



# PORTFOLIO

## DATA ANALYST

---

Welcome to my Portfolio! I am Sobia Khanam, a Data Analyst with a passion for uncovering insights and driving data-driven decision-making. Explore my work and see how I can help you leverage data to achieve your business goals.



### **LinkedIn**

<https://www.linkedin.com/in/sobia-khanam/>

### **Gmail**

[sobiakhanam2000@gmail.com](mailto:sobiakhanam2000@gmail.com)

### **GitHub**

<https://github.com/SobiaKhanam>

### **Portfolio**

<https://sobiakhanam.github.io/>

# ABOUT ME

## WHO I AM

With over 2 years of experience in data analysis, I have a proven track record of turning raw data into meaningful information. My analytical skills, combined with a deep understanding of various industries, allow me to provide valuable insights that drive business success.

## SKILLS & EXPERTISE

- ❖ Data Cleaning & Preparation
- ❖ Statistical Analysis
- ❖ Data Visualization
- ❖ SQL & Database Management
- ❖ Python for Data Analysis
- ❖ Machine Learning
- ❖ Business Intelligence Tool (Power BI)

# PORTFOLIO

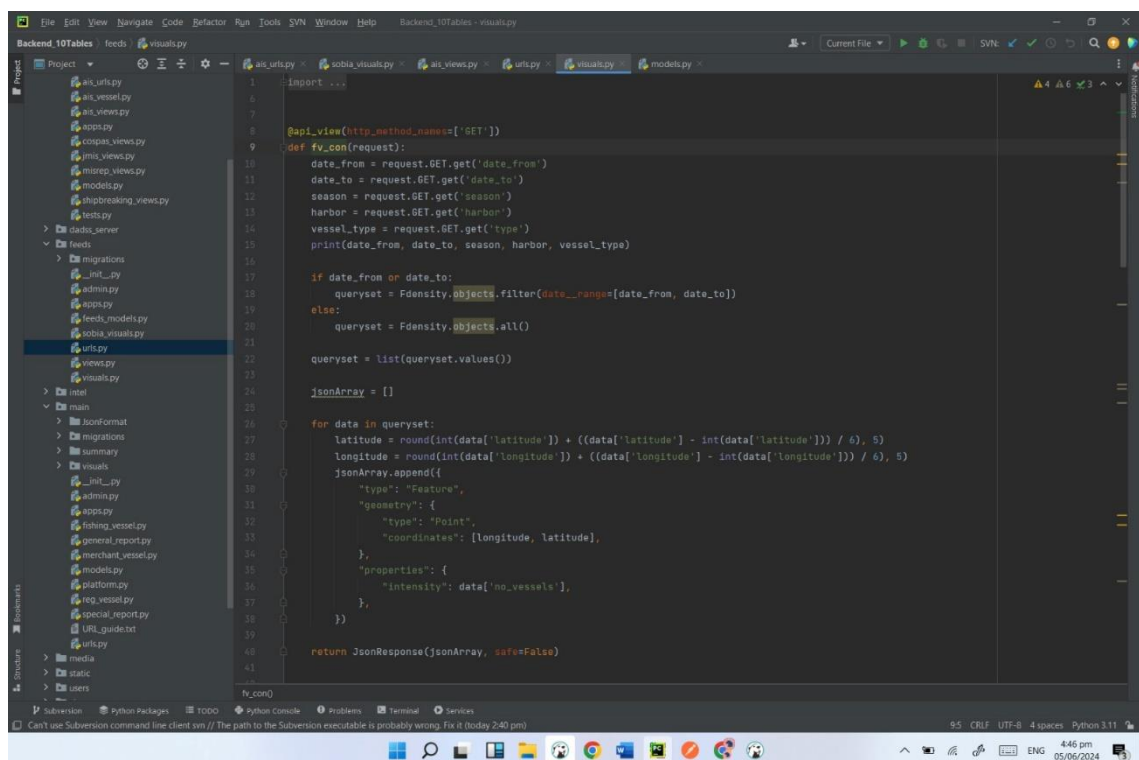
## HIGHLIGHTS

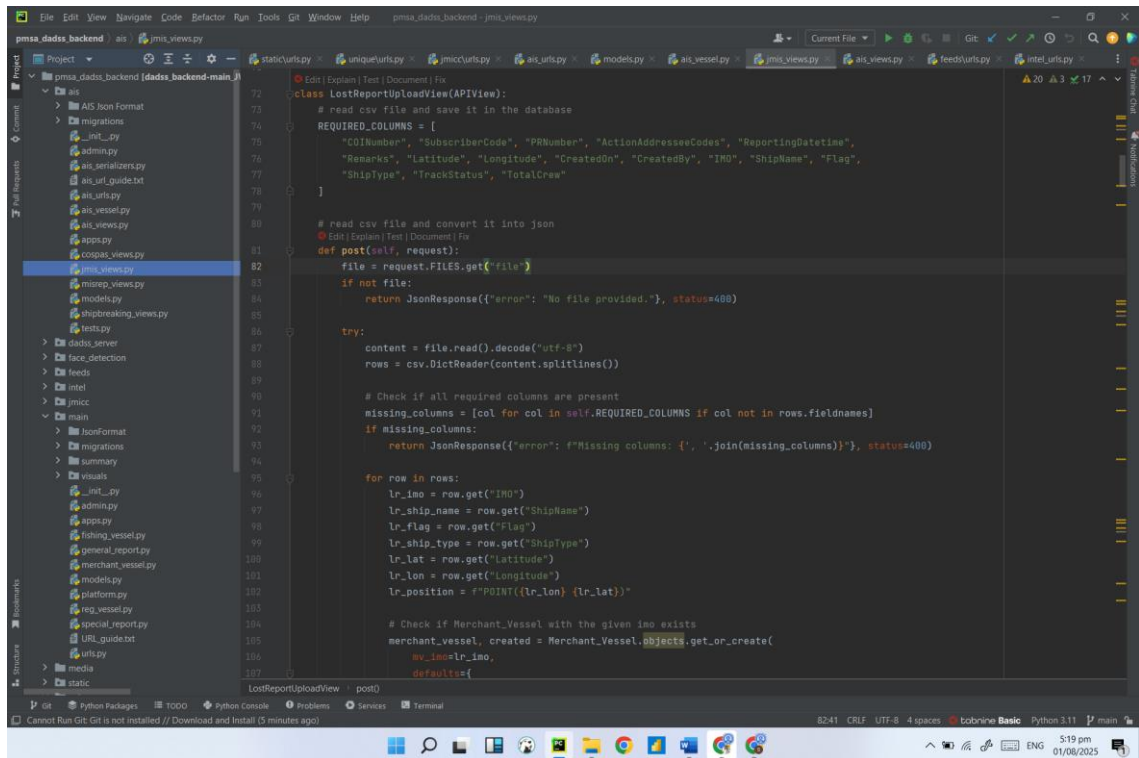
### DATA ANALYSIS & DECISION SUPPORT SYSTEM (DADSS)

The DADSS project was designed to automate the collection, storage, analysis, and visualization of large-scale maritime data. The system integrates multiple data sources, applies rigorous data processing, and delivers actionable insights through advanced GIS analysis and dashboards. This end-to-end solution eliminates manual workflows and provides stakeholders with real-time, location-based intelligence. The key steps in this project include:

#### 1. Backend Development:

- Designed robust backend architecture in Django to handle dynamic data pipelines.
- Created RESTful APIs for seamless data integration and external service communication.

