Civic Story Database Project – Spring 2020

- Matthew Izzo | 908-941-2351
- Sam Hajnasrollahi | 201-313-6455
- Babette Chao | 732-804-9003
- Joseph Candiano | 732-503-1449

Team 3 – CivicStory Database Improvement Project

GitHub: https://github.com/samihn/cab-sustainable-food-options

GitHub Link: https://github.com/samihn/cab-sustainable-food-options

Inception: "Executive Summary"

- Need: the important stakeholder and market need your group identified
  - Eating foods involving animal products and other harmful elements has more of an impact on the environment than say vegan options. We want to solve the issue of some people not being aware of these options or being able to find that.
  - The issue we want to address is part of the larger issue of climate change and sustainability.
- Approach: your unique and defensible approach
  - Our module will allow for people who live in New Jersey, or are thinking about moving here, to see how we are doing in terms of sustainability and if there is a certain town or county that is leading the pack.
  - Created a web page where users can search either a municipality, or a whole county within New Jersey to query our database and give the user the sustainability statistics for that certain municipality or the county. The query to our database should also return the local restaurants with plant-based food options within a certain radius of the initial search.
- Benefits: the value of your product when compared to the status quo or alternatives
  - Estimated budget for database development:
    - $1,000-2,500 for 6 months development (approvedindex AI)
  - Estimated budget for storage cost:
    - $0.02 per GB, extra $0.05 for every 100GB (Google BigQuery)
    - $0.0184 per GB, every 10,000 write operations will cost $0.05 (Microsoft Azure)
  - Our product costs none to develop, as it was an educational purpose
- Cost: the stakeholder cost to implement, e.g. does your approach replace an existing website, extend an existing website, or would it be a completely new website?
  - PHP implementation does not cost anything to implement

Elaboration: Project Proposal and Specifications

Problem Statement

The issue we want to address is part of the larger issue of climate change and sustainability. Eating foods involving animal products and other harmful elements has more of an impact on the environment than say vegan options. We want to solve the issue of some people not being aware of these options or being able to find that.

Objective

As the consequences and long-term effects of global CO2 emissions have become prominent, many companies and countries are trying to do their part by working toward climate control and carbon neutrality. Another big factor in helping climate change is making the change from meat-based food options to plant-based and vegan food options. Not only does this diet have a benefit for your personal health but the impact on the environment is just as great. Our module will allow for people who live in New Jersey, or are thinking about moving here, to see how we are doing in terms of sustainability and if there is a certain town or county that is leading the pack.

Desired End Product

Create a web page where users can search either a municipality, or a whole county within New Jersey to query our database and give the user the sustainability statistics for that certain municipality or the county. The query to our database should also return the local restaurants with plant-based food options within a certain radius of the initial search.

Importance and Need

In regards to working on the issue of certain foods causing more harm to the environment than others, our module should be important to a good amount of people. It is of course supposed to be more localized to New Jersey so it won't solve the entire issue of not eating a plant based diet, by it will help and there is a need for this type of module.

Research

Our group has and will be using some of civicstory.org's, along with srhub.org's, articles to gain knowledge on the sustainability issue. We would use the dataset of all of New Jersey's municipalities' sustainability numbers to return to users. We will also research some of the top restaurants in New Jersey with either an emphasis on their plant-based options and/or their plant-based options at the top of the menu.

Similar Systems

There are no similar applications that we could find pertaining to web pages that search a database of food options. However, there are applications like Yelp and Grubhub which let you search food options near you which could include vegan options. Our system is not being designed to work exactly like that though. Also there is a widespread effort on informing the general public and making the change to better the environment through dietary changes.

Other Possible Applications

This web-page could be modified in the future to allow users to input their diet and learn more about the importance of a plant-based diet. Another modification could be to allow users and municipalities to interact with each other to offer ideas and feedback about improving a certain area. Could also be modified and scaled to support a wider range than just New Jersey, this would only be possible if the application becomes a larger effort.

Performance

Our current idea for this website is not going to truly focus on performance. We will of course make sure we follow proper coding standards to ensure our database and web page communicate as smoothly as possible. It is important that the search does not take too long as that will be annoying to any user, so we will try to avoid the basic mistakes of coding, and avoid nested if/for loops.

Security

Since users will enter information about their food choices, the database should be secured to protect the user identities and food choice. The data gathered is used to provide more sustainable options near them, so assuming the users will also enter information about their local address, the database would have to protect the users' locations. One way we can try to secure the information given by the user is by encoding the data before using the information to find more sustainable options near the user. We can also use access control, by only allowing the admin to access the information, and establish identity upfront, so make users log in, in order to enter / view information in the database.

Backup and Recovery

Backup and recovery will be handled using GitHub. We will make sure to actively push and pull updates to and from using Git commands to avoid any loss of data. Other data such as research could also be stored on the cloud using Google drive along with local copies on our separate machines.

