



# DATE-A-PARK

Meet the Park of your  
Dreams

TEAM YOTTABYTES

A.Saracina – F.Coba – A.Erfani – A.Abraham

2018 Data Analytics Bootcamp

The George Washington University



# Go Outside **NOW**, Be in **NATURE**

---

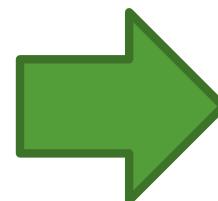
- 55% of all people live in urban areas, growing to 68% by 2050
- Endlessly bound to our tech and our jobs, we all yearn to look outwards
- We need space – We need to feel free
- **WE NEED TO MEET THE PARK OF OUR DREAMS**

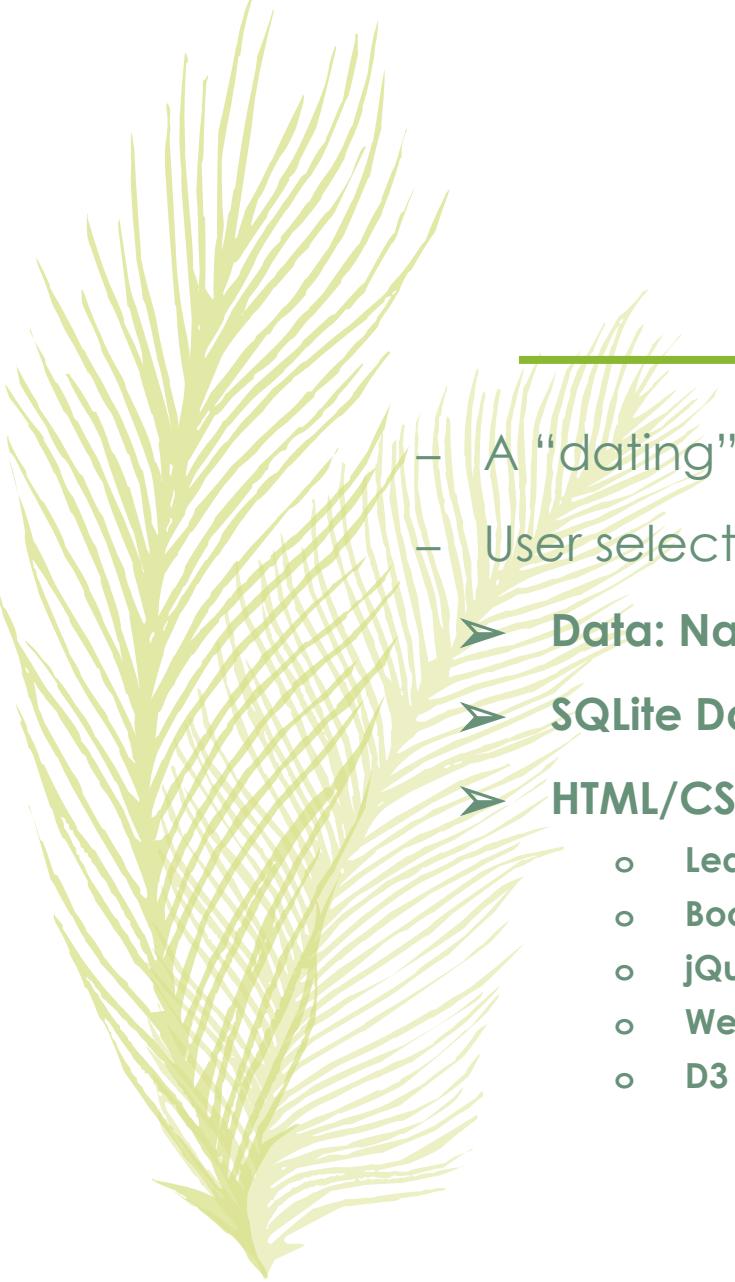


# Date-a-Park is the Answer

---

- 100% of all dating today involves some form of data analytics.
- Date-a-Park provides an analytical dashboard to help users find an ideal park experience for existential and physical wellbeing.





