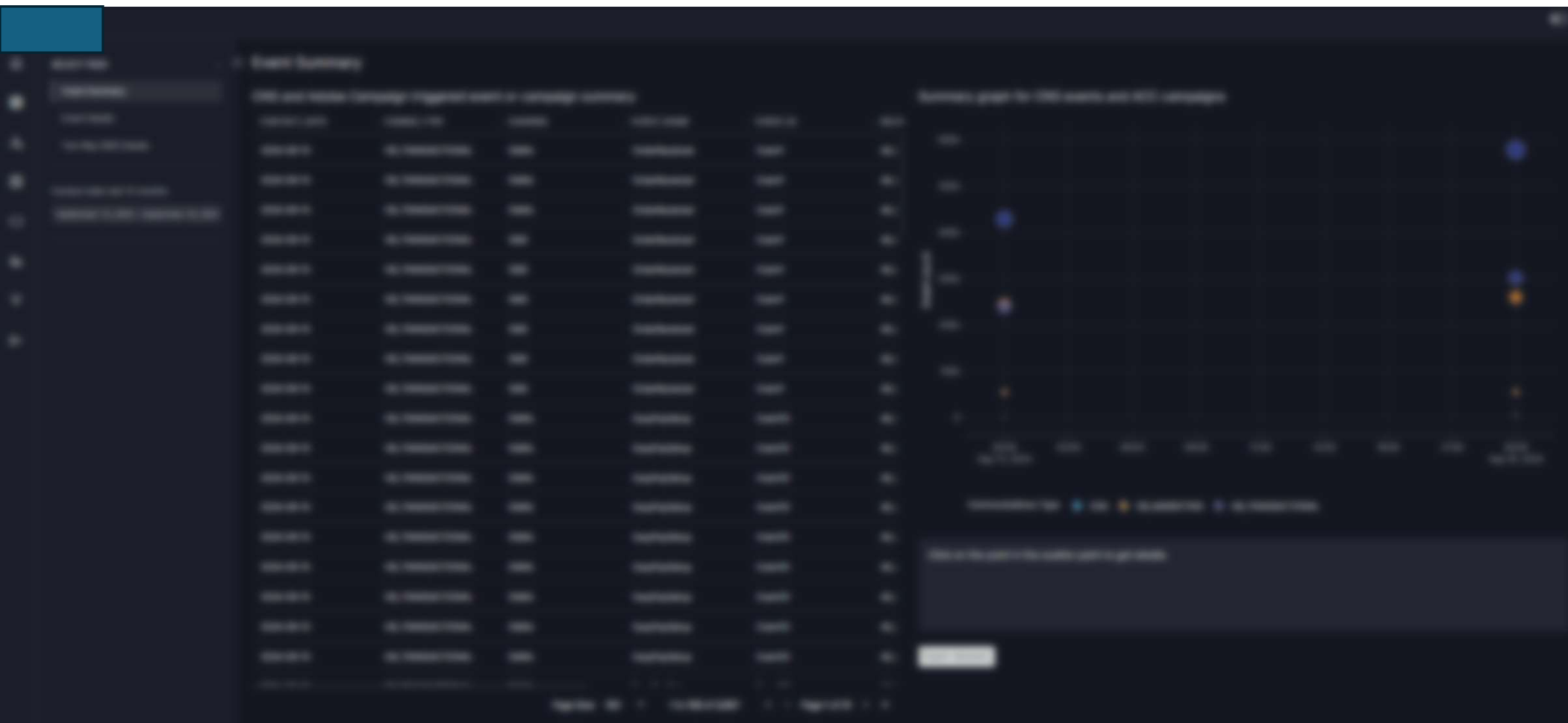
Dashboard example

- Created for strategy and product owners
- Functions:
 - Provides yearly summary
 - Ability to view data for specific time window
 - Ability to send email and SMS proof for a specific template
- Built using:
 - Python
 - Plotly Dash and Vizro libraries
 - Snowflake queries and CSV utility
 - PostgreSQL
 - AWS Lambda and step functions for data manipulation
 - Hosted on AWS Elastic Beanstalk

