Earth Similarity Index Analysis

Naga Jyothi Madamanchi

***Abstract*—** The purpose of this project is to utilize the publicly available exo-planet database to analyze 4 variants of the ESI (Earth Similarity Index) equation. The sub tables generated by the use of each variant will be compared to one another.

***Index Terms*—** ESI (Earth Similarity Index), Transit, Stellar Flux, Standard ESI, Weighted ESI, Red Dwarf Star, Tidally Locked, Main Sequence Star, Custom ESI, Revised ESI, R Language, R Studio, Identifying Relationship, Primary Key, Gaben, PHP, CSS

I. Introduction

In 1992, radio astronomers Aleksander Wolszcan and Dale Frail discovered two planets orbiting the pulsar PSR 1256+12. This discovery became the first of thousands to come in the following years. With the search for a habitable exoplanet now on the horizon, astronomer Schulze-Makuch and his team developed the ESI (Earth Similarity Index) as a means to determine how similar was an exo-planet to Earth [7].

The implementation of the ESI saw the rise numerous potential Earth-like planets. One major hurdle that plagued the scientists however was the limits in the data they obtained from telescopes. The transit method, the most popular of the observation methods, relied on the finding of a dip in brightness of the host star. That dip in brightness, and the periodical recurrence of that same dip would indicate the existence of an planet [1].

II. NASA's ESI Variants

Early methods of exo-planet discovery provided a limited amount of data. Most of the available information consisted of the stellar flux, and planet radius. The stellar flux, the measure of radiation a planet receives from its host star, became the focal point of the first equation, the Standard ESI equation. Earth's radius (Re) and stellar flux (Fe) were used as the baseline for the equation.

Standard ESI:

$$\mathrm{ESI}(F,R) = 1 - \sqrt{\frac{1}{2}\left[\left(\frac{F-Fe}{F+Fe}\right)^2 + \left(\frac{R-Re}{R+Re}\right)^2\right]}$$

Highest ESI: Teegarden's Star B = 0.9502

The integration of two variables in the Standard ESI yielded Teegarden's Star B as the exo-planet with the greatest similarity to Earth. The planet's orbit in Teegarden's Star's habitable zone and its very similar radius to Earth reflect the high ESI score. However many in the scientific community still felt the use of two variables was inadequate. As a consequence, a new Weighted ESI equation was formulated to account for more factors.

The Weighted ESI equation utilizes a total of 5 variables: stellar flux, radius, density, escape velocity, and temperature. Each one of these variables has weights that affect the sensitivity of each factor. Temperature holds the highest weight because of the important role it plays in the habitability of planets.

Weighted ESI variables:

Variables	Weights
Flux	1
Radius	0.57
Density	1.07
Escape Velocity	0.70
Temperature	5.58

Weighted ESI:

$$ESI(F,R,D,E,T)=1-\sqrt{\frac{1}{5}\left[\left(\frac{F-Fe}{F+Fe}\right)^2+(0.57)\left(\frac{R-Re}{R+Re}\right)^2+(1.07)\left(\frac{D-De}{D+De}\right)^2+(0.70)\left(\frac{E-Ee}{E+Ee}\right)^2+(5.58)\left(\frac{T-Te}{T+Te}\right)^2\right]}$$

Highest ESI: Teegarden's Star B = 0.9636

The Weighted ESI's 5 variables deem Teegarden's Star B once again as the planet with the greatest similarity to Earth. The planet's similar radius, mass, and density to Earth suggests that the conditions on that world may be very similar to that of our planet. However Teegarden's Star B does have one startling distinction, the planet orbits a red dwarf star with a mass roughly 9 percent that of our Sun [3].

III. Custom ESI Variants

Red dwarf planetary systems are host to a multitude of issues that may prevent the development of life. The first notable problem lies with the range of the habitable zone. Due to the cooler temperature of red dwarfs, planets in the habitable zone often have orbital periods that range from 5 to 12 days. The close proximity of the planet to its star causes the planet to be tidally locked. One side of the planet would always face the star, the other side would be left in perpetual night. The absence of a day/night cycle would render most of the surface of such planets unsuitable for life.

Another major problem that plagues red dwarf systems is the radiation of the stars. Red dwarf stars have a life cycle that is considerably longer than that of Sun-sized main sequence stars. Sun-massed stars often live 10 billion years. Red dwarf stars can ensure as long as 1 trillion years. Younger stars tend to send out frequent and more solar storms. Every red dwarf star in the universe is in its infancy. As a consequence, many of the planets orbiting such systems are much more irradiated than Earth or even Mercury [9].