# Date-a-Park Project Scope

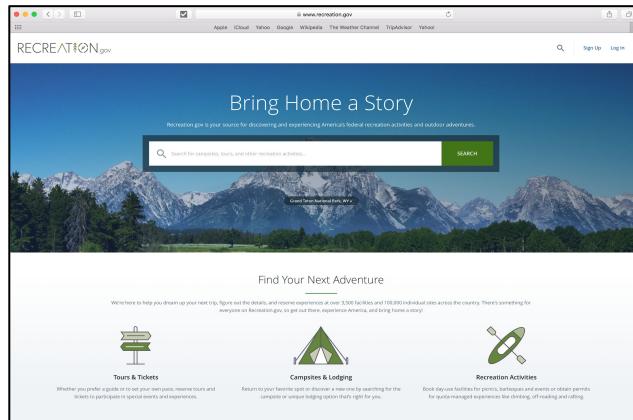
---

- A “dating” app for users to explore US National Parks and Recreation Areas
- User selects park location and features to create their best match.
  - **Data: National Parks Service and Recreation.gov source data**
  - **SQLite Database and Flask App**
  - **HTML/CSS/JavaScript**
    - Leaflet
    - Bootstrap
    - jQuery
    - WebGL/Cesium (NASA)
    - D3

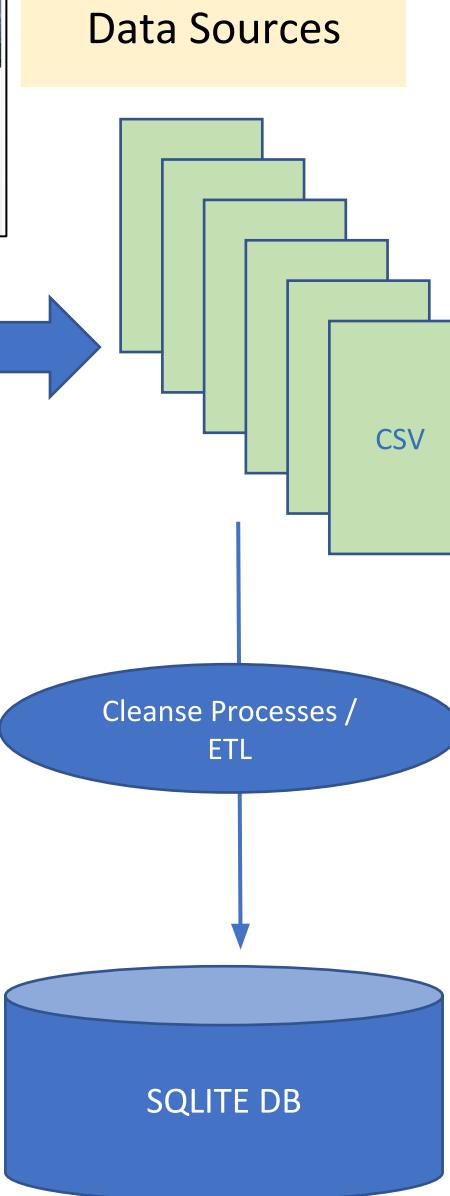
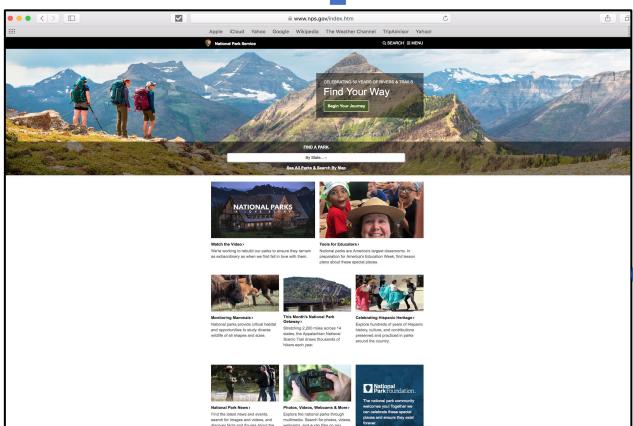


# Data Collection, Collation, and Cleaning

- API Query to National Park Service for GeoJSON for national parks and recreation areas.
- National Park Service and federal Recreation.gov websites provided .csv files containing detailed park attributes.
- 223,900 Location/Activity Pair