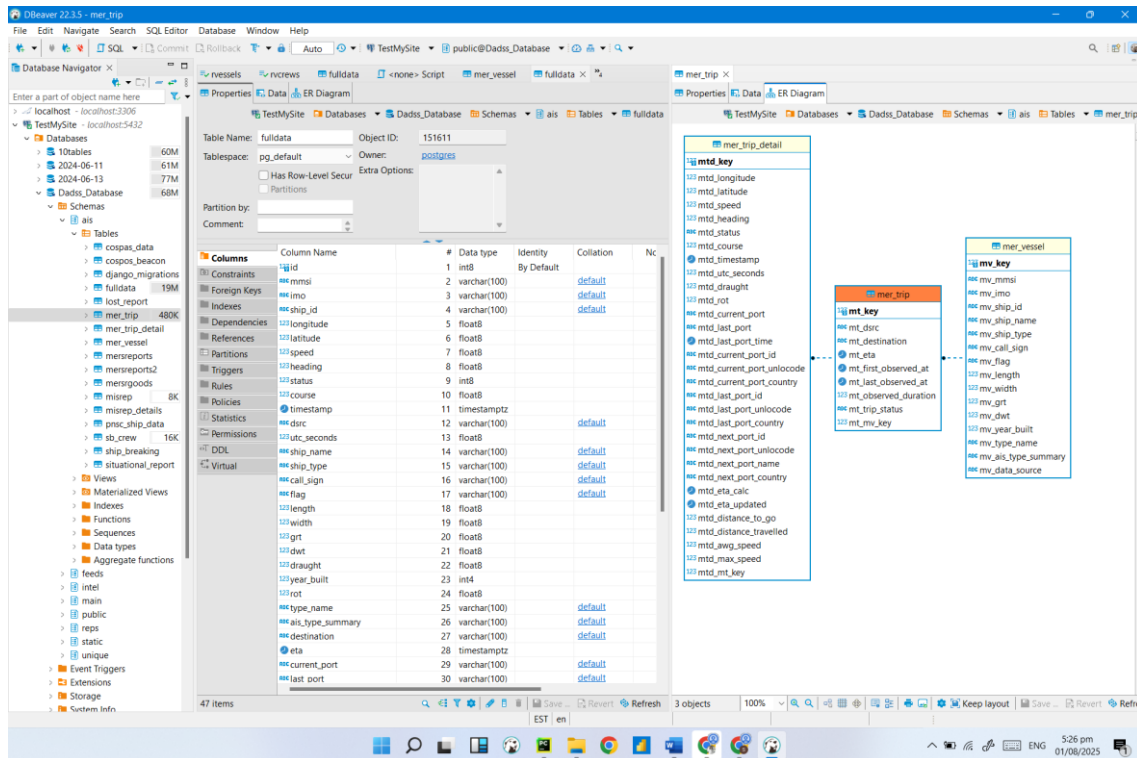




## 2. Database Design & Management:

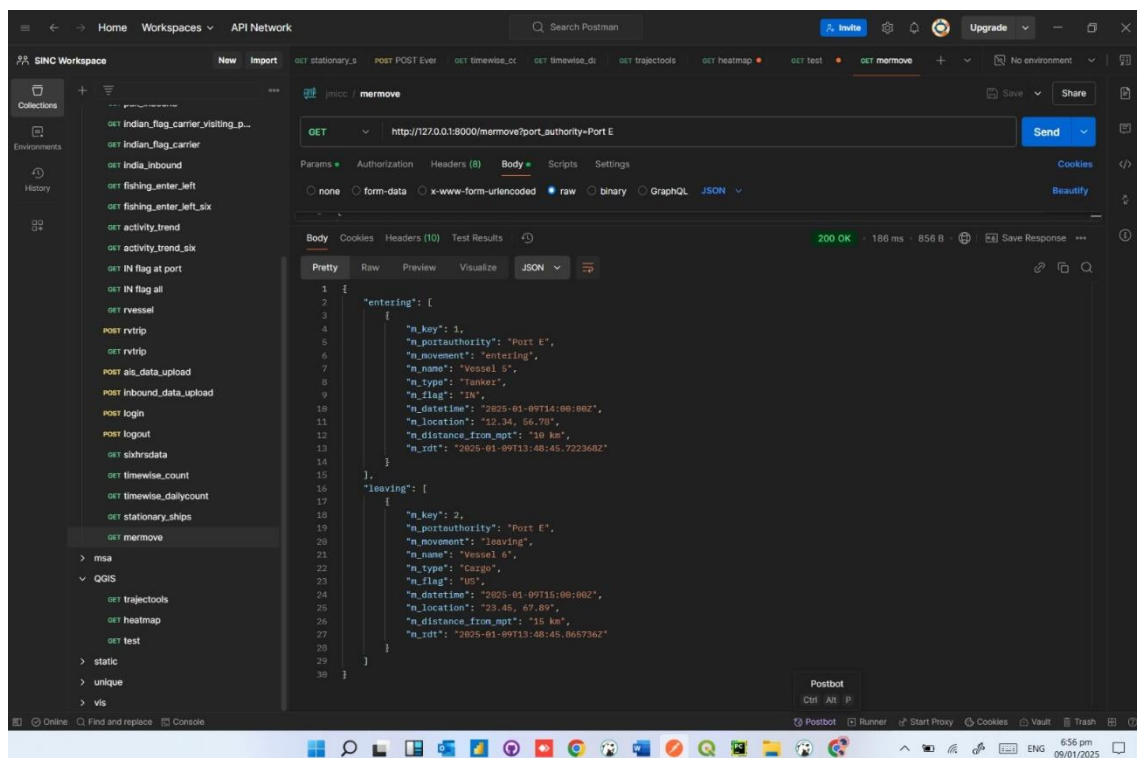
- Developed normalized PostgreSQL database schemas optimized for large datasets.
- Implemented PostGIS for geospatial data storage and querying.

id	msmsl	imo	ship_id	longitude	latitude	speed	heading	status	course	timestamp	dsr
1	356540000	9182497	306592	60.91991	24.39538	50	511	0	6	2023-08-16 02:21:51.000 -0400	TER
2	257654000	9493975	307989	62.71	24.56667	110	511	0	270	2023-08-16 02:17:37.000 -0400	SAT
3	463033101	9382360	678334	66.98333	24.84332	0	67	3	139	2023-08-16 02:20:26.000 -0400	TER
4	636013530	9256224	754136	67.03515	24.54437	65	25	0	22	2023-08-16 02:22:07.000 -0400	TER
5	538010770	9464314	755244	66.97701	24.83517	8	36	0	76	2023-08-16 02:21:49.000 -0400	TER
6	636016404	9288526	756269	61.16451	23.5176	9	283	2	34	2023-08-16 02:21:47.000 -0400	TER
7	677015100	9147447	756335	66.01907	24.04368	11	511	0	292	2023-08-16 02:20:49.000 -0400	TER
8	304939000	9329643	757845	66.9025	24.67217	4	277	1	303	2023-08-16 02:20:12.000 -0400	TER
9	352002069	9327011	757955	66.99729	24.53957	0	275	1	334	2023-08-16 02:17:35.000 -0400	TER
10	463057101	8749030	1069166	67.30532	24.77485	21	310	0	305	2023-08-16 02:20:38.000 -0400	TER
11	11440308000	9303651	314879	66.98128	24.83627	0	29	5	100	2023-08-16 02:21:25.000 -0400	TER
12	626062000	9213222	406915	66.9733	24.81637	0	352	5	175	2023-08-16 02:19:04.000 -0400	TER
13	636022964	9683130	408067	65.81834	23.55	100	511	0	285	2023-08-16 02:17:47.000 -0400	SAT
14	440279000	9292266	408720	66.81599	24.58992	89	30	0	37	2023-08-16 02:21:54.000 -0400	TER
15	355275000	9225031	410817	66.97762	24.83062	0	29	5	140	2023-08-16 02:20:37.000 -0400	TER