Technologies and Database Knowledge Needed

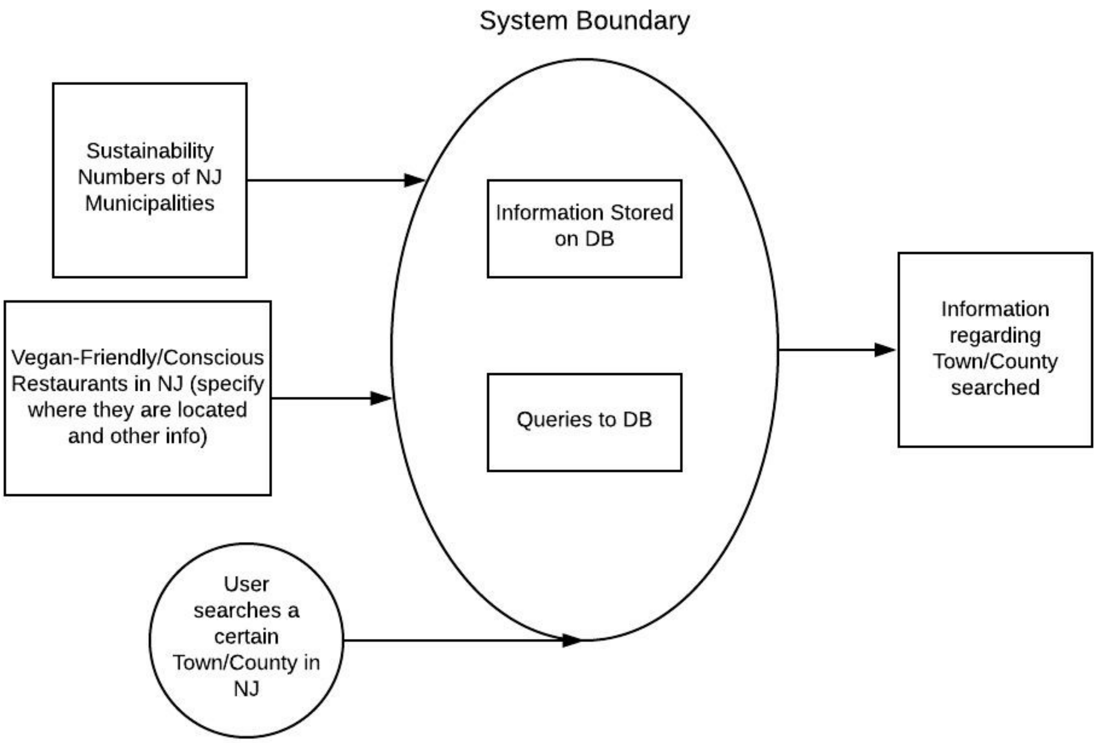
The team would need to learn PostgreSQL, while learning Python and PHP. Since most of us already have knowledge on Python and PHP, we would want to focus on learning PostgreSQL. To learn what was mentioned, we would watch some tutorials on Youtube online, while also finding credible and legitimate websites online that have information on learning PostgreSQL.

Some websites we could use:

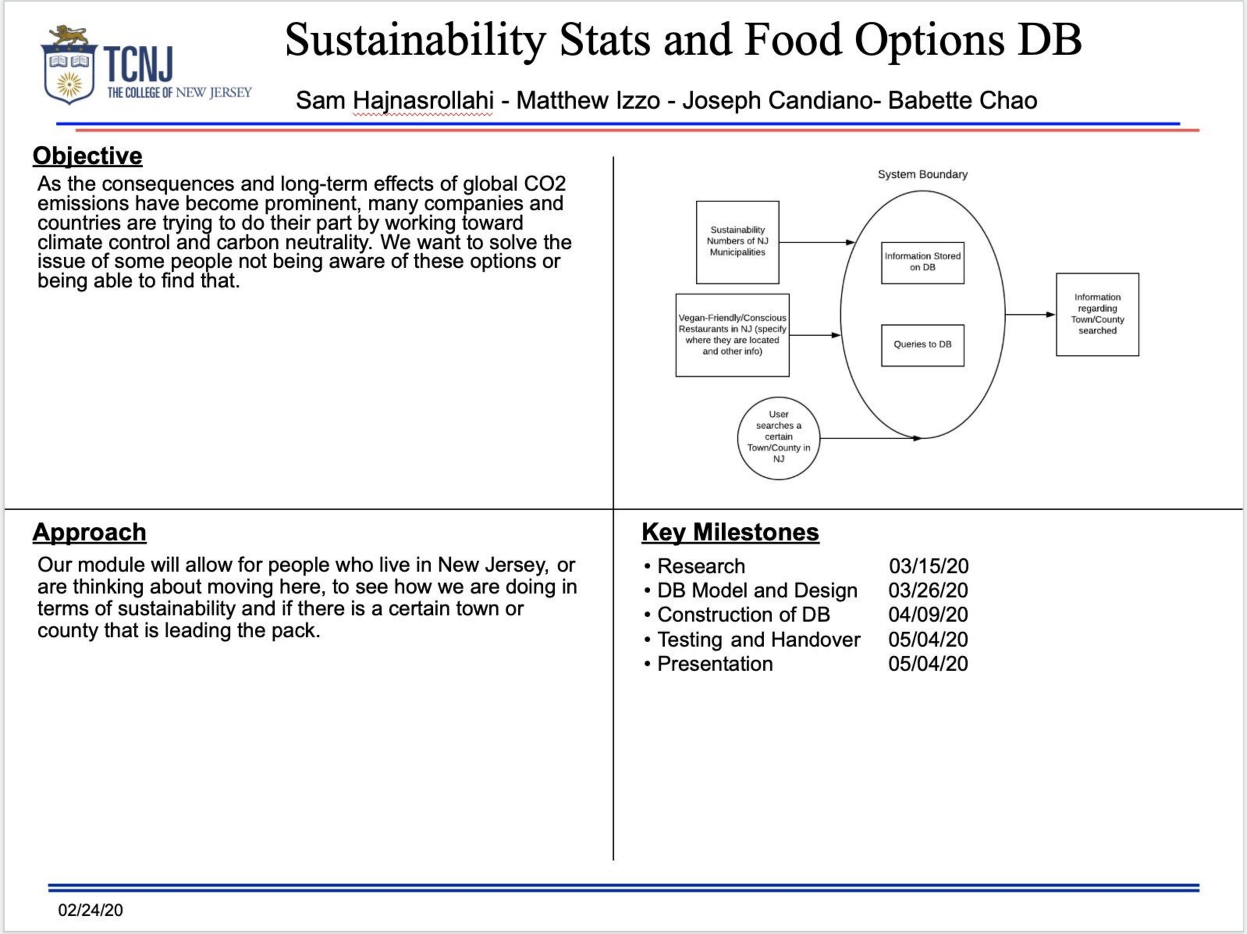
- https://www.postgresqltutorial.com/
- https://www.postgresql.org/docs/8.0/tutorial.html
- https://www.tutorialspoint.com/postgresql/index.htm

Elaboration: Design

Diagrammatic Representation of System Boundary


System Boundary
Sustainability Numbers of NJ Municipalities
Vegan-Friendly/Conscious Restaurants in NJ (specify where they are located and other info)
User searches a certain Town/County in NJ
Information Stored on DB
Queries to DB
Information regarding Town/County searched

1-page quad chart



TCNJ THE COLLEGE OF NEW JERSEY

Sustainability Stats and Food Options DB

Sam Hajnasrollahi - Matthew Izzo - Joseph Candiano- Babette Chao

Objective

As the consequences and long-term effects of global CO2 emissions have become prominent, many companies and countries are trying to do their part by working toward climate control and carbon neutrality. We want to solve the issue of some people not being aware of these options or being able to find that.