Older ESI models fail to take the mass of the star into account. A planet such as Teegarden's Star B perhaps has a temperature comparable to that of Earth. But it will likely have a surface as irradiated as the ruins of Chernobyl. To mitigate this problem, I have formulated two new variants of the ESI equation. The first of these variants takes in a total of 9 variables, star temperature and star mass among them, all unweighted. I have titled this first equation the Custom ESI. The structure of the equation is similar to that of the Standard ESI. The one notable difference is the use of 9 variables. The equation below is simply a compressed representation of a longer Custom ESI formula.

Variables: Flux, Radius, Gravity, Planet Mass, Temperature, Star Temperature, Star Mass, Orbital Period, Density.

Custom ESI:

$$\mathrm{ESI}(F,R,G,M,T,K,S,O,D) = \prod_{i=1}^{9}\left(1-\left|\frac{x_i - x_{io}}{x_i + x_{io}}\right|\right)$$

Highest ESI: Venus = 0.7996

One problem that is immediately apparent with the Custom ESI formula is its inaccuracy. As many in the scientific community are aware, Venus is far too hot to be a habitable planet. The reason for this inaccuracy is due to the failure of the equation to weigh its values. Temperature should hold the highest weight. Factors such as star mass should be included, but they should not allow planets such as Venus to be counted as Earth analogues.

Revised ESI variables:

Variables	Weights
Flux	3.2
Radius	0.57
Gravity	4.75
Planet Mass	0.4
Temperature	10.58
Star Temperature	1
Star Mass	1
Orbital Period	1
Density	2.8

Revised ESI:

$$\mathrm{ESI}(F,R,G,M,T,K,S,O,D) = \prod_{i=1}^{9}\left(1-\left|\frac{x_i - x_{io}}{x_i + x_{io}}\right|\right)^{\frac{wi}{9}}$$

