

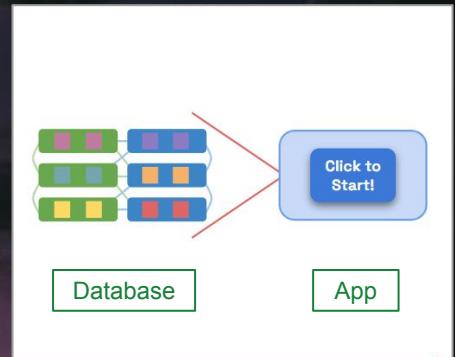
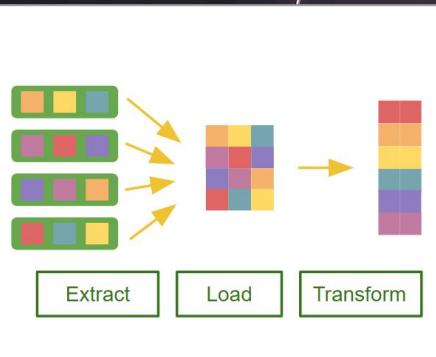
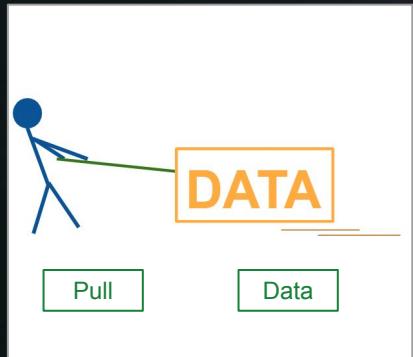


Natural Disaster Insights

Project 3 | Group 9

Ian O'Connor, Lily Saltonstall, Angelina Wright, Eylem Yildirim

Project Roadmap



Data Pull

Tools:
EM-DAT (.xlsx)

ETL Process

Tools:
Jupyter Notebook
Libraries:

- Pandas
- Hashlib

Database Setup

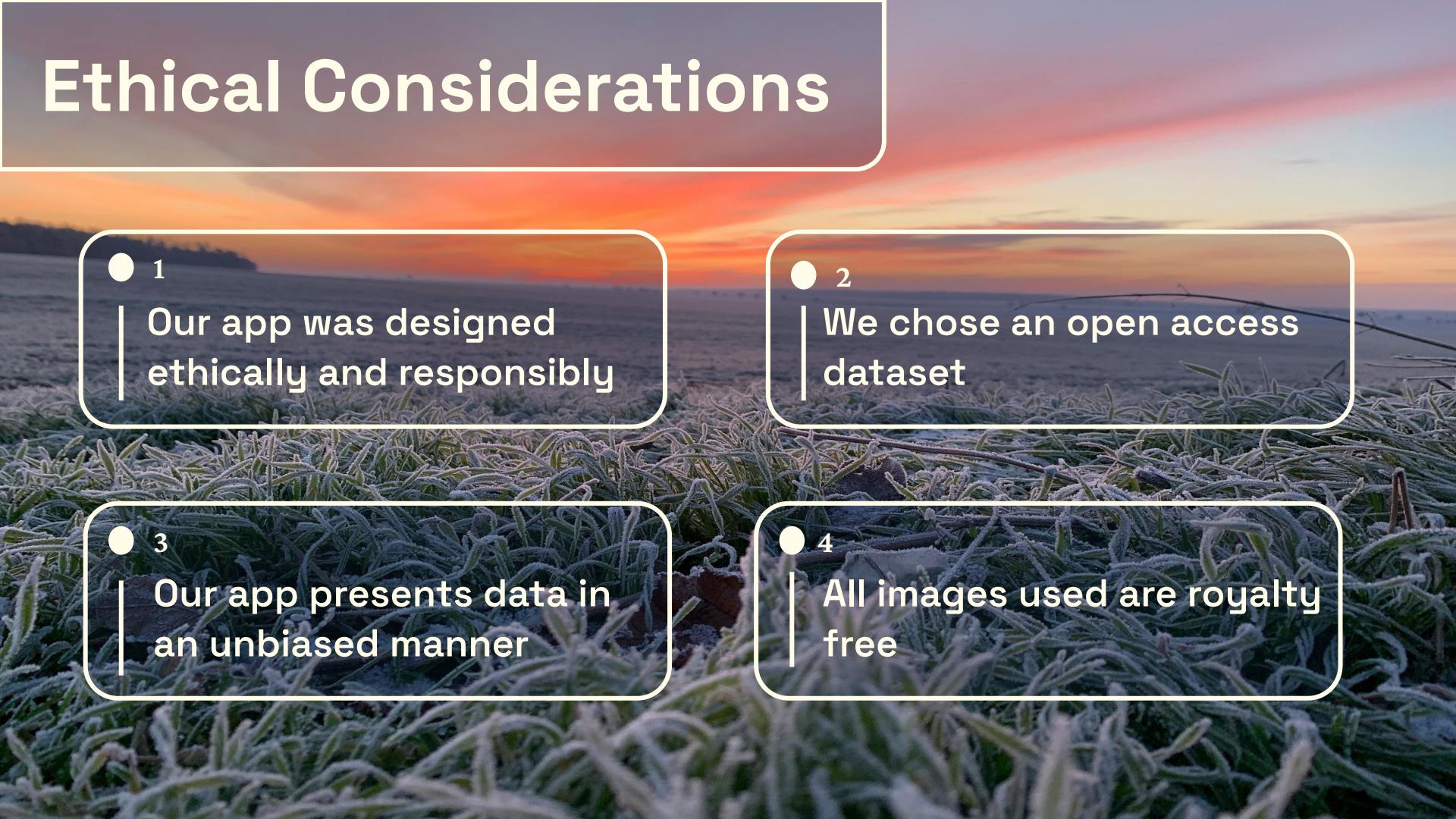
Tools:
PostgreSQL
QuickDBD

App Creation

Tools: Python
Libraries:

- PIL (Python Imaging Library)
- Tkinter
- SQLAlchemy

Ethical Considerations



1

Our app was designed
ethically and responsibly

2

We chose an open access
dataset

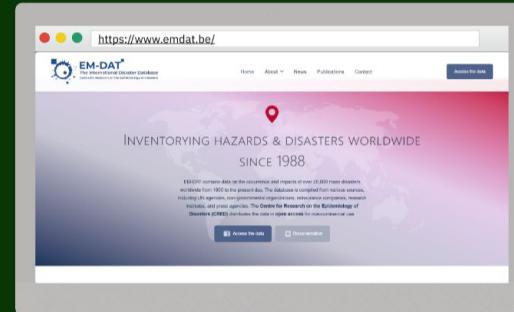
3

Our app presents data in
an unbiased manner

4

All images used are royalty
free

The Data



01

Dataset Overview

Source

EM-DAT (Emergency Events Database)

- Created by Centre for Research on Epidemiology of Disasters (CRED) and World Health Organization (WHO)

Purpose

Provides data (26,000 disasters worldwide) to enhance decision-making in disaster preparedness, risk assessment, and the prioritization of resource allocation.

Natural Disasters Dataset

North America
2000-2024

Original Data

Rows:

- 697

Columns:

- 46

ETL Process

Extract

Pandas

- Extracted raw data and imported into a Pandas DataFrame for cleaning and preprocessing

Transform

Dropped Columns

- Null Values

Hashlib Library

- ImpactID, LocationID

Combined year, month, day columns

Summary Statistics

- Summarize deaths and injuries by disaster type
- Min, max, mean, variance by Total_Affected