System Boundary
Sustainability Numbers of NJ Municipalities
Information Stored on DB
Vegan-Friendly/Conscious Restaurants in NJ (specify where they are located and other info)
Queries to DB
Information regarding Town/County searched
User searches a certain Town/County in NJ

Approach

Our module will allow for people who live in New Jersey, or are thinking about moving here, to see how we are doing in terms of sustainability and if there is a certain town or county that is leading the pack.

Key Milestones

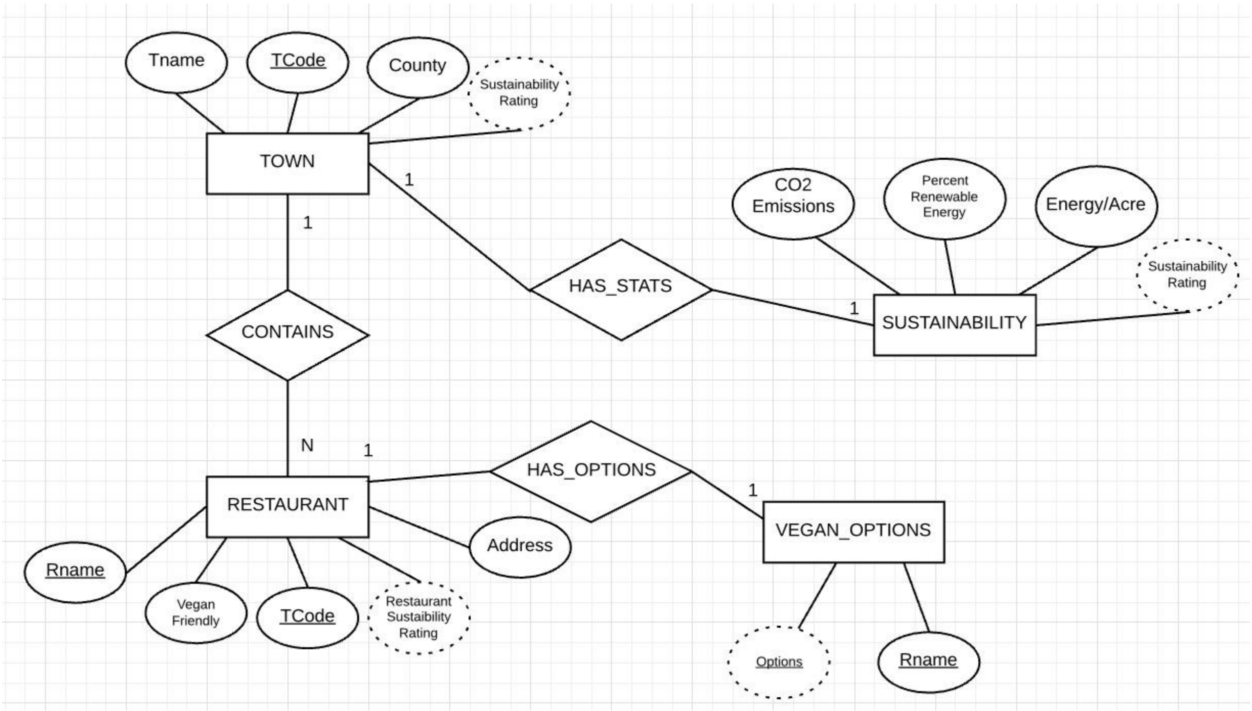
• Research	03/15/20
• DB Model and Design	03/26/20
• Construction of DB	04/09/20
• Testing and Handover	05/04/20
• Presentation	05/04/20

02/24/20

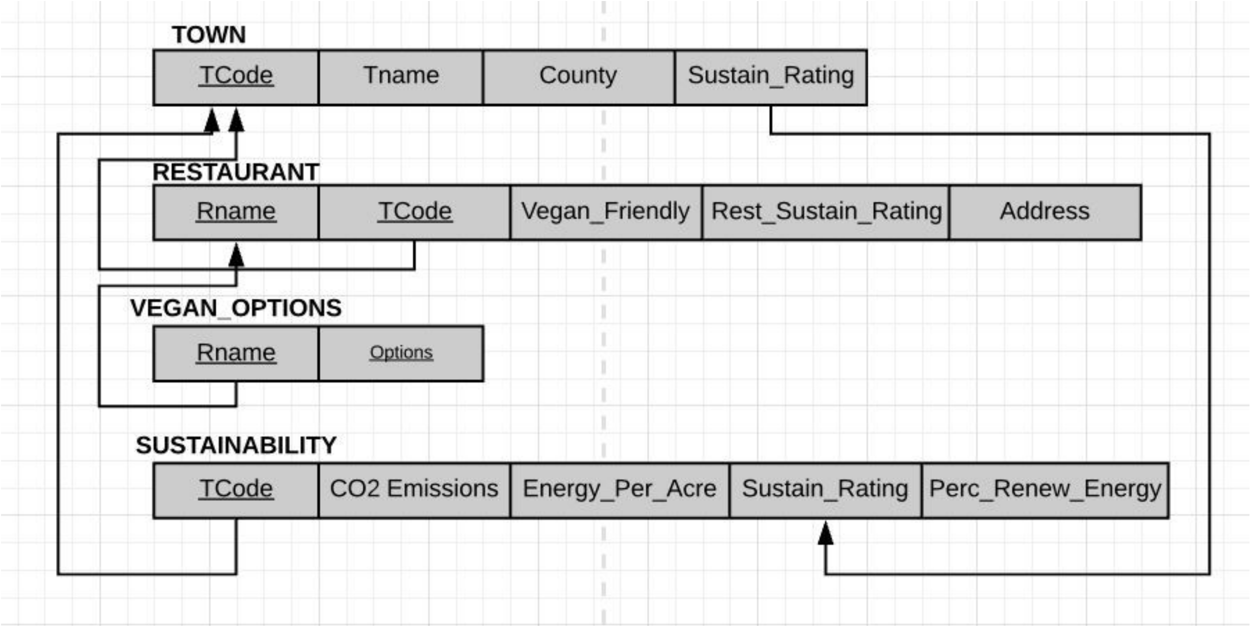
Team Roles

Our integration lead will be Joe Candiano . All members from the databases course will split time coding the project, helping to design and model, and assisting in integration as well. The journalism members will mostly focus on gathering research and data for the database.