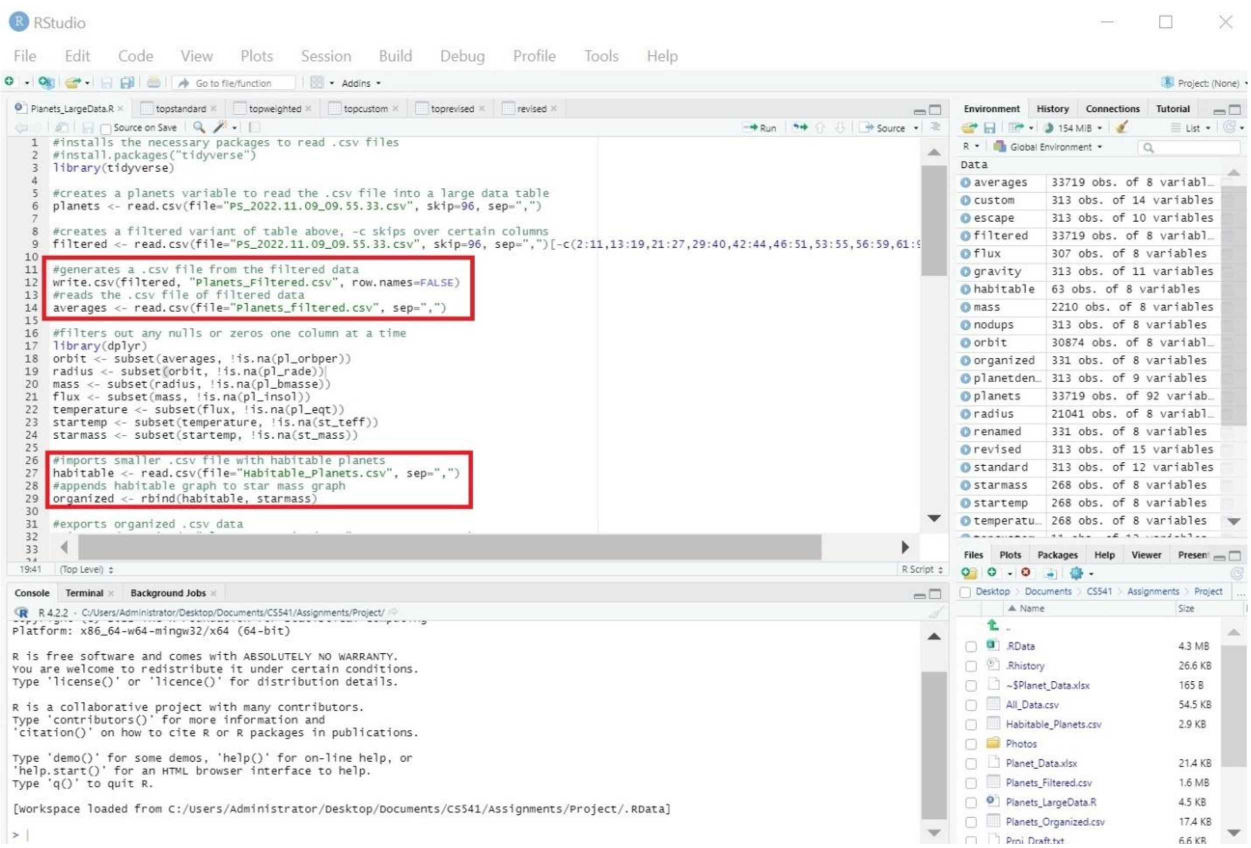
Highest ESI: Kepler-452 B = 0.8478

Kepler-452 B orbits a star similar in mass to the Sun. It has a temperature and orbital period similar to that of Earth. The one notable difference is the density of the planet. Earth has a density of 5.51 g/cm3, Kepler-452 B has a density of 4.21 g/cm3. The exo-planet has a larger mass and radius than Earth. The lower density suggests the planet may be a semi gaseous planet or an ocean planet.

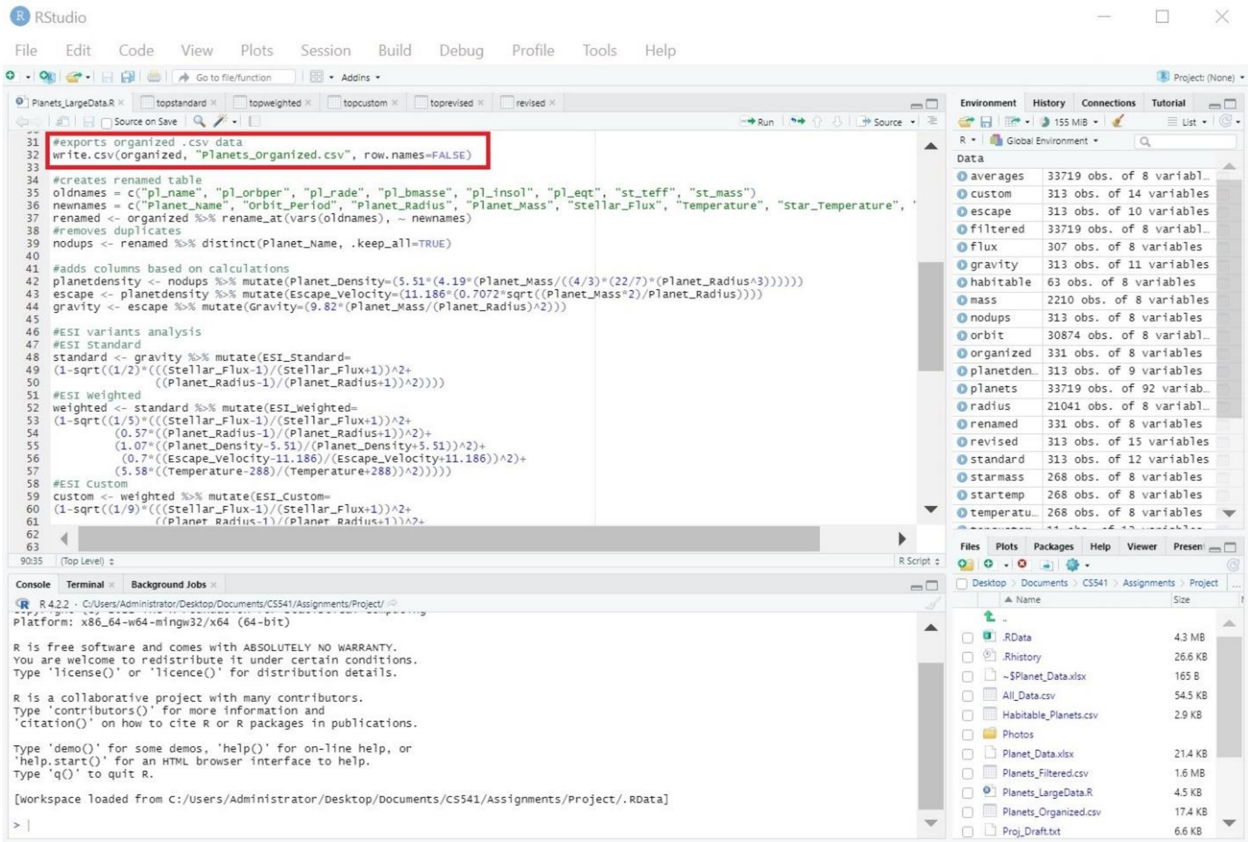
The Revised ESI is still not as reliable as it could be. However with further revisions, the Revised ESI could one day become a highly accurate method for determining the habitability of exo-planets. In the time being, this formula, and the other 3 formulas, will serve as the algorithms I will test on NASA's exoplanet catalog.