16	463070101	9831048	998680	67.23331	24.78082	78	511	0	242	2023-08-16 02:21:53.000 -0400	TER
17	538007362	9772046	4920214	66.97425	24.81291	0	342	5	104	2023-08-16 02:21:26.000 -0400	TER
18	462062101	0	5684159	66.96874	24.82888	0	511	0	0	2023-08-16 02:21:01.000 -0400	TER
19	376766000	9825166	6201638	66.97697	24.83421	3	322	8	67	2023-08-16 02:20:12.000 -0400	TER
20	311001204	9646895	2758	66.98158	24.55713	3	260	1	182	2023-08-16 02:20:02.000 -0400	TER
21	419001620	9320752	200961	62.56977	23.7665	154	511	0	108	2023-08-16 02:21:02.000 -0400	SAT
22	352001643	9349655	714053	64.99167	23.6	100	511	0	103	2023-08-16 02:17:17.000 -0400	SAT
23	538007637	9514078	5825380	67.2688	24.79082	0	102	5	152	2023-08-16 02:19:30.000 -0400	TER
24	271049616	9931393	6929869	67.33676	24.77073	0	38	12	238	2023-08-16 02:19:12.000 -0400	TER
25	636019833	9249116	212352	67.28835	24.78341	0	119	6	119	2023-08-16 02:17:40.000 -0400	TER
26	538010513	9592305	214768	61.99588	23.64904	16	303	0	18	2023-08-16 02:18:05.000 -0400	TER
27	636022427	9189926	1236143	66.98064	24.8354	0	211	5	154	2023-08-16 02:19:32.000 -0400	TER
28	234557043	0	4695828	66.9107	24.82462	11	82	0	84	2023-08-16 02:20:09.000 -0400	TER
29	255900000	9731781	4701006	66.83424	24.84661	115	511	15	23	2023-08-16 02:21:00.000 -0400	TER
30	463001199	0	5971601	66.97614	24.80752	46	511	99	157	2023-08-16 02:20:36.000 -0400	TER
31	376864000	9908085	6899475	66.9773	24.83668	33	188	13	215	2023-08-16 02:21:16.000 -0400	TER
32	271049389	9931367	6903910	67.33884	24.77213	0	228	8	32	2023-08-16 02:16:31.000 -0400	TER
33	477108300	9175535	58071531	67.29787	24.77675	48	120	0	118	2023-08-16 02:21:14.000 -0400	TER
34	373026000	9800049	5954368	66.9295	24.67726	1	268	1	356	2023-08-16 02:21:33.000 -0400	TER