ER Diagram



Relational Model



Based on our research and understanding of the social issue addressed here, we are approximating there will be around 565 records in our database, one for each municipality in New Jersey. As for the types of searches, there will be queries to a certain municipality or a whole county in NJ; furthermore, a user can also search for a specific restaurant in NJ and see what sustainable options they have to offer. Lastly, we approximate that the average number of searches will be anywhere from 10-100.

Elaboration: Design

TOWN table

- This table is in BCNF because the attribute, Tcode, is the superkey for this relational schema. Each functional dependency from Tcode is trivial, meaning the resulting tuples from each Tcode are unique and are within the set of Tcode's values.

Tcode	Tname	County	Sustain_Rating
07430	Ringwood	Passaic	Average
08638	Ewing	Mercer	Average
07307	Jersey City	Hudson	Below Average

RESTAURANT table

- This table is in BCNF because it also contains the superkey for this relational schema, Tcode. However, the functional dependency is not trivial because there could be multiple restaurants that are vegan-friendly within the same Tcode.

Rname	TCode	Vegan_Friendly	Rest_Sustain_Rating	Address
Artemio's Prime & Proper	07430	Yes	High	1131 Greenwood Lake TPKE Ringwood, NJ 07456
The Hutton Restaurant & Bar	07307	Yes	High	225 Hutton St Jersey City, NJ 07307

VEGAN_OPTIONS table

- Although this table does not contain the superkey for this relational schema, it is still in BCNF. That is, because each restaurant in the table will have its own unique set of options, which makes the functional dependency trivial.

Rname	Options
Artemio's Prime & Proper	Salads, Substitute Tofu for most entrees, Cauliflower steak
The Hutton Restaurant & Bar	Small Cheese Plates, Chickpea pasta and vegetables, veggie burgers

SUSTAINABILITY table

- This table is in BCNF because it contains the superkey for this relational schema, Tcode, and each town in the table produces its own unique sustainability statistics. This shows that the functional dependency of Tcode is trivial.

Tcode	CO2_Emissions	Energy_Per_Acre_Per_Year	Sustain_Rating	Perc_Renew_Energy
07430	20k - 40k	357 MWh	Average	25%
07307	70k - 5M	300 MWh	Below Average	16%

- Define the different views required. For each view list the data and transaction requirements. Give a few examples of queries, in English, to illustrate.

town_sustain_view:

```
----
SELECT sustain_rating,
Tcode,
Tname,
County,
            ...
FROM Town
------------------------------------------------------------------------------------------ restaurant_sustain_view:
----
```

```
SELECT Rname,
Tcode,
rest_sustain_rating,
Address,
            ...
FROM Restaurant
------------------------------------------------------------------------------------------
restaurant_sustain_vegan_view:
----
SELECT Rname,
Tcode,
rest_sustain_rating,
Address,
Options,
            ...
FROM Restaurant
  JOIN Vegan_Options

------------------------------------------------------------------------------------------ vegan_options_view:
----
SELECT Rname,
 Tcode,
vegan_friendly,
Options,
            ...
FROM Restaurant
  JOIN Vegan_Options

------------------------------------------------------------------------------------------ sustainability_view:
----
SELECT Tcode,
CO2_Emissions,
Sustain_rating,
Options,
            ...
FROM Sustainability

------------------------------------------------------------------------------------------ town_sustainability_view:
----
SELECT Tcode,
CO2_Emissions,
Energy_Per_Acre_Per_Year,
Sustain_Rating,
Perc_Renew_Energy,
Tname,
County, Sustain_Rating

         ...
FROM Town
 JOIN Sustainability
```

- Design a complete set of queries to satisfy the transaction requirements identified in the previous stages.
  - **TOWN Queries**
    - Retrieve all towns from TOWN and display info (Tcode, Tname, County, and/or Sustain_Rating).
    - Retrieve info (Tcode, Tname, County, and/or Sustain_Rating) on a town named ‘Specific Town’ from TOWN.
    - Retrieve info (Tcode, Tname, County, and/or Sustain_Rating) on a town in ‘Specific County’ from TOWN.
    - Retrieve info (Tcode, Tname, County, and/or Sustain_Rating) on a town with Tcode ‘Specific Tcode’ from TOWN.
    - Retrieve towns which have a (high, average, low) sustain ratings from TOWN and display info (Tcode, Tname, County, and/or Sustain_Rating).
  - **RESTAURANT Queries**
    - Retrieve all restaurants from RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, and/or Address).
    - Retrieve restaurants which have a (high, average, low) sustain ratings from RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, and/or Address).
    - Retrieve a restaurant named ‘Specific Restaurant from RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, and/or Address).
    - Retrieve restaurants located in Tcode ‘Specific Tcode’ from RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, and/or Address).
    - Retrieve restaurants which are vegan friendly from RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, and/or Address).
    - Retrieve restaurants located at address ‘Specific Address’ from RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, and/or Address).
  - **VEGAN_OPTIONS Queries**
    - Retrieve all vegan options from VEGAN_OPTIONS and display info (Rname and/or Options).
    - Retrieve list of vegan options from restaurant named ‘Specific Restaurant’ from VEGAN_OPTIONS and display info (Rname and/or Options).
    - Retrieve restaurants which have vegan option ‘Specific Vegan Option’ from VEGAN_OPTIONS and display info (Rname and/or Options). ○

  SUSTAINABILITY Queries
    - Retrieve all sustainability stats from SUSTAINABILITY and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, and/or Perc_Renew_Energy).

