

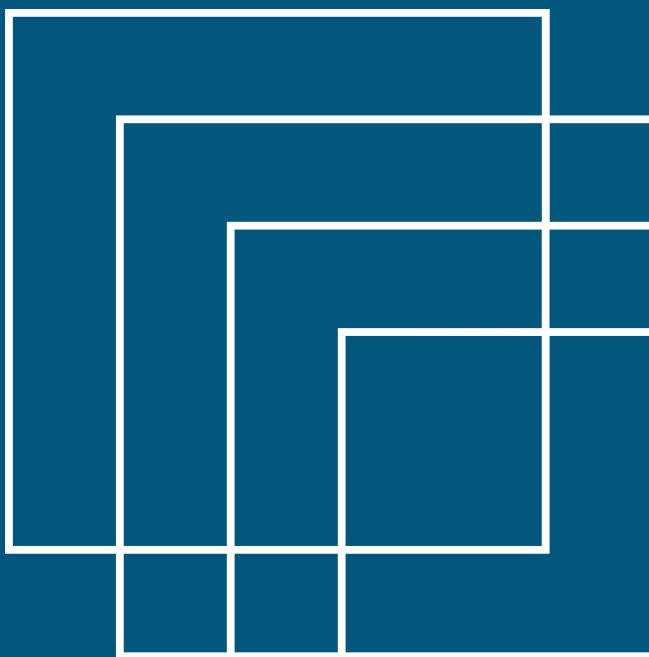
# PROJECT 3

# CREDIT CARD FRAUD

GROUP: ANDREAS HILL, LAYLA ABDUL SATER, OSMAR EDUARDO  
LUNA LUACTERO, PRISCILA GLIENKE

# CONTENT

- 01** CONCEPT
- 02** OUR DATA
- 03** BACK END (ETL)
- 04** WEB SCRAPING
- 05** NEW JS LIBRARY
- 06** VIZUALIZATIONS
- 07** DASHBOARD