IV. The Setup

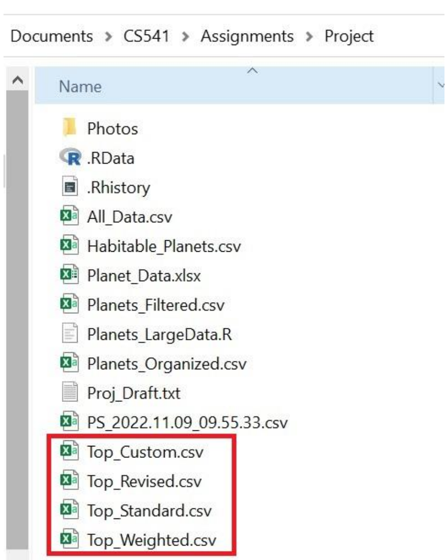
The setup for the analysis of NASA's exoplanet catalog begins with the installation of the language R and the download of the program R Studio. The R Studio streamlines the use of the "PS_2022.11.09_09.55.33.csv" file which contains the entirety of NASA's planet database. The original database contains 92 columns in total, many of which are not needed. The "Planets_LargeData.R" file contains the code needed to filter out the unnecessary columns. The new file "Planets_Filtered.csv", is used for the rest of the analysis.



The data from "Planets_Filtered.csv" file is passed through a series of filters that remove any null data entries column by column. The total row count is reduced from 33,719 to 268. Data concatenated from the "Habitable_Planets.csv" file generated from the "Planet_Data.xlsx" excel file increases the total row count to 331. The data in the excel sheet was sourced from the Planetary Habitability Laboratory (PHL) [2].



Data extracted from one data set to the other is saved as a .csv after an important checkpoint is reached. In the case of the "Planets_Organized.csv" file, the important checkpoint equates to the filtration of all blank data. The ESI equations require between 2 to 9 variables. If any one of the variables are blank, the equations return a value of zero. A number of columns that correspond to one of the 4 ESI variants are generated for the "All_Data.csv" file. The data from that one file is used to create 4 .csv files, each containing the top 10 calculated ESI values and their corresponding planets.