Added Calculation of Duration

Challenges

- Incomplete records
- Formatting
- Irrelevant Data
- Null Values

Load

5 DataFrames → .csv

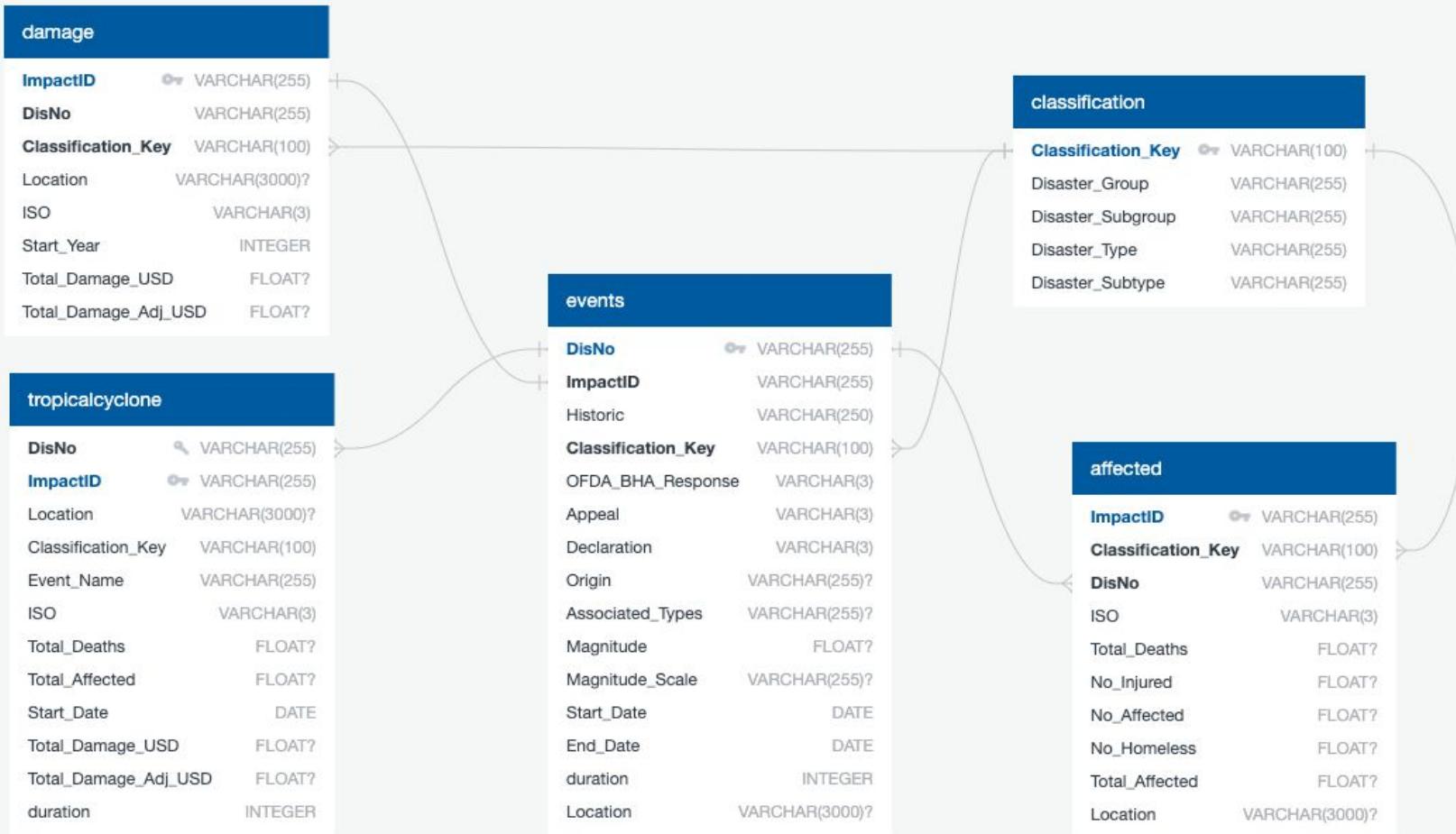
- Classification - 26 rows
- Events - 654 rows
- Damage - 654 rows
- Affected - 305 rows
- Tropical Cyclone- 73 rows

Database: PostgreSQL

DB Name: NaturalDisasters

- Suitable for smaller datasets
- Ideal for this project which involved multiple related tables with PK/FK
- Data integrity and validation via strict schema rules such as data types and constraints
- Chosen over MongoDB for its structured schema capabilities. MongoDB is better for large, unstructured/semi-structured data

Database Setup



Troubleshooting & Adjustment

Naming Conventions

- Used lowercase table names to avoid quotation requirements
- Updated column names not to cause issues when creating ERD

Testing & Debugging

- Adjusted order of import and deletion during tests
- Handled NULL and incomplete values in table columns
- Removed special characters from column values (comma, single quotations) for proper table imports



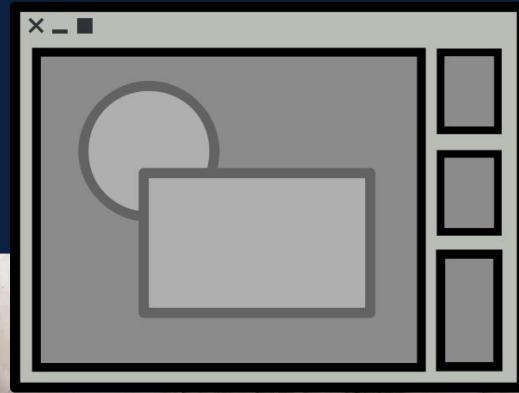
Events → events

DisNo. → DisNo

02



The App



App Overview

Command-Line

Click
Buttons

Tkinter

Select
Options

- ✓ User-driven experience
- ✓ Simple to navigate
- ✓ Accessible for everyone!