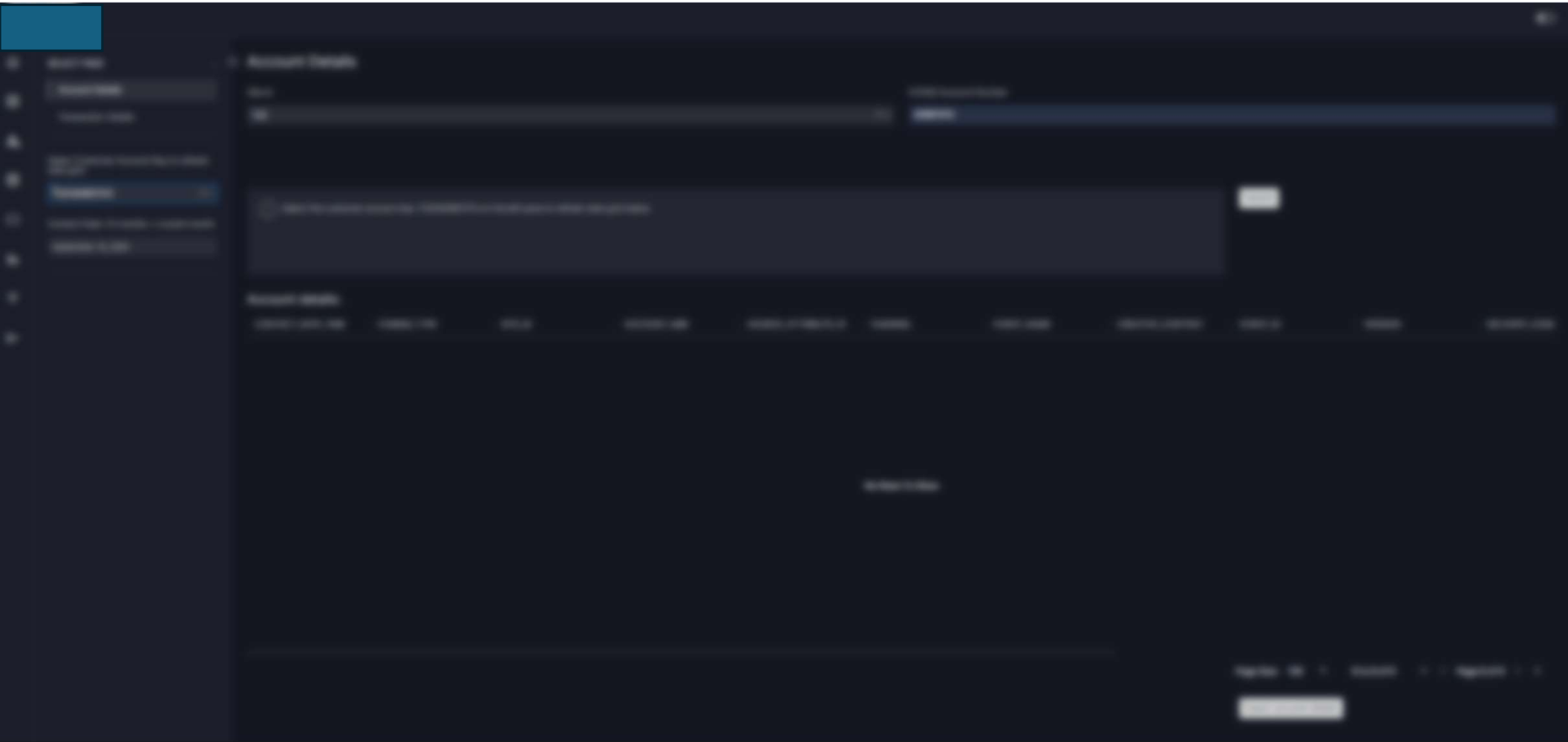
Dashboard – main screen with different pages listed out as clickable cards



Dashboard – ability to view data and generate graphs for side-by-side view



Dashboard – user input for a specific data point and generate data grid



Dashboard – data view for specific time and ability to interact with data by generating multiple view



Code – application.py, total lines of code 2000, provide 12 different pages

```
import os
import pandas as pd
import vizro.models as vm
from vizro import Vizro
from vizro.managers import data_manager
from vizro.models.types import capture
from typing import List, Literal
import dash_bootstrap_components as dbc
from vizro.tables import dash_ag_grid

try:
    from pydantic.v1 import Field
except ImportError: # pragma: no cov
    from pydantic import Field

from vizro.models import Action, VizroBaseModel
from vizro.models._action._actions_chain import _action_validator_factory
from vizro.models._models_utils import _log_call
from df_account_data_prep import *
from get_s3_mirror_page_image import *

os.environ["DBUS_SESSION_BUS_ADDRESS"] = '/dev/null'

# incorporate data
# df_prep_yearly_summary = df_prep("yearly_summary", '', '')

def load_graph_yearly_summary_data():
    graph_summary_data_charts = df_prep(selected_option="yearly_summary", event_number_name='', env_selected='').groupby(
        by=["FIRST_DAY_MONTH", "CHANNEL"]).agg(
            {"EVENT_COUNT": "sum"}).reset_index()
    graph_summary_data_charts["EVENT_COUNT"] = pd.to_numeric(graph_summary_data_charts["EVENT_COUNT"])
    return graph_summary_data_charts

data_manager["graph_summary_data_charts"] = load_graph_yearly_summary_data

default_404 = vm.Page(
    title="Default 404",
    components=[
        vm.Card(
```

13 9 52

1:1 CRLF UTF-8 4 spaces Python 3.11 (martech_app_1) (2)