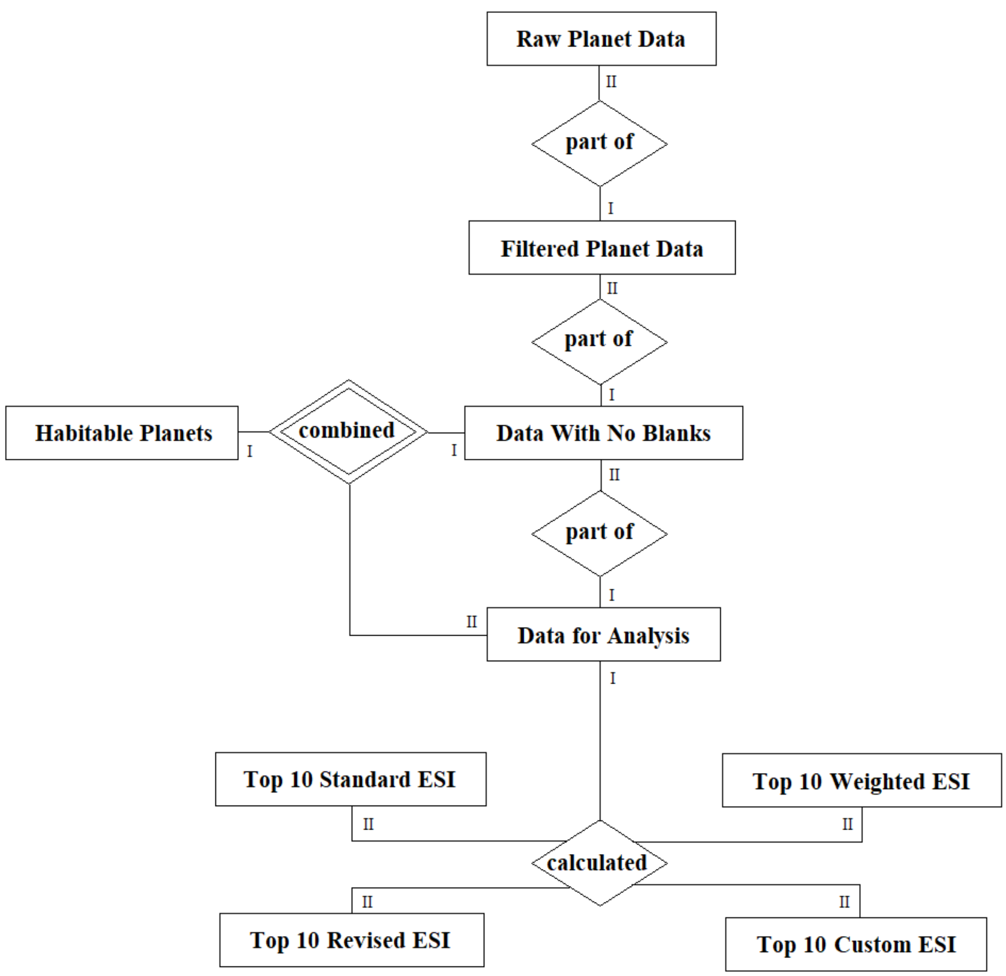


Once I have created all the necessary .csv files, I then transfer the generated databases into the "CS541" class Gaben folder. I utilize the SQL language to save the data. The reason I created the tables as .csv files was to have a healthy copy of the in the event that one of the databases gets corrupted. I could simply make a new functional version of the database if needed.

V. ESI Database Diagrams

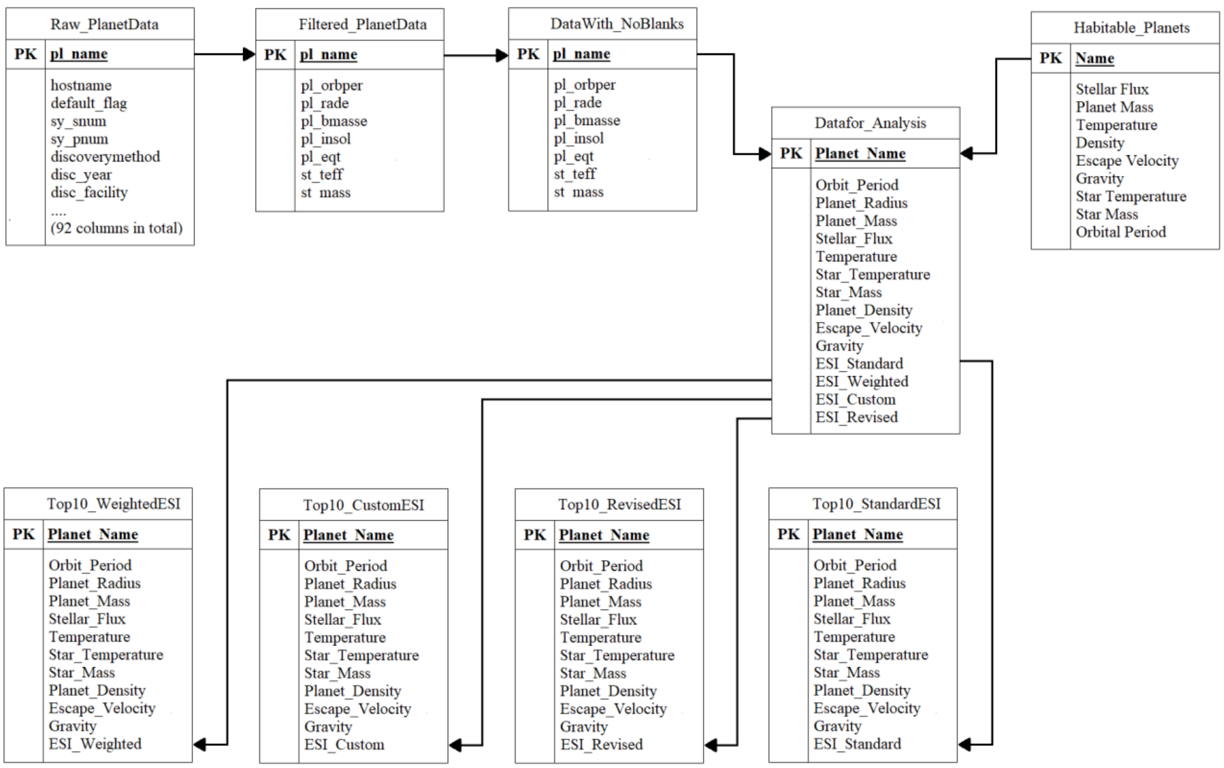
Each step in the construction of the ESI database can be visualized with a series of diagrams. The first of these is the Relational Diagram. The "Raw Planet Data" from the "PS_2022.11.09_09.55.33.csv" file serves as the source of all the data. Each subsequent table below the "Raw Planet Data" is subset of its predecessor above. The "Habitable Planets", which contains the supplemental data from the "Planet_Data.xlsx" has column names that differ from those of the "Data With No Blanks" table. It is for this reason that the relationship between these two tables is an identifying relationship.