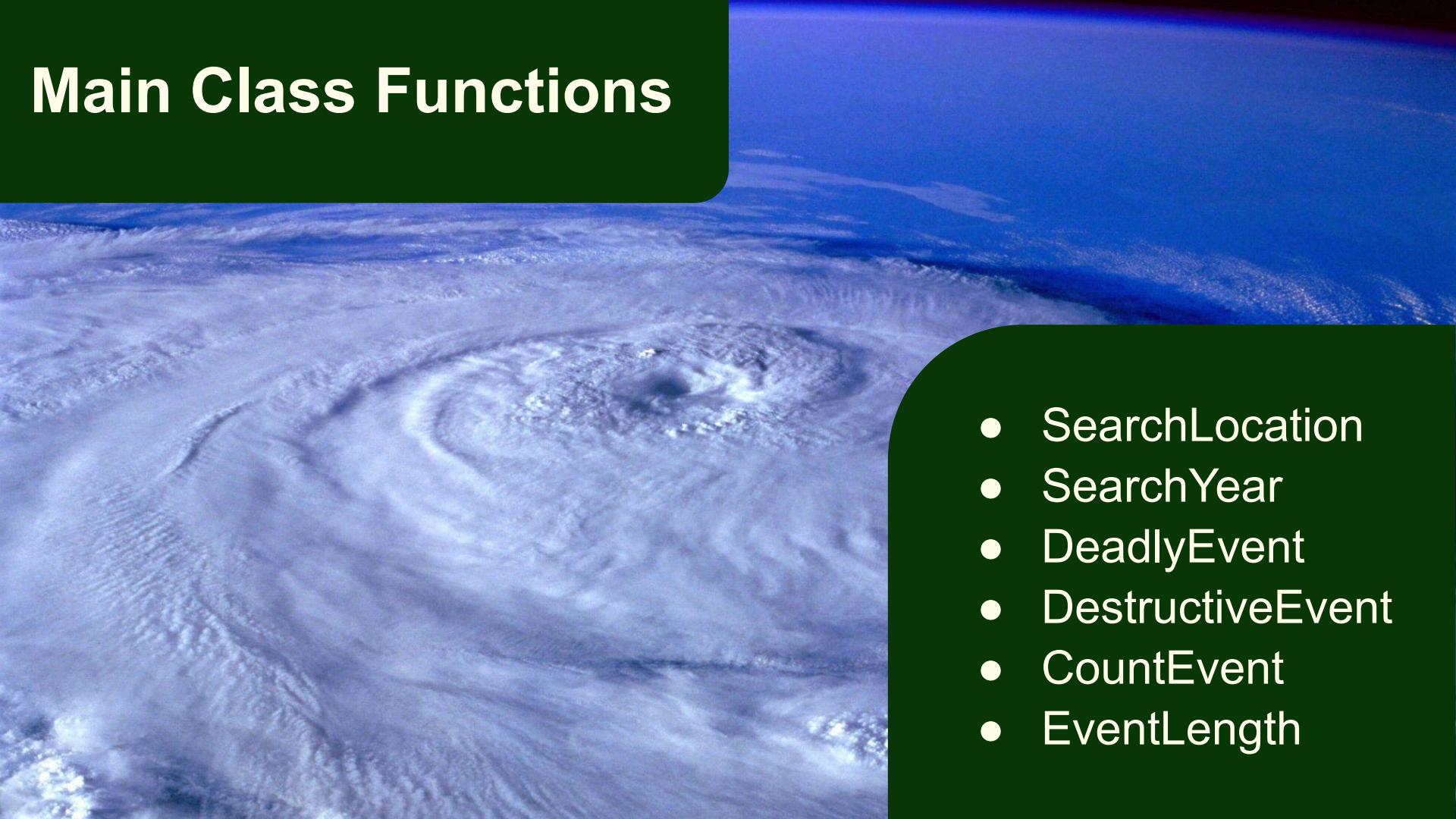
App Structure

NaturalDisasterApp

```
└── main.py  
└── Splash.py  
└── Menu.py  
└── AppController.py  
└── SearchLocation.py  
└── SearchYear.py  
└── DeadlyEvent.py  
└── DestructiveEvent.py  
└── CountEvent.py  
└── EventLength.py  
└── EventImage.py  
└── Database.py  
└── InputValidation.py
```



Main Class Functions



- SearchLocation
- SearchYear
- DeadlyEvent
- DestructiveEvent
- CountEvent
- EventLength

App Demo





Questions?

Thank You!