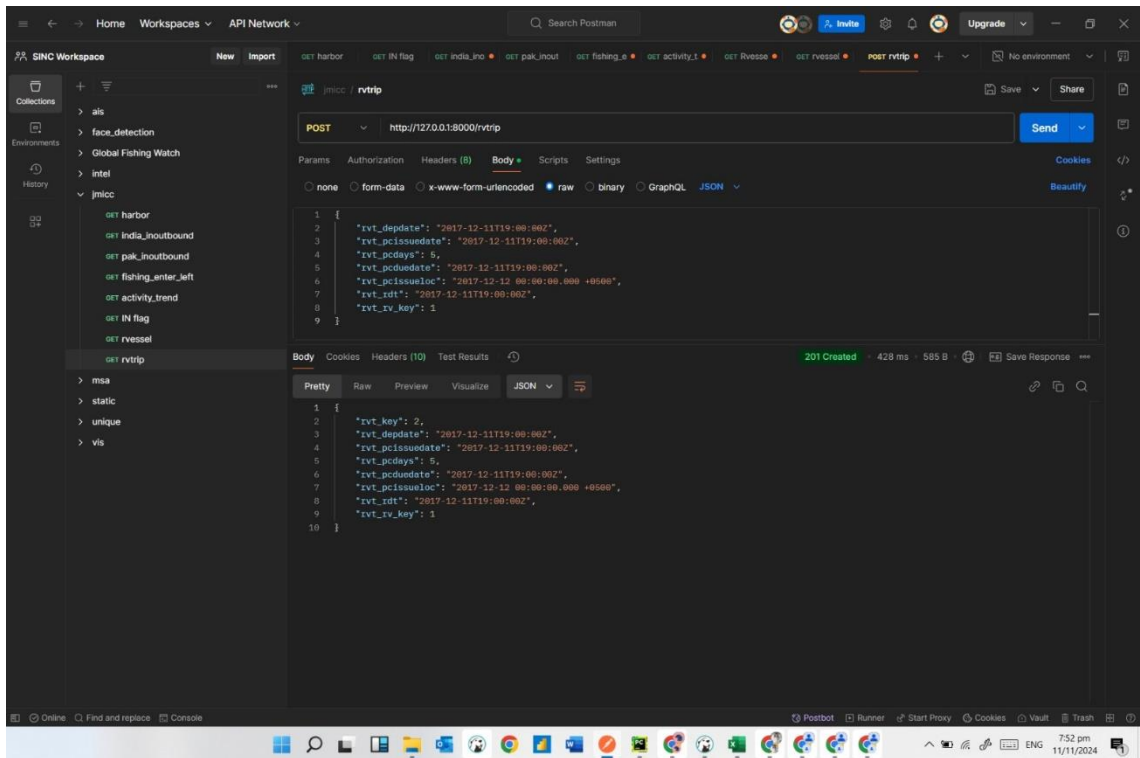


### 3. Data Ingestion & API Integration:

- Configured automated pipelines to fetch live data from external APIs using Postman-tested endpoints.
- Built scheduled scripts for continuous data synchronization.