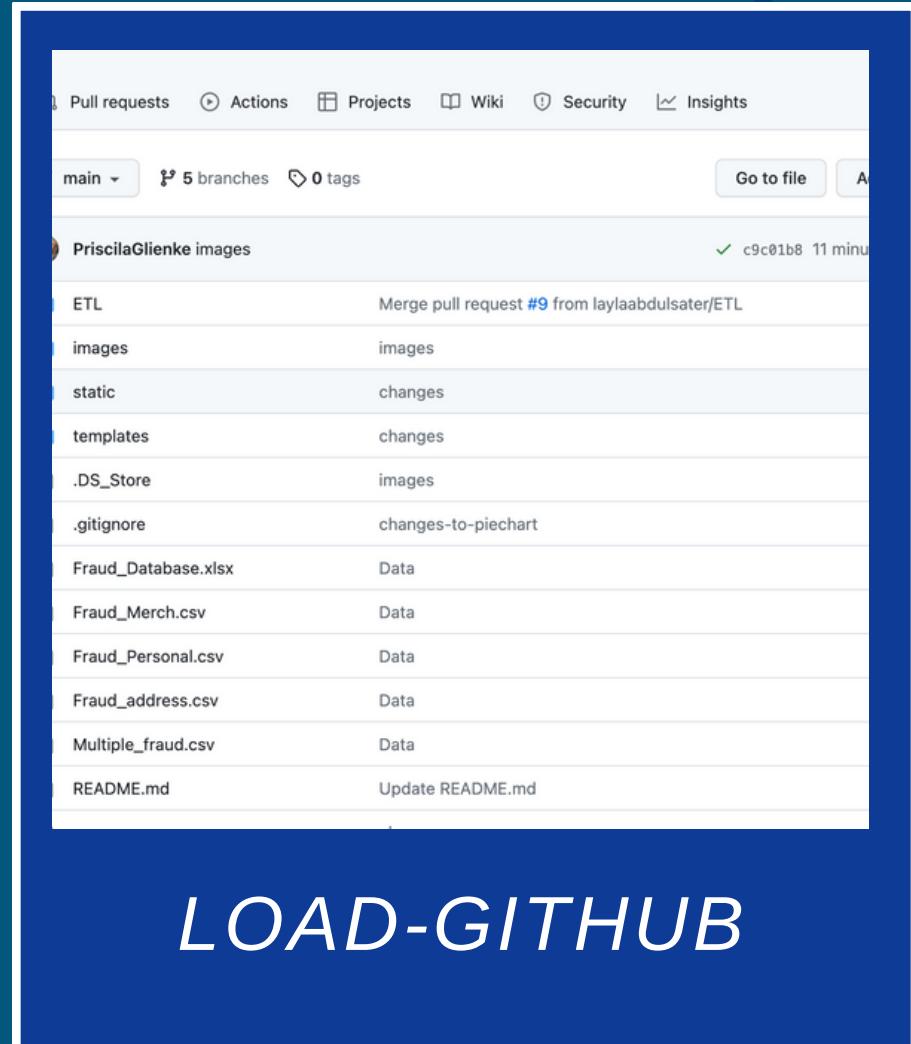
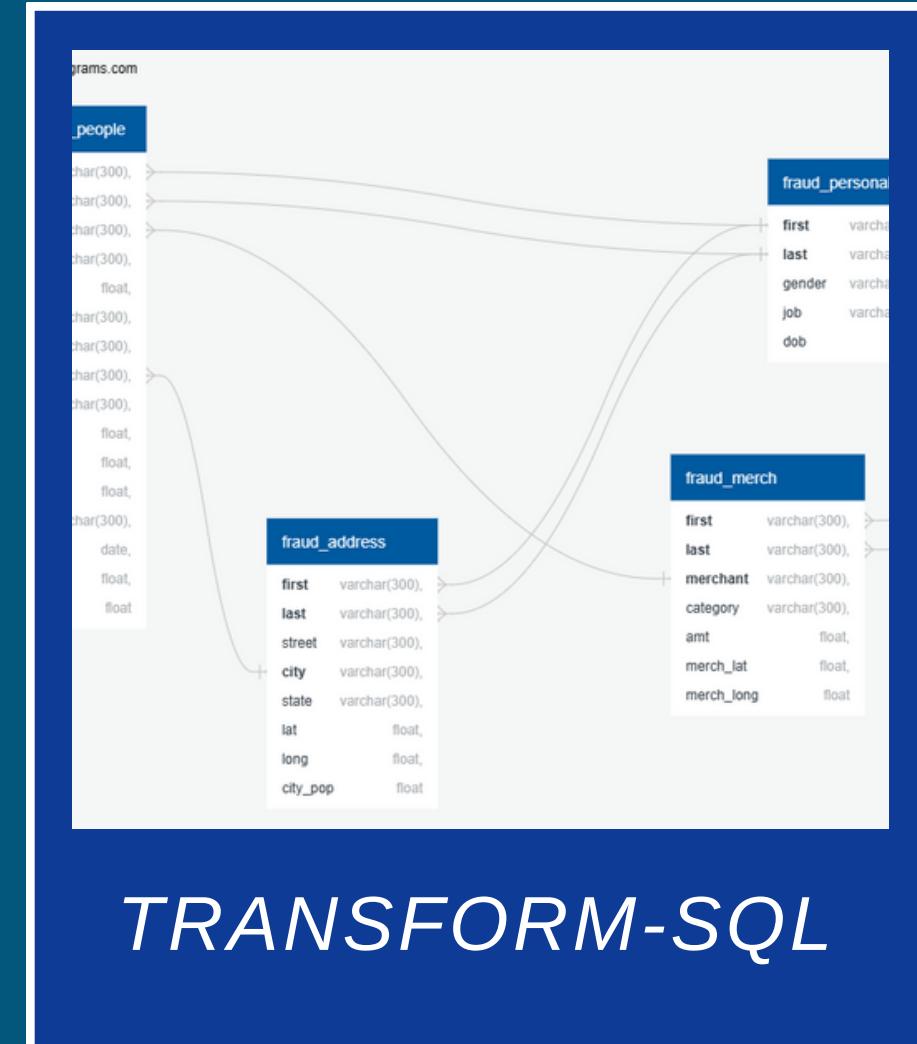
# CONCEPT

- Our assignment proposal focuses on creating a compelling and interactive learning experience that highlights cases of credit card fraud.
- In today's digital age, credit card fraud has become a significant concern affecting individuals and businesses worldwide. By completing this assignment and developing a web app focused on educating and increasing awareness about credit card fraud, we aim to address this pressing issue. Through interactive and engaging features, such as real-life case studies, and visualizations, our web app will serve as a valuable resource to educate users about common fraud cases.