        - Retrieve sustainability stats from town with Tcode ‘Specific Tcode’ from SUSTAINABILITY and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, and/or Perc_Renew_Energy).
        - Retrieve sustainability stats on town with CO2 Emissions </>/= ‘Specific Number’ from SUSTAINABILITY and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, and/or Perc_Renew_Energy).
        - Retrieve sustainability stats on town with Energy_Per_Acre_Per_Year </>/= ‘Specific Number’ from SUSTAINABILITY and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, and/or Perc_Renew_Energy).
        - Retrieve towns which have a (high, average, low) sustain ratings from SUSTAINABILITY and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, and/or Perc_Renew_Energy).
        - Retrieve sustainability stats on town with Perc_Renew_Energy </>/= ‘Specific Number’ from SUSTAINABILITY and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, and/or Perc_Renew_Energy).
    - **Examples of Queries Using Multiple Tables**
        - Retrieve towns which have a (high, average, or low) sustain ratings from SUSTAINABILITY and TOWN and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, Perc_Renew_Energy, Tname, County, and/or Sustain_Rating).
        - Retrieve towns which have a restaurant with a (high, average, or low) from TOWN and RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, Address, Tname, County, and/or Sustain_Rating).
        - Retrieve all restaurants in the town named ‘Specific Town Name’ from TOWN and RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, Address, Tname, County, and/or Sustain_Rating).
        - Retrieve all restaurants in the county ‘Specific County’’ from TOWN and RESTAURANT and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, Address, Tname, County, and/or Sustain_Rating).
        - Retrieve all restaurants with (high, average, or low) sustain ratings and vegan options from RESTAURANT and TOWN and display info (Rname, Tcode, Vegan_Friendly, Rest_Sustain_Rating, Address, and/or Vegan_Options).
        - Retrieve town named ‘Specific Town’ from TOWN and SUSTAINABILITY and display info (Tcode, CO2_Emissions, Energy_Per_Acre_Per_Year, Sustain_Rating, Perc_Renew_Energy, Tname, County, and/or Sustain_Rating).

Construction: Tables, Queries, and User Interface

```
CREATE TABLE TOWN
   (Tcode varchar(5) NOT NULL UNIQUE PRIMARY KEY,
   Tname text NOT NULL,
   County text,
   Sustain_rating text NOT NULL);


CREATE TABLE RESTAURANT
   (Rname text NOT NULL UNIQUE PRIMARY KEY,
   Tcode varchar(5) NOT NULL,
   Vegan_friendly boolean,
   Rest_sustain_rating text NOT NULL,
           Address text NOT NULL,
           FOREIGN KEY (Tcode) REFERENCES TOWN (Tcode));


CREATE TABLE VEGAN_OPTIONS
   (Rname text NOT NULL PRIMARY KEY,
   Options text,
   FOREIGN KEY (Rname) REFERENCES RESTAURANT (Rname));


CREATE TABLE SUSTAINABILITY
   (Tcode varchar(5) NOT NULL PRIMARY KEY,
   CO2_Emissions decimal NOT NULL,
   Energy_per_acre integer NOT NULL,
   Sustain_rating text NOT NULL,
   Perc_renew_energy integer NOT NULL,
   FOREIGN KEY (Tcode) REFERENCES TOWN (Tcode));
```

Python Script to Read in, Format, and Load in

```
# Script to read in data from 4 files and then write them into 4 tables in our database

#! /usr/bin/python2

import psycopg2 from
config import config

if __name__ == '__main__':
   # Iniatlize connection
conn = None
          # read connection parameters
params = config()

          # connect to the PostgreSQL server
print('Connecting to the %s database...' % params['database'])
conn = psycopg2.connect(**params)     print('Connected.\n')
conn.autocommit = True

   # create a cursor
cur = conn.cursor()

   print('Loading in town data...')

   # Open file with town data
f = open("town.txt")

   # Go through line by line
for y in f:
           # Format and get each part of the
line       x = (y.split('; '))             one =
str(x[0].strip())   two = str(x[1].strip())
three = str(x[2].strip())    four =
str(x[3].strip())
           # Insert into town table in form of a psql query    cur.execute("INSERT INTO TOWN
VALUES('%s', '%s', '%s', '%s');" %(one, two, three, four))
   # Close file
f.close()

   print('Loading in restaurant data...')
   # Open file with restaurants data
f = open("restaurants.txt")

   # Go through line by line
for y in f:
            # Format and get each part of the
line        x = (y.split('; '))              one =
str(x[0].strip())   two = str(x[1].strip())
            three = str(x[2].strip())
 four = str(x[3].strip())    five =
str(x[4].strip())
            # Insert into restaurant table in form of a psql query         cur.execute("INSERT INTO
RESTAURANT VALUES('%s', '%s', '%s', '%s', '%s');" %(one, two, three, four, five))     # Close file
f.close()

   print('Loading in vegan options data...')

   # Open file with vegan options data
f = open("vegan_options.txt")

   # Go through line by line
for y in f:
            # Format and get each part of the
line        x = (y.split('; '))              one =
str(x[0].strip())   two = str(x[1].strip())
            # Insert into vegan options table in form of a psql query
cur.execute("INSERT INTO VEGAN_OPTIONS VALUES('%s', '%s');" %(one, two))
   # Close file
f.close()

   print('Loading in sustainability data...')

   # Open file with sustainability data
f = open("sustainability.txt")

   # Go through line by line
for y in f:
            # Format and get each part of the
line        x = (y.split('; '))              one =
str(x[0].strip())   two = str(x[1].strip())
three = str(x[2].strip())    four =
str(x[3].strip())   five = str(x[4].strip())
            # Insert into sustainability table in form of a psql query   cur.execute("INSERT INTO
SUSTAINABILITY VALUES('%s', '%s', '%s', '%s', '%s');" %(one, two, three, four, five))
```

```
    # Close file
    f.close()

    print('All data succesfully loaded and inserted.')

    # Close connection to db
cur.close()
```