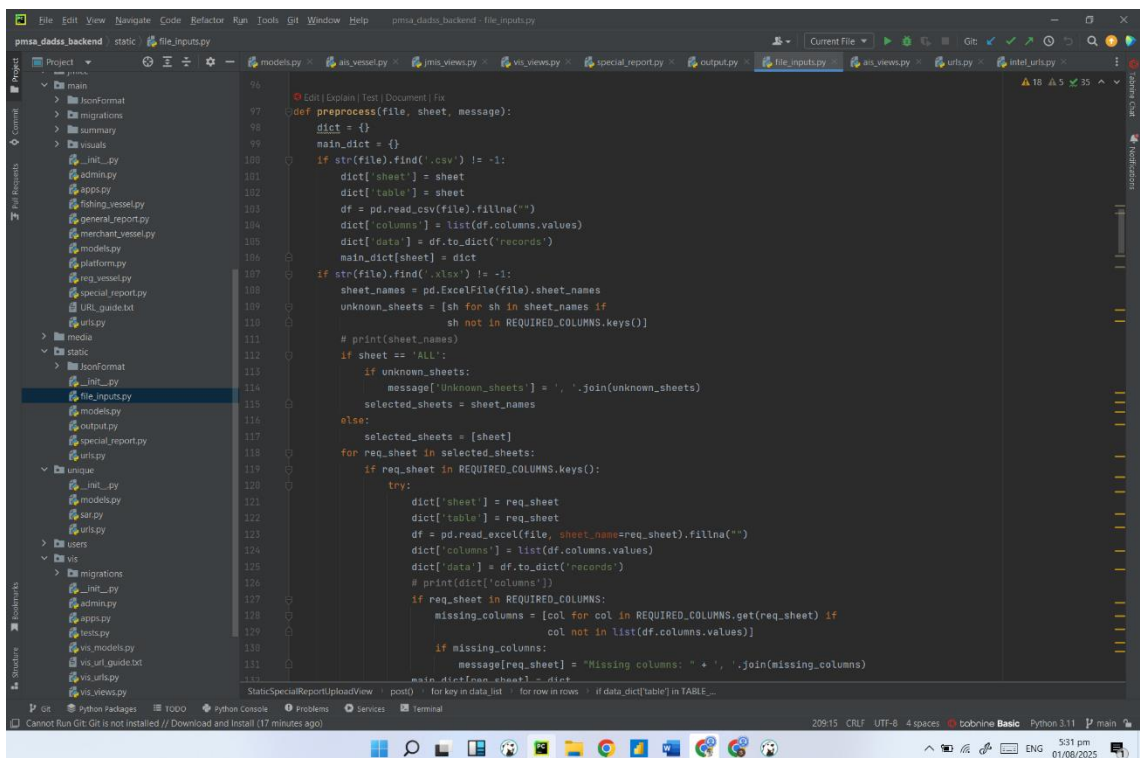






#### 4. Data Cleaning & Preprocessing:

- Implemented Python-based ETL processes to validate, clean, and enrich CSV and API data.
- Addressed missing values, duplicate entries, and inconsistent formats before database insertion.



```

dict['sheet'] = req_sheet
dict['table'] = req_sheet
df = pd.read_excel(file, sheet_name=req_sheet, fillna(""))
dict['columns'] = list(df.columns.values)
dict['data'] = df.to_dict('records')
# print(dict['columns'])

if req_sheet in REQUIRED_COLUMNS:
    missing_columns = [col for col in REQUIRED_COLUMNS.get(req_sheet) if
                        col not in list(df.columns.values)]

    if missing_columns:
        message[req_sheet] = "Missing columns: " + ', '.join(missing_columns)
    main_dict[req_sheet] = dict
    dict = {}
except ValueError:
    message[req_sheet] = 'Error in sheet.'

return main_dict

class StaticSpecialReportUploadView(APIView):
    # read csv file and convert it into json
    def post(self, request):
        file = request.FILES.get('file')
        sheet = request.POST.get('sheet')

        if not file:
            return JsonResponse({"error": "No file provided."}, status=400)
        if sheet != 'ALL' and sheet not in list(REQUIRED_COLUMNS.keys()):
            return JsonResponse({"error": "Unknown sheet."}, status=400)
        if str(file).find('.csv') == -1 and str(file).find('.xlsx') == -1:
            return JsonResponse({"error": "Unknown file format."}, status=400)

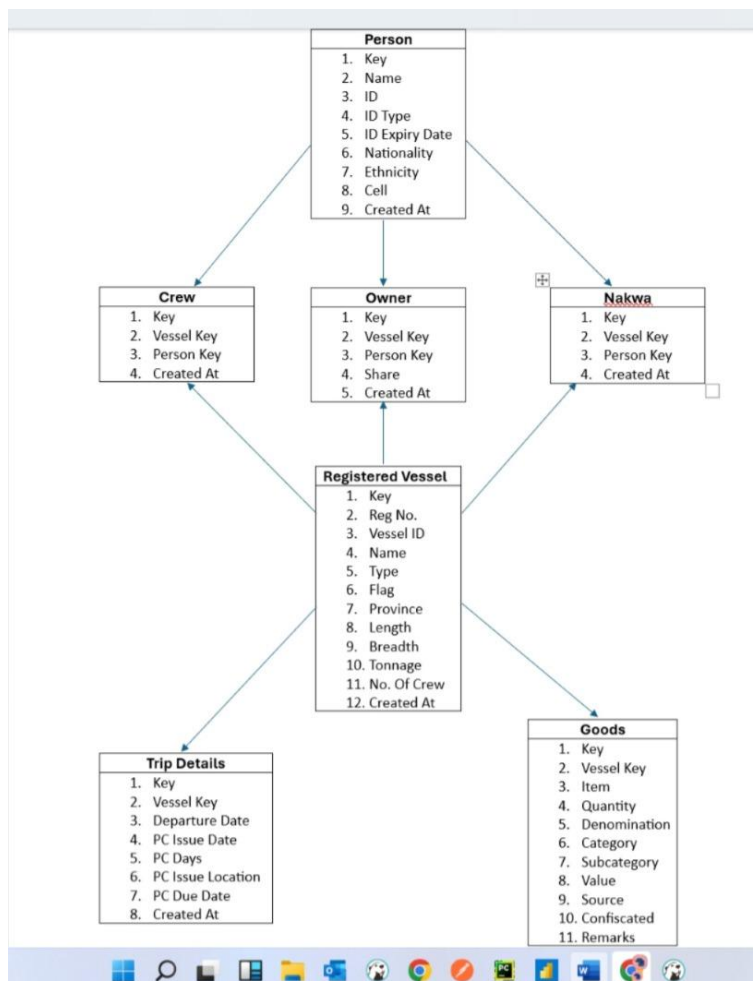
        message = {}
        data_list = preprocess(file, sheet, message)

        if len(data_list) == 1 and len(message) == 1:
            return JsonResponse({"sheet": message[sheet]}, status=400)