Relational Diagram:



The next table below is the relational Table Layout. With the names of the columns visible, the difference between the "Data With No Blanks" and the "Habitable Planets" table becomes apparent. Another important thing to note is that the name of the planets serves as the primary key of every table in the database. The only exception being the "Raw Planet Data" and the "Filtered Planet Data" which have multiple entries of the same planet. But the planet name remains the identifier of the data.

Table Layout:



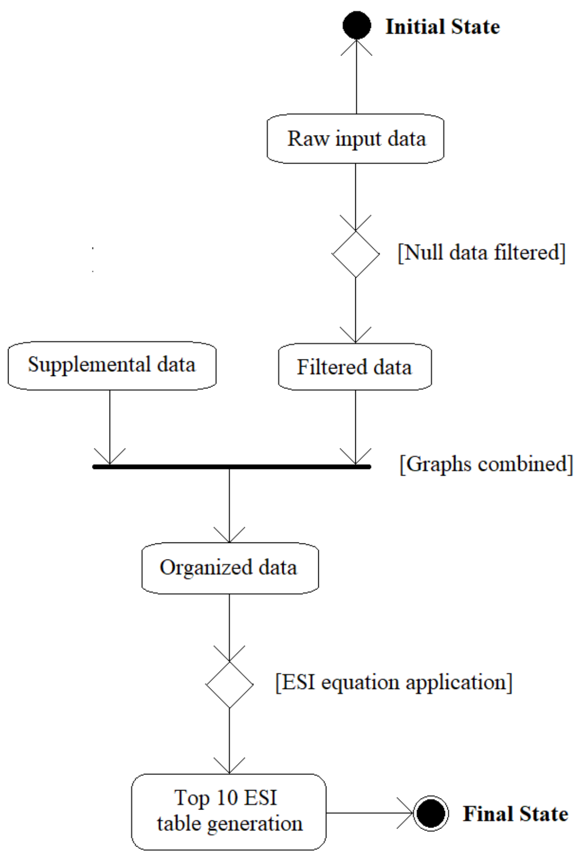
Most of the tables are subsets of one another. As a consequence, many share common variable names. In the case of the bottom 4 Top 10 ESI tables, each one of those inherits the ESI data that corresponds to their names.