Queries

```
SELECT * FROM TOWN;


SELECT * FROM TOWN
WHERE Tname = 'Atlantic City';


SELECT Tcode, Tname, Sustain_rating FROM TOWN
WHERE County = 'Bergen';


SELECT Tname, County, Sustain_rating FROM TOWN
WHERE Tcode = '08601';


SELECT * FROM TOWN
WHERE Sustain_rating = 'High';


SELECT * FROM RESTAURANT;


SELECT Rname, Tcode, Address FROM RESTAURANT
WHERE Rest_sustain_rating = 'Average';


SELECT Rname, Tcode, Address FROM RESTAURANT
WHERE Rname = 'Greens and Grains';


SELECT Rname, Tcode, Rest_sustain_rating, Address FROM RESTAURANT
WHERE Tcode = '08723';
```

```sql
SELECT * FROM RESTAURANT
WHERE Vegan_friendly = 't';


SELECT * FROM RESTAURANT
WHERE Address = '4 Hamburg Ave (at Loomis), Sussex, New Jersey';


SELECT * FROM VEGAN_OPTIONS;


SELECT * FROM VEGAN_OPTIONS
WHERE Rname = 'Leatherhead Pub';


SELECT Rname FROM VEGAN_OPTIONS
WHERE Options = 'hummus plate, cauliflower pizza, fried artichoke plus salads, veggie sandwiches';


SELECT * FROM SUSTAINABILITY;


SELECT * FROM SUSTAINABILITY
WHERE Tcode = '08043';


SELECT * FROM SUSTAINABILITY
WHERE CO2_Emissions <= 38;


SELECT Tcode, Energy_per_acre FROM SUSTAINABILITY
WHERE Energy_per_acre >= 15;


SELECT Tcode, Sustain_rating FROM SUSTAINABILITY
WHERE Sustain_rating = 'Average';


SELECT Tcode, Perc_renew_energy FROM SUSTAINABILITY
WHERE Perc_renew_energy >= 25;
```

```
SELECT Tname, County, SUSTAINABILITY.*
FROM TOWN JOIN SUSTAINABILITY on TOWN.Tcode = SUSTAINABILITY.Tcode
WHERE TOWN.Sustain_rating = 'Average';


SELECT RESTAURANT.Rname, Tcode, Vegan_Friendly, Vegan_options
FROM RESTAURANT JOIN VEGAN_OPTIONS ON RESTAURANT.Rname =
VEGAN_OPTIONS.Rname
WHERE Vegan_Friendly = 't';


SELECT Tname, Sustain_Rating, Rname, Address
FROM TOWN JOIN RESTAURANT ON RESTAURANT.Tcode = TOWN.Tcode
WHERE Sustain_rating = 'Low';

SELECT COUNT(TOWN.Tcode) AS COUNT_TOWNS_OVER_15_EPA
FROM TOWN JOIN SUSTAINABILITY ON TOWN.Tcode = SUSTAINABILITY.Tcode WHERE
Energy_per_acre > 15;

SELECT Tname, Rname, SUSTAINABILITY.Sustain_rating, CO2_Emissions
FROM
(SELECT TOWN.Tname, RESTAURANT.*
FROM TOWN JOIN RESTAURANT on RESTAURANT.Tcode = TOWN.Tcode
WHERE Sustain_rating = 'Low') AS TOWN_RESTAURANT_LOW
JOIN SUSTAINABILITY ON SUSTAINABILITY.Tcode = TOWN_RESTAURANT_LOW.Tcode
WHERE CO2_Emissions < 35;

SELECT Tname, Rname, Vegan_Options
FROM
(SELECT Vegan_Options, RESTAURANT.Rname, Tcode
FROM VEGAN_OPTIONS JOIN RESTAURANT on RESTAURANT.Rname =
VEGAN_OPTIONS.Rname)
AS RESTAURANT_VEGAN_OPTIONS
JOIN TOWN ON TOWN.Tcode = RESTAURANT_VEGAN_OPTIONS.Tcode
WHERE County = 'Bergen';
```