# OUR DATA

## BACK END



# WEB SCRAPING

```
[7] List = zip(Title, Paragraphs)

[8]

> v tuple(List)

[9] .. ((<span class="subheading-medium">1. Evaluate websites before you buy</span>
       <span class="body-copy">Join and get over $1,200 in value!</span>
     (<span class="subheading-medium">2. Keep your credit card</span>
       <span class="body-copy">Start saving today, tax-free. Discover</span>
     )
```

A screenshot of a code editor showing a Python script for web scraping. The script uses the `urllib.request` module to fetch a JSON file, then parses it with `json.loads`. It prints the title and metadata. It then iterates through the features, printing each feature's properties. It also prints events where there are feit reports. The code includes comments explaining its purpose.

```
jsondata_start.py X
...
# Example file for parsing and displaying JSON data
# import modules
import urllib.request
import json

# printResults(data)
# Use the json module to load the string data into a dictionary
theJSON = json.loads(data)

# now we can access the contents of the JSON like this
if "title" in theJSON["metadata"]:
    print(theJSON["metadata"]["title"])

# output the number of events, this is the length of the array
count = theJSON["metadata"]["count"]
print(str(count) + " events recorded")

# for each event, print the place name and category
for i in theJSON["features"]:
    print(i["properties"]["place"])
    print("-----\n")

# print the events that only have a certain category
for i in theJSON["features"]:
    if i["properties"]["mag"] > 0.0:
        print("%2.1f" % i["properties"]["mag"])
        print("-----\n")

# print only the events where there were felt reports
print("Events that were felt")
for i in theJSON["features"]:
    feltReports = i["properties"]["felt"]
    if feltReports != None:
        if feltReports > 0:
            print("%2.1f" % i["properties"]["mag"] + " reported " + str(feltReports))
            print("-----\n")
```

## NEW JS LIBRARY

It's a customizable replacement for the standard JavaScript alert box. SweetAlert allows you to create attractive and responsive pop-up modals for displaying important messages or notifications.



# sweetalert2

```
<link rel="stylesheet" href="sweetalert2.min.css">
```

```
<script src="sweetalert2.min.js"></script>
```