```

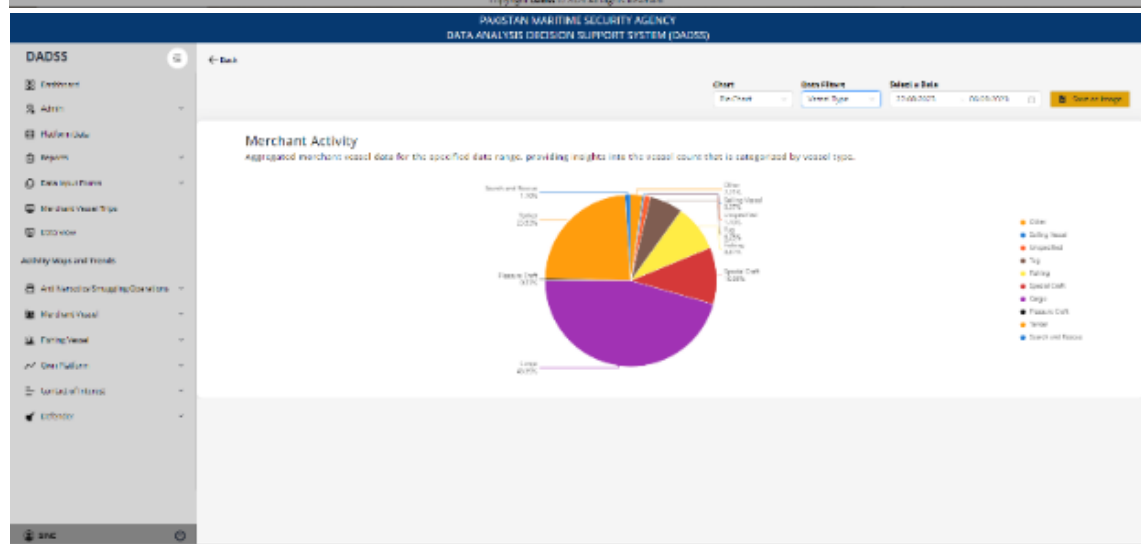
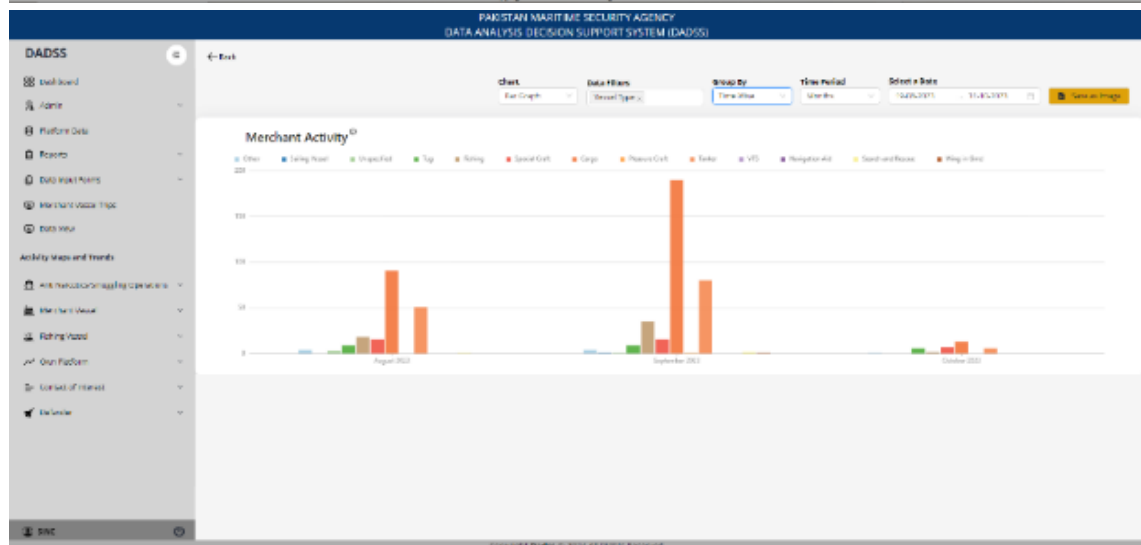
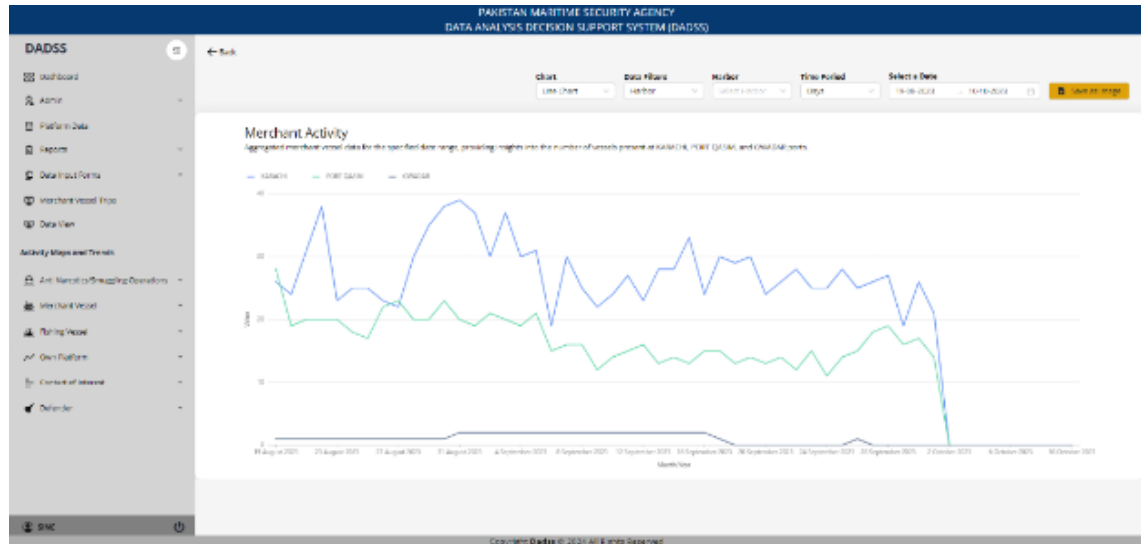
## 5. Data Modeling:

- Established relational models connecting vessel details, trips, and activity metrics.
- Designed foreign key relationships ensuring data integrity and high query performance.



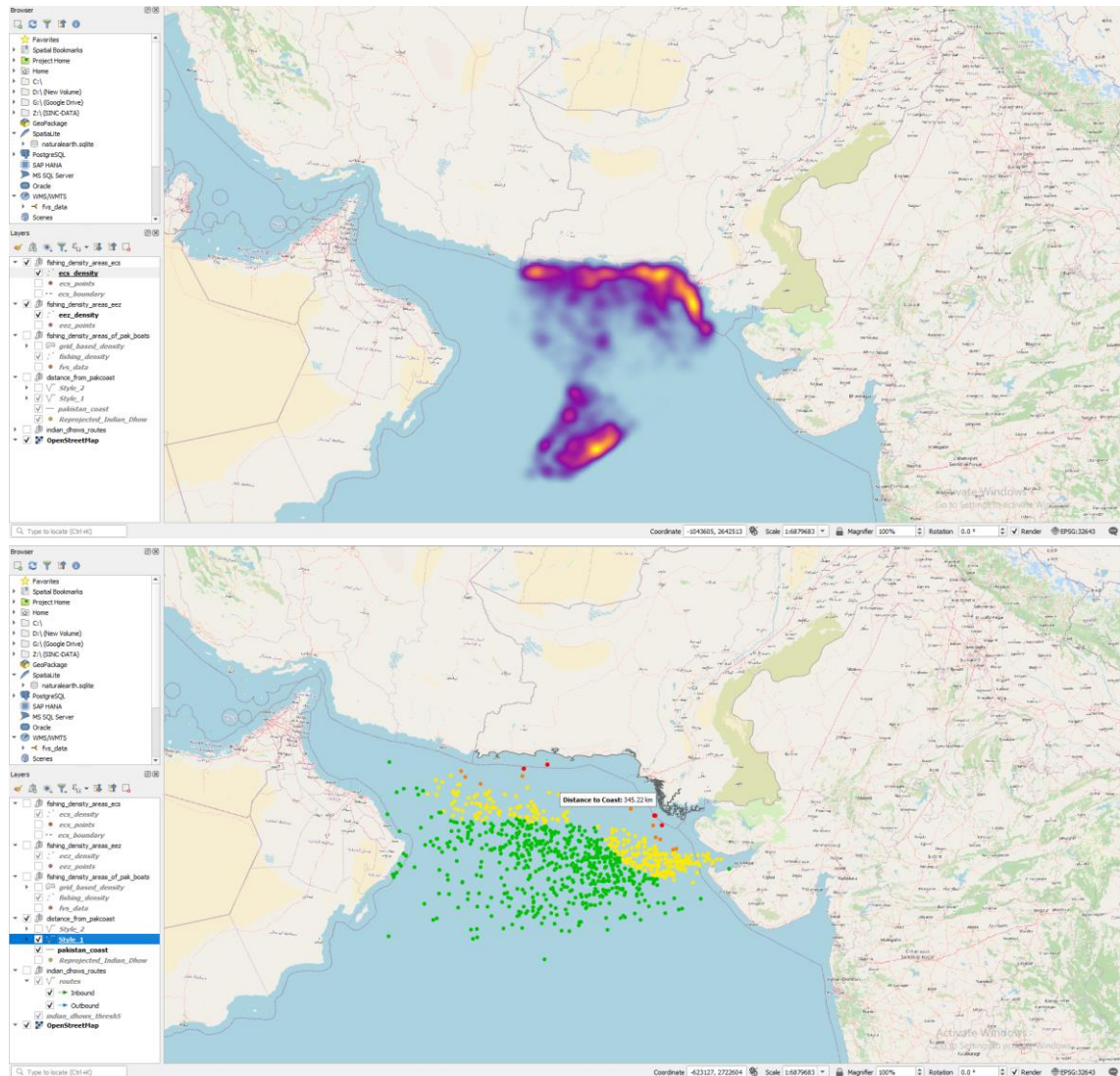
## 6. Data Visualization:

- Created custom dashboards and charts to represent key performance metrics and vessel activity.
- Utilized advanced visualization libraries for clarity and interactivity.