PHP Code for User Interface

```
<!-- CSC 315 - NJSus Database Final Project -->
<?php
session_start();
?>
<!DOCTYPE html>
<head>
    <title>Search the NJSus Database</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <style>
        li {
            list-style: none;
        }
        body {
            background-image:
url("https://www.toptal.com/designers/subtlepatterns/patterns/more-leaves-on-green.png");
            background-repeat: repeat;
            font-family: "Trebuchet MS", Helvetica, sans-serif;
        }
    </style>
</head>
<body>
    <h1 style="text-align:center">NJ Sustainability Stats Database</h1>
    <ul style="text-align:center">
        <form name="display" action="test.php" method="POST">
            <li>What kind of data are you looking for?:</li>
            <li><input style="border-radius: 5px; padding: 5px; font-size: 15px;
transition: all 0.3s;" type="submit" name="town" value="Towns" /></li><br>
            <li><input style="border-radius: 5px; padding: 5px; font-size: 15px;
transition: all 0.3s;" type="submit" name="restaurant" value="Restaurants" /></li><br>
            <li><input style="border-radius: 5px; padding: 5px; font-size: 15px;
transition: all 0.3s;" type="submit" name="vegan_options" value="Vegan Options" /></li><br>
            <li><input style="border-radius: 5px; padding: 5px; font-size: 15px;
transition: all 0.3s;" type="submit" name="sustainability" value="Sustainability" /></li><br>
            <li><input style="border-radius: 5px; padding: 5px; font-size: 15px;
transition: all 0.3s;" type="submit" name="query" value="I'd like to enter my own PSQL query."
/></li>
        </form>
    </ul>
    <?php
        // Connect to database
        $db = pg_connect("host=localhost port=5432 dbname=njsus user=osc
password=osc");
        //Check for what option was chosen from the initial dropdown
        if(isset($_POST['town'])){
            $_SESSION["chosenTable"] = "town";
        } elseif(isset($_POST['restaurant'])){
            $_SESSION["chosenTable"] = "restaurant";
        } elseif(isset($_POST['vegan_options'])){
            $_SESSION["chosenTable"] = "vegan_options";
        } elseif(isset($_POST['sustainability'])){
            $_SESSION["chosenTable"] = "sustainability";
        } elseif(isset($_POST['query'])){
            $_SESSION["chosenTable"] = "query";
        }

        // If town is chosen display the following options
        if($_SESSION["chosenTable"] == 'town'){
    ?>
            <ul style="text-align:center">
                <form name="display" action="test.php" method="POST">
                    <li>Check all the data you'd like to get:</li>
                    <li><input type="checkbox" name="check_list[]"
value="tcode"/><label> Town Code </label></li>
                    <li><input type="checkbox" name="check_list[]"
value="tname"/><label> Town Name </label></li>
                    <li><input type="checkbox" name="check_list[]"
value="county"/><label> County </label></li>
                    <li><input type="checkbox" name="check_list[]"
value="sustain_rating"/><label> Sustainability </label></li><br>

                    <li>If you'd like, select a datapoint to compare against a
value:</li>
                    <li><select id="whereTown" name="where">
                        <option value="tcode">Town Code</option>
                        <option value="tname">Town Name</option>
                        <option value="county">County</option>
                        <option value="sustain_rating">Sustainability
Rating</option>
                    </select></li><br>

                    <li>How would you like to check that value?:</li>
                    <li><select id="checkTown" name="check">
                        <option value="=">Is/Is equal to</option>
                        <option value="<">Is less than</option>
                        <option value=">">Is greater than</option>
                    </select></li><br>
```

```
                        <li>And that value is?</li>
                        <li><input type ="text" name="value"></li>
                        <li><input type="submit" name="submit" value="Submit"
/></li>
                    </form>
                </ul>
    <?php
        // Make selected into a comma seperated list
        $selectList = implode(", ", $_POST['check_list']);
        //Format compare value with basic security measures and proper layout
        $compareValue = "'".strip_tags(trim($_POST[value]))."'";
        //Check if a value is included
        if($_POST[value]){
            $result = pg_query($db, "SELECT $selectList FROM TOWN WHERE
$_POST[where] $_POST[check] $compareValue;");
        // If just a select from query
        }elseif($selectList){
            $result = pg_query($db, "SELECT $selectList FROM TOWN;");
        }
        // If submit is clicked then we start making tables
        if (isset($_POST['submit'])){

        //Now move onto building the table
    ?>

    <table align="center" border="1px" style="width:600px; line-height:40px; background-
color: green;">
            <tr>
                <th colspan="4"><h2>Towns</h2></th>
            </tr>
            <t>
                    <th> Town Code </th>
                    <th> Town Name </th>
                    <th> County </th>
                    <th> Sustainability Rating </th>
            </t>
            <?php
                // Go through each row and print results
                while($row = pg_fetch_assoc($result)) {
            ?>
                    <tr>
                        <td><?php echo $row['tcode']; ?></td>
                        <td><?php echo $row['tname']; ?></td>
                        <td><?php echo $row['county']; ?></td>
                        <td><?php echo $row['sustain_rating']; ?></td>
                    </tr>
                <?php
                    }
                ?>

        <?php
            }
        // If restaurant was the initial chosen option
        }elseif($_SESSION["chosenTable"] == 'restaurant'){
    ?>
            <ul style="text-align:center">
                <form name="display" action="test.php" method="POST">
                    <li>Check all the data you'd like to get:</li>
                    <li><input type="checkbox" name="check_list[]"
value="rname"/><label> Restaurant Name </label></li>
                    <li><input type="checkbox" name="check_list[]"
value="tcode"/><label> Town Code </label></li>
                    <li><input type="checkbox" name="check_list[]"
value="vegan_friendly"/><label> Vegan Friendly </label></li>
                    <li><input type="checkbox" name="check_list[]"
value="rest_sustain_rating"/><label> Restaurant Sustainability Rating </label></li>
                    <li><input type="checkbox" name="check_list[]"
value="address"/><label> Address </label></li><br>

                    <li>If you'd like, select a datapoint to compare against a
value:</li>
                    <li><select id="whereTown" name="where">
                        <option value="rname">Restaurant Name</option>
                        <option value="tcode">Town Code</option>
                        <option value="vegan_friendly">Vegan
Friendly</option>
                        <option value="rest_sustain_rating">Restaurant
Sustainability Rating</option>
                        <option value="address">Address</option>
                    </select></li><br>

                    <li>How would you like to check that value?:</li>
                    <li><select id="checkTown" name="check">
                        <option value="=">Is/Is equal to</option>
                        <option value="<">Is less than</option>
                        <option value=">">Is greater than</option>
                    </select></li><br>
                    <li>And that value is?</li>
                    <li><input type ="text" name="value"></li>
                    <li><input type="submit" name="submit" value="Submit"
/></li>
                </form>
            </ul>
    <?php
        // Make selected into a comma seperated list
        $selectList = implode(", ", $_POST['check_list']);
        //Format compare value with basic security measures and proper layout
        $compareValue = "'".strip_tags(trim($_POST[value]))."'";
        //Check if a value is included
        if($_POST[value]){
            $result = pg_query($db, "SELECT $selectList FROM RESTAURANT WHERE
$_POST[where] $_POST[check] $compareValue;");
        // If just a select from query
        }elseif($selectList){
            $result = pg_query($db, "SELECT $selectList FROM RESTAURANT;");
        }
        // If submit is clicked then we start making tables
        if (isset($_POST['submit'])){
    ?>
    <table align="center" border="1px" style="width:600px; line-height:40px; background-
color: green;">
            <tr>
                <th colspan="5"><h2>Restaurants</h2></th>
            </tr>
            <t>
                    <th> Restaurant Name </th>
                    <th> Town Code </th>
                    <th> Vegan Friendly </th>
                    <th> Restaurant Sustainability Rating </th>
                    <th> Address </th>
            </t>
            <?php
                // Go through each row and print results
                while($row = pg_fetch_assoc($result))
                {
            ?>
                    <tr>
                        <td><?php echo $row['rname']; ?></td>
                        <td><?php echo $row['tcode']; ?></td>
                        <td><?php echo $row['vegan_friendly']; ?></td>
                        <td><?php echo $row['rest_sustain_rating'];
?></td>
                        <td><?php echo $row['address']; ?></td>
                    </tr>
            <?php
                }
        }
    //If vegan options is the initially chosen option
    }elseif($_SESSION["chosenTable"] == 'vegan_options'){
    ?>
        <ul style="text-align:center">
            <form name="display" action="test.php" method="POST">
                <li>Check all the data you'd like to get:</li>
                <li><input type="checkbox" name="check_list[]"
value="rname"/><label> Restaurant Name </label></li>