## Data Sources



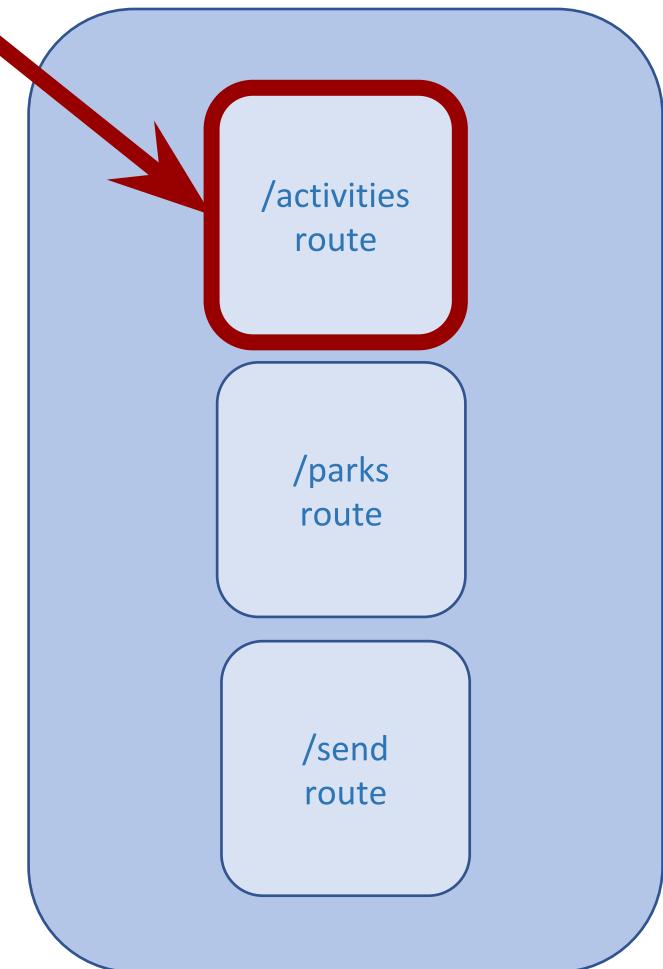
Transactional  
Database

# Making and using the databases!

```
@app.route("/activities")
def activities():
    """ Get all the activities """
    db_path = os.path.join('.', 'static','data',
    'date_a_park_SQLITEDB')
    con = sql.connect(db_path)
    cursor = con.cursor()
    cursor.execute("SELECT DISTINCT ACTIVITYNAME FROM activities
    ORDER BY ACTIVITYNAME ASC;")
    data = jsonify(cursor.fetchall())
    con.close()

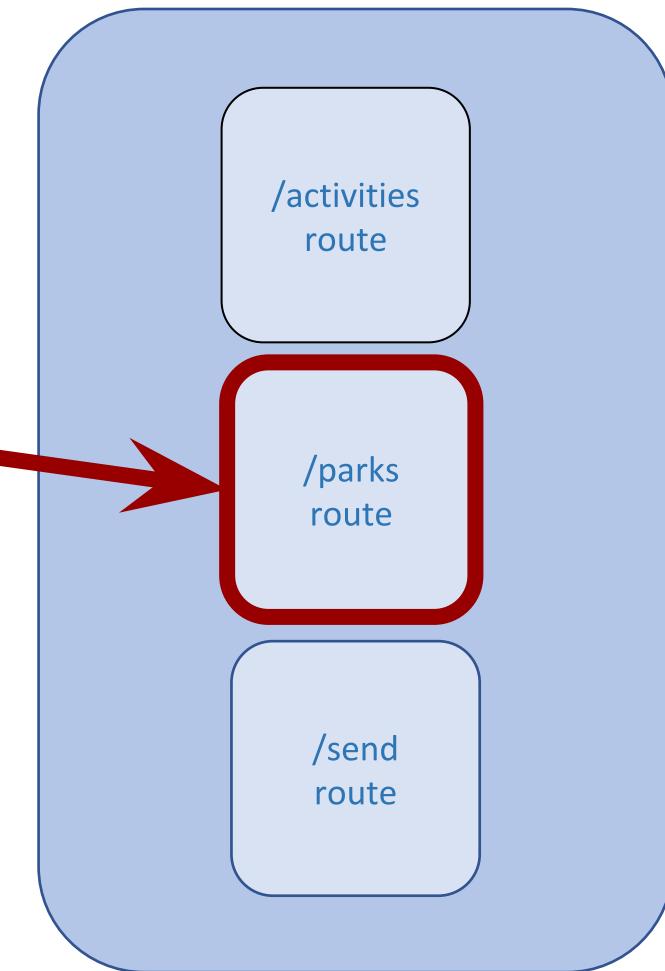
    return data
```