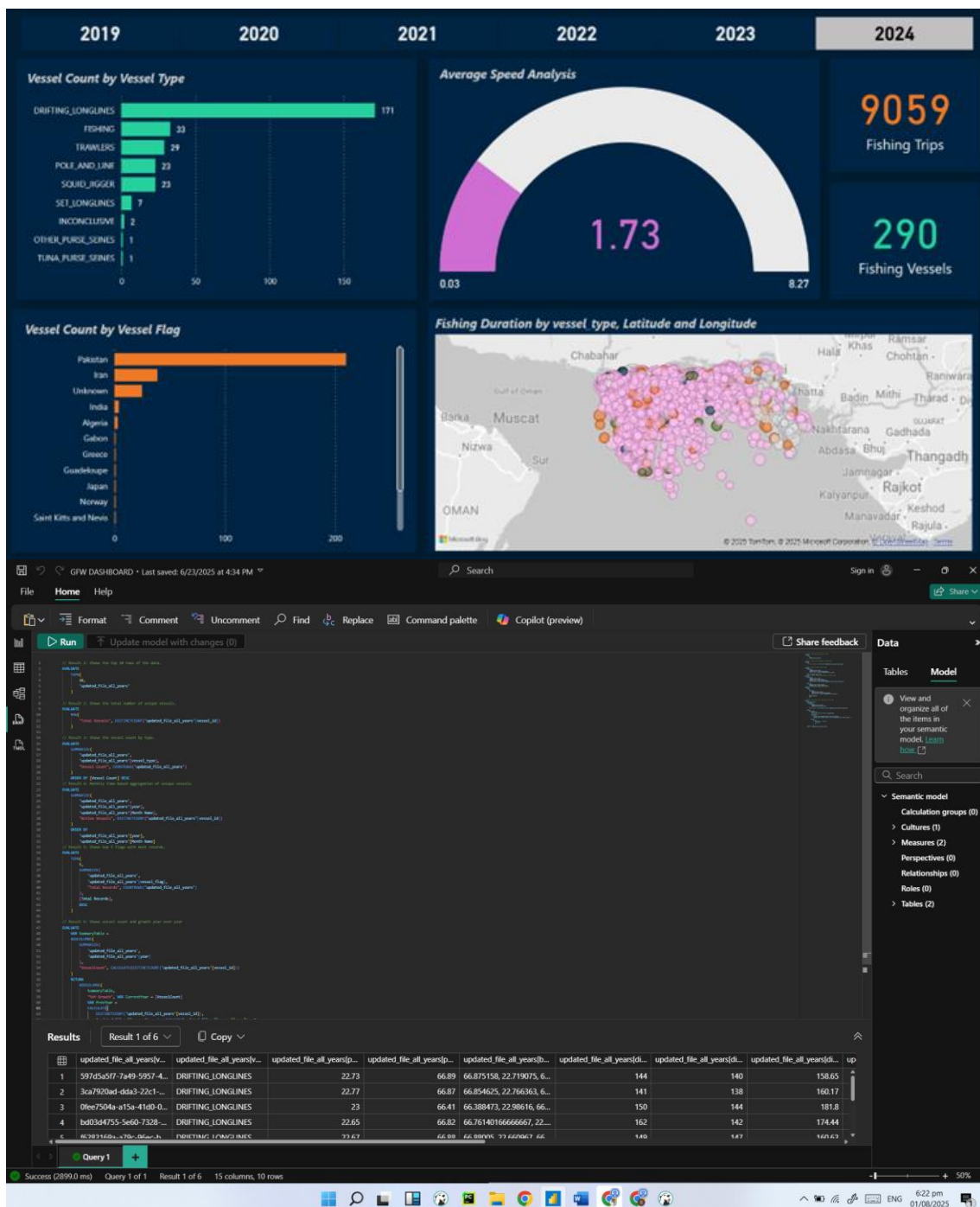


## 8. Reporting & Automation:

- Generated automated analytical reports and alerts, significantly reducing manual efforts.
- Enabled stakeholders to access real-time data visualizations and downloadable reports.

# Power BI Maritime Vessel Dashboard with DAX Exploration

Designed a comprehensive Power BI dashboard to analyze maritime vessel activity using real-world geospatial and temporal data. The dashboard visualizes key metrics including vessel types, flag distribution, fishing trip counts, and average vessel speed. An interactive map plots fishing zones using latitude and longitude. Alongside, a set of DAX queries were implemented in Query View for deeper data exploration—covering yearly trends, monthly activity, flag-wise summaries, and vessel-type breakdowns.





# CUSTOMER DATA ANALYSIS

## KPMG DATA ANALYTICS INTERNSHIP

Conducted data quality assessment on customer and transactional datasets, including wealth segments and financial metrics, identifying gaps in completeness, validity, consistency, and uniqueness, prepared remediation recommendations. Gained exposure to financial datasets like those used in wealth management, brokerage, and securities environments. Designed and delivered visual dashboards in Power BI for client-style presentations, covering segmentation, trend analysis, and KPI insights.