                <li><input type="checkbox" name="check_list[]"
value="options"/><label> Options </label></li><br>

                <li>If you'd like, select a datapoint to compare against a value:</li>
                <li><select id="whereTown" name="where">
                    <option value="rname">Restaurant Name</option>
                    <option value="options">Options</option>
                </select></li><br>

                <li>How would you like to check that value?:</li>
                <li><select id="checkTown" name="check">
                    <option value="=">Is/Is equal to</option>
                    <option value="<">Is less than</option>
                    <option value=">">Is greater than</option>
                </select></li><br>

                <li>And that value is?</li>
                <li><input type ="text" name="value"></li>
                <li><input type="submit" name="submit" value="Submit" /></li>
            </form>
        </ul>
    <?php
        // Make selected into a comma seperated list
        $selectList = implode(", ", $_POST['check_list']);
        //Format compare value with basic security measures and proper layout
        $compareValue = "'".strip_tags(trim($_POST[value]))."'";
        //Check if a value is included
        if($_POST[value]){
            $result = pg_query($db, "SELECT $selectList FROM VEGAN_OPTIONS
WHERE $_POST[where] $_POST[check] $compareValue;");
        // If just a select from query
        }elseif($selectList){
            $result = pg_query($db, "SELECT $selectList FROM VEGAN_OPTIONS;");
        }
        // If submit is clicked then we start making tables
        if (isset($_POST['submit'])){
    ?>

    <table align="center" border="1px" style="width:600px; line-height:40px; background-
color: green;">
            <tr>
                    <th colspan="2"><h2>Vegan Options</h2></th>
                </tr>
                <t>
                        <th> Restaurant Name </th>
                        <th> Options </th>
                </t>
                <?php
                    // Go through each row and print results
                    while($row = pg_fetch_assoc($result)) {
                ?>
                    <tr>
                        <td><?php echo $row['rname']; ?></td>
                        <td><?php echo $row['options']; ?></td>
                    </tr>
                <?php
                }
            }
        // If sustainability is chosen intially
        }elseif($_SESSION["chosenTable"] == 'sustainability'){
    ?>
        <ul style="text-align:center">
            <form name="display" action="test.php" method="POST">
                <li>Check all the data you'd like to get:</li>
                <li><input type="checkbox" name="check_list[]"
value="tcode"/><label> Town Code </label></li>
                <li><input type="checkbox" name="check_list[]"
value="co2_emissions"/><label> CO2 Emissions </label></li>
                <li><input type="checkbox" name="check_list[]"
value="energy_per_acre"/><label> Energy Per Acre </label></li>
                <li><input type="checkbox" name="check_list[]"
value="sustain_rating"/><label> Sustainability Rating </label></li>
```

```
                                        <li><input type="checkbox" name="check_list[]"
value="perc_renew_energy"/><label> Percent Renewable Energy </label></li><br>

                                        <li>If you'd like, select a datapoint to compare against a value:</li>
                                        <li><select id="whereTown" name="where">
                                                  <option value="tcode">Town Code</option>
                                                  <option value="co2_emissions">CO2 Emissions</option>
                                                  <option value="energy_per_acre">Energy Per
Acre</option>
                                                  <option value="sustain_rating">Sustainability
Rating</option>
                                                  <option value="perc_renew_energy">Percent Renewable
Energy</option>
                                        </select></li><br>

                                        <li>How would you like to check that value?:</li>
                                        <li><select id="checkTown" name="check">
                                                  <option value="=">Is/Is equal to</option>
                                                  <option value="<">Is less than</option>
                                                  <option value=">">Is greater than</option>
                                        </select></li><br>

                                        <li>And that value is?</li>
                                        <li><input type ="text" name="value"></li>
                                        <li><input type="submit" name="submit" value="Submit" /></li>
                              </form>
                    </ul>

          <?php
                    // Make selected into a comma seperated list
                    $selectList = implode(", ", $_POST['check_list']);
                    //Format compare value with basic security measures and proper layout
                    $compareValue = "'".strip_tags(trim($_POST[value]))."'";
                    //Check if a value is included
                    if($_POST[value]){
                              $result = pg_query($db, "SELECT $selectList FROM SUSTAINABILITY WHERE
$_POST[where] $_POST[check] $compareValue;");
                    // If just a select from query
                    }elseif($selectList){
                              $result = pg_query($db, "SELECT $selectList FROM SUSTAINABILITY;");
                    }
                    // If submit is clicked then we start making tables
                    if (isset($_POST['submit'])){
          ?>
```

```
    <table align="center" border="1px" style="width:600px; line-height:40px; background-
color: green;">
            <tr>
                    <th colspan="5"><h2>Sustainability</h2></th>
                </tr>
                <t>
                        <th> Town Code </th>
                        <th> CO2 Emissions </th>
                        <th> Energy Per Acre </th>
                        <th> Sustainability Rating </th>
                        <th> Percent Renewable Energy </th>
                </t>
                <?php
                    // Go through each row and print results
                    while($row = pg_fetch_assoc($result))
                    {
                ?>
                    <tr>
                        <td><?php echo $row['tcode']; ?></td>
                        <td><?php echo $row['co2_emissions']; ?></td>
                        <td><?php echo $row['energy_per_acre']; ?></td>
                        <td><?php echo $row['sustain_rating']; ?></td>
                        <td><?php echo $row['perc_renew_energy'];
?></td>
                    </tr>
                <?php
                    }
                }
            //If the advanced query option is selected.
            }elseif($_SESSION["chosenTable"] == 'query'){
                ?>
                <ul style="text-align:center">
                    <form name="display" action="test.php" method="POST" >
                        <li>Enter the query below</li>
                        <li><input type ="text" name="qry"></li>
                        <li><input type="submit" name="submit"
value="Submit" /></li>
                    </form>
                </ul>
                <?php
                //Once submitted
                if (isset($_POST['submit'])){
                    // Some basic validation security formatting
                    $result = pg_query($db,strip_tags(trim($_POST[qry])))
                ?>
                <table align="center" border="1px" style="width:600px; line-
height:40px; background-color: green;">
                    <tr>
                        <th colspan="4"><h2>Advanced Query
Results</h2></th>
                    </tr>
                    <?php
                        // Go through each resulting row and format/print
results
                        while($row = pg_fetch_row($result))
                        {
                    ?>
                            <tr>
                                <td><?php
                                // Get name of each field and print in
one row

                                $j = pg_num_fields($result);
                                for ($i=0; $i<$j; $i++){
                                    echo(pg_field_name($result,
$i));

                                    echo "<br>";
                                }
                                ?></td>
                                <td><?php
                                // Print values in other row
                                echo '<pre>';
                                print_r($row);
                                echo '</pre>';
                                ?></td>
                            </tr>
                        <?php
                            }
                        ?>
                </table>
            <?php
            }
        }
        ?>
    </table>
    <h4 style="text-align:center">Instructions</h4>
    <p style="text-align:center;">
```

```
                    First select the category of data you'd like to retrieve from the database by
clicking a button. <br>
                    Then proceed to choose what types of data you'd like from that dataset. <br>
                    You can then press submit or you can further complexify the results by comparing
data. <br>
                    - Choose which datapoint you'd like to compare against a value. <br>
                    - Then select from the dropdown how you'd like to compare it. <br>
                    - Lastly, enter the value which you'd like to compare the datapoint against
(Remember to use correct capatlization so a match can be found in the database!). <br>
                    If desired, you can enter your own PSQL query and submit it for results.
          </p>
</body>
</html>
```