Python Flask App  
Middleware



## Python Flask App Middleware

```
@app.route("/parks")
def parks():
    return jsonify(json.load(open(os.path.join('.', 'static','data',
    'parks.geojson'))))
```

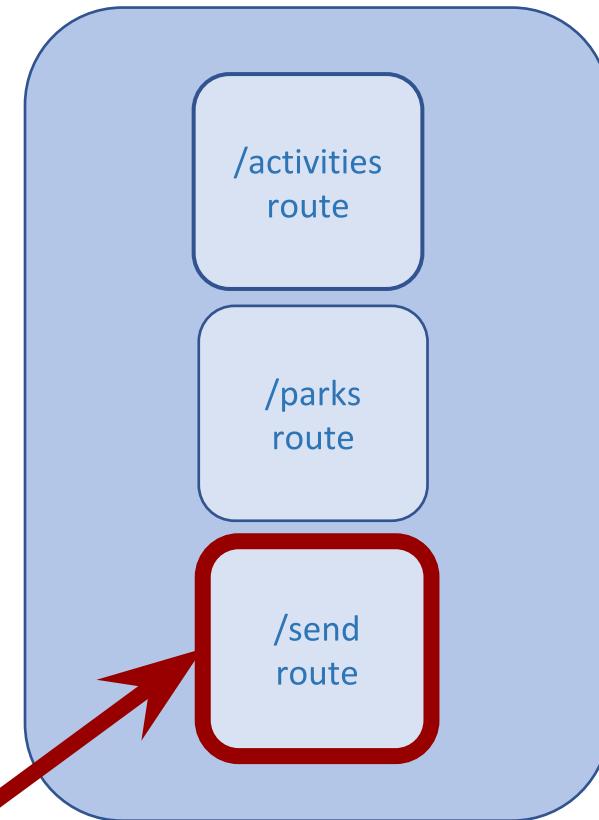


Python Flask App  
Middleware

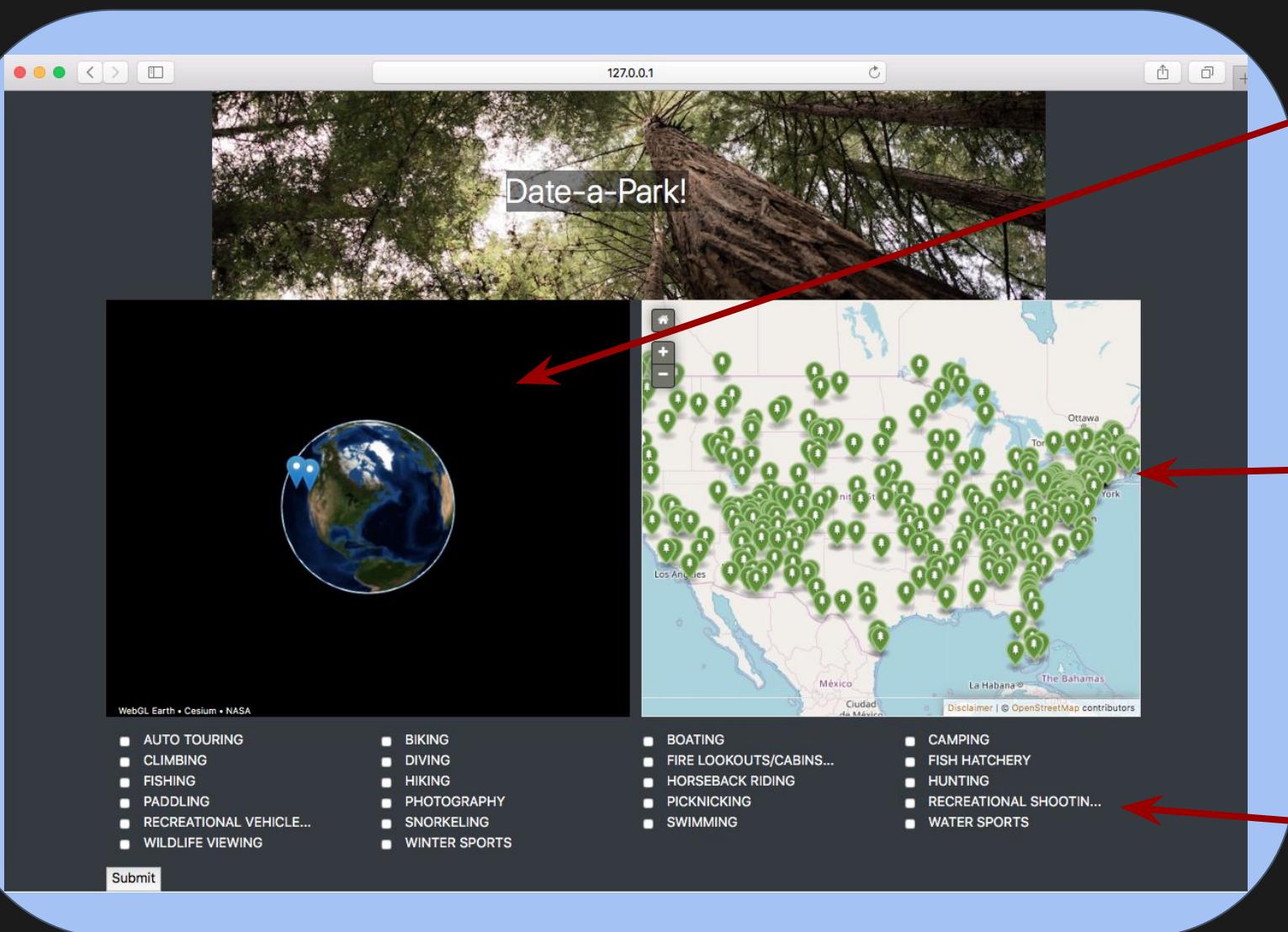
```
form_data = request.args
# get the activity keys
activities = []
query = "SELECT DISTINCT parks.NAME, parks.LAT,
parks.LONG FROM parks "
query += "INNER JOIN activities ON parks.ID =
activities.ID "

query += "WHERE activities.ACTIVITYNAME in (" 
for activity in form_data:
    query += "'" + activity + "' , "

# remove that last " AND " (5 characters)
query = query[:-3]
query += ");"
```