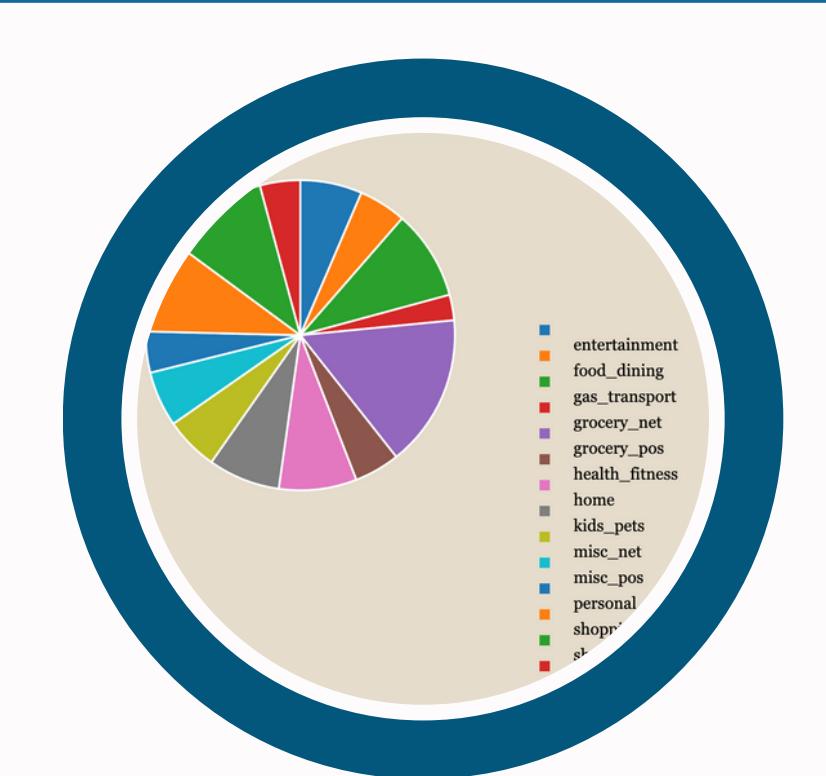


A message with auto close timer

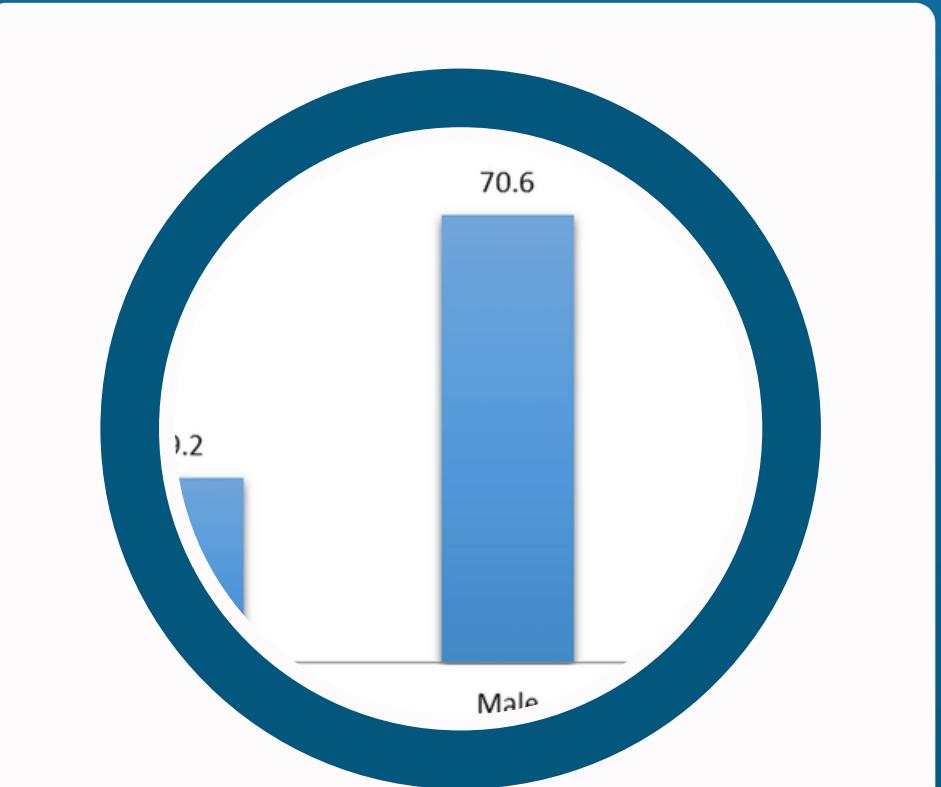
# VISUALIZATIONS



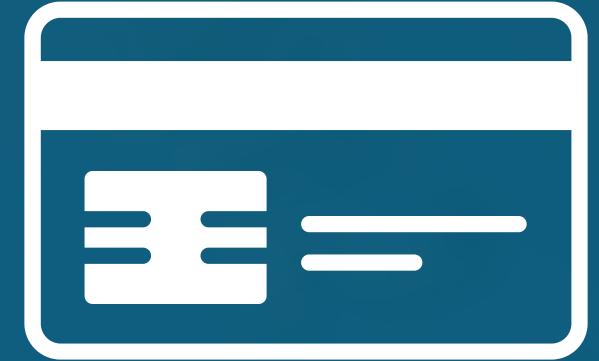
**Fraud Density**  
Map



**Distribution of Fraud Category**  
Pie Chart



**Gender**  
Bar Chart



# DASHBOARD

[HTTPS:127.0.0.1:5000](https://127.0.0.1:5000)