# JavaScript/Front-End

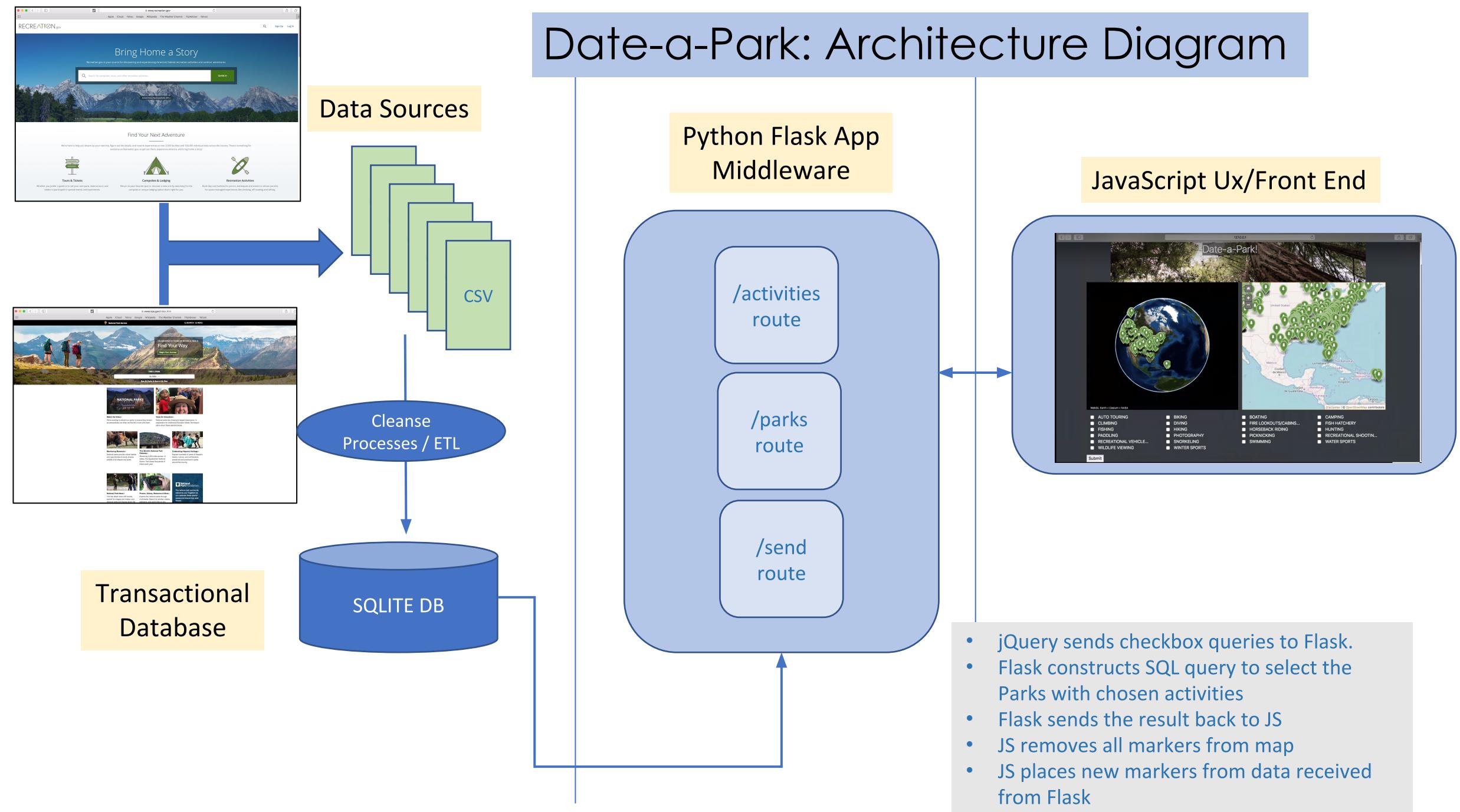


```
WE.tileLayer('http://tile...  
minZoom: 0,  
maxZoom: 5,  
attribution: 'NASA'  
}).addTo(earth);
```

```
var NPMMap = {  
  div: 'map',  
  baseLayers: [  
    'openstreetmap'  
  ],  
  overlays: [...  
};
```

```
d3.json('/activities').then(activities => {  
  activities.forEach(option => {  
    let div = d3.select('#activityOptions')  
      .append('div')  
      .attr('class', 'col-3');
```

# Date-a-Park: Architecture Diagram





# Date-a-Park Demonstration

---

Please kindly hold all applause  
until the presentation ends.



# Challenges & Future Work

- Debugging queries:
  - Map markers
  - User selections
  - Hover & popups
- Heroku deployment
- Travel distance
- Gaming features:
  - Earn wellness points
  - Display calories burned
- Rankings & reviews:
  - Trail difficulty
  - Peak times to visit
- Number of visitors
- Economic indicators
- Environmental impact

A photograph of a forest scene. In the foreground, a person wearing a red shirt and dark pants stands on a fallen log, looking out over a dense forest of tall, thin trees. The trees are closely packed, creating a vertical pattern. The lighting suggests it might be early morning or late afternoon, with sunlight filtering through the canopy.

GET OUT THERE! Time is short!

Write a new, better chapter in your life  
today!

**FIND YOUR DREAM PARK**

---

THANK YOU!