Top 10 ESI Revised Table:

	Planet_Name	Orbit_Period	Planet_Radius	Planet_Mass	Stellar_Flux	Temperature	Star_Temperature	Star_Mass	Planet_Density	Escape_Velocity	Gravity	ESI_Revised
1	Earth	365.0	1.00	1.00	1.00	288	5778	1.000	5.509374	11.18747	9.82000	0.9999683
2	Kepler-452 b	384.8	1.63	3.31	1.11	295	5578	1.020	4.210827	15.94235	12.23388	0.8478286
3	Kepler-1638 b	259.3	1.87	4.16	1.39	312	5710	0.970	3.504861	16.68622	11.68212	0.7807578
4	Kepler-1544 b	168.8	1.78	3.81	0.84	276	4820	0.743	3.721922	16.36758	11.80855	0.7758191
5	Kepler-22 b	289.9	2.38	6.90	1.10	294	5594	0.960	2.819814	19.04882	11.96208	0.7490546
6	Kepler-442 b	112.3	1.35	2.36	0.70	263	4563	0.720	5.284610	14.79181	12.71616	0.7489752
7	Kepler-1653 b	140.3	2.17	5.34	1.04	291	4771	0.719	2.879150	17.54983	11.13610	0.7210008
8	Kepler-283 c	92.7	1.82	3.97	0.84	280	4351	0.690	3.628094	16.52310	11.76953	0.7189633
9	Kepler-1606 b	196.4	2.07	4.93	1.64	325	5402	0.921	3.062232	17.26515	11.29842	0.7155341
10	Kepler-443 b	177.7	2.35	6.04	0.89	279	4781	0.750	2.564104	17.93563	10.74021	0.7133981
11	Kepler-1701 b	169.1	2.22	5.56	1.42	314	5146	0.870	2.799743	17.70488	11.07848	0.7108680

Another aspect to consider when planning the construction of a database is the design of the website and the user interface. The Software Design diagram below summarizes the website design process. In the initial state, the “Raw input data” is entered into the R file which filters and combines the data through a series of table subset creations. The 4 top 10 tables serve as the Final state of the software.

Software Design Diagram:

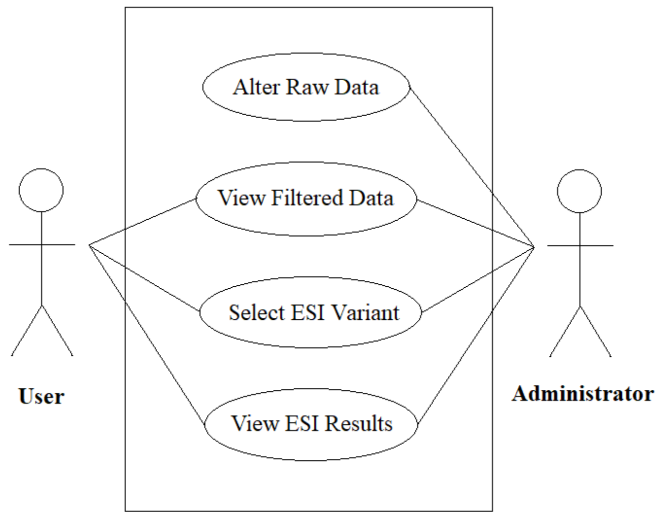


All of the tables listed above have been placed into a Gaben server as .csv files. The larger .csv files remain as .csv files due to the complexity of converting over 30,000 rows into SQL data tables. However, the small 12 row ESI variant tables are stored as SQL files in order to be inserted into the HTML interface.

CSS is used to create the options menu for the user to select the ESI variants. There were some plans to use PHP to generate graphs from the data. However, since the coding of the graphs requires the installation of a third-party library, the graphs have to be made in some alternative manner. Nevertheless, the website itself remains functional enough to allow users to compare each of the 4 ESI equations to one another.

The structure of the interface itself can be visualized by the Case Diagram on the next page. As one can see, the Administrator and User have the ability to view the filtered data, select the ESI variants, and view the ESI results. The one privilege that remains exclusive to the Administrator is the alteration of the data in the tables.

Case Diagram:



VI. Insertion into SQL

After I completed the ESI analysis using the R language, I recreated my initial analysis in the class Gaben server using SQL queries. Using the filtered "All_Data.csv" file I had copied into the server directory, I created a table titled "alldata" to contain all the usable exo-planet data. With the "Bulk Insert" command, I placed all the data from the .csv file into the "alldata" SQL table.

```
mysql> describe alldata;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| pname     | varchar(20) | NO   | PRI | NULL    |       |
| operiod   | double      | YES  |     | NULL    |       |
| radius    | double      | YES  |     | NULL    |       |
| pmass     | double      | YES  |     | NULL    |       |
| flux      | double      | YES  |     | NULL    |       |
| temp      | double      | YES  |     | NULL    |       |
| sttemp    | double      | YES  |     | NULL    |       |
| stmass    | double      | YES  |     | NULL    |       |
| density   | double      | YES  |     | NULL    |       |
| velocity  | double      | YES  |     | NULL    |       |
| gravity   | double      | YES  |     | NULL    |       |
| standard  | double      | YES  |     | NULL    |       |
| weighted  | double      | YES  |     | NULL    |       |
| custom    | double      | YES  |     | NULL    |       |
| revised   | double      | YES  |     | NULL    |       |
+-----------+-------------+------+--  Screenshot  -------+
15 rows in set (0.02 sec)
```

```
CREATE TABLE `alldata` (
  `pname` varchar(20) NOT NULL,
  `operiod` real DEFAULT NULL,
  `radius` real DEFAULT NULL,
  `pmass` real DEFAULT NULL,
  `flux` real DEFAULT NULL,
  `temp` real DEFAULT NULL,
  `sttemp` real DEFAULT NULL,
  `stmass` real DEFAULT NULL,
  `density` real DEFAULT NULL,
  `velocity` real DEFAULT NULL,
  `gravity` real DEFAULT NULL,
  `standard` real DEFAULT NULL,
  `weighted` real DEFAULT NULL,
  `custom` real DEFAULT NULL,
  `revised` real DEFAULT NULL,
  PRIMARY KEY (`pname`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;


BULK INSERT alldata
from 'All_Data.csv'
with (firstrow = 2,
      fieldterminator = ',',
      rowterminator='\n',
      batchsize=313,
      maxerrors=10);
```

After I created the initial table, I then proceeded to recreate the ESI equation columns. The <u>last 4 columns</u> of the “alldata” table: “standard”, “weighted”, “custom”, and “revised”, serve as the storage columns for the output of the ESI formulas. The SQL queries that I used for the insertion of the ESI planet scores share a number of similarities to the R code used in the initial analysis. The image below displays the SQL lines. The screenshot below that one displays the R syntax.

```
SELECT pname, (1-SQRT((1/2)*(POWER((flux-1)/(flux+1),2)+
              POWER((radius-1)/(radius+1),2))))
              as esistandard
FROM alldata
Group by pname;

SELECT pname, 1-SQRT((1/5)*(POWER((flux-1)/(flux+1),2)+
              (0.57*POWER((radius-1)/(radius+1),2))+
              (1.07*POWER((density-5.51)/(density+5.51),2))+
              (0.7*POWER((velocity-11.186)/(velocity+11.186),2))+
              (5.58*POWER((temp-288)/(temp+288),2))))
              as esiweighted
FROM alldata
Group by pname;

SELECT pname, 1-SQRT((1/9)*(POWER((flux-1)/(flux+1),2)+
              POWER((radius-1)/(radius+1),2)+
              POWER((gravity-9.82)/(gravity+9.82),2)+
              POWER((mass-1)/(mass+1),2)+
              POWER((temp-288)/(temp+288),2)+
              POWER((sttemp-5778)/(sttemp+5778),2)+
              POWER((stmass-1)/(stmass+1),2)+
              POWER((operiod-365)/(operiod+365),2)+
              POWER((density-5.51)/(density+5.51),2)))
              as esicustom
FROM alldata
Group by pname;

SELECT pname, 1-SQRT((1/9)*(3.2*(POWER((flux-1)/(flux+1),2))+
              (0.57*POWER((radius-1)/(radius+1),2))+
              (4.75*POWER((gravity-9.82)/(gravity+9.82),2))+
              (0.2*POWER((mass-1)/(mass+1),2))+
              (10.58*POWER((temp-288)/(temp+288),2))+
              POWER((sttemp-5778)/(sttemp+5778),2)+
              POWER((stmass-1)/(stmass+1),2)+
              POWER((operiod-365)/(operiod+365),2)+
              (2.8*POWER((density-5.51)/(density+5.51),2))))
              as esirevised
FROM alldata
Group by pname;
```

```
#ESI variants analysis
#ESI Standard
standard <- gravity %>% mutate(ESI_Standard=
(1-sqrt((1/2)*(((Stellar_Flux-1)/(Stellar_Flux+1))^2+
               ((Planet_Radius-1)/(Planet_Radius+1))^2))))
#ESI Weighted
weighted <- standard %>% mutate(ESI_Weighted=
(1-sqrt((1/5)*(((Stellar_Flux-1)/(Stellar_Flux+1))^2+
         (0.57*((Planet_Radius-1)/(Planet_Radius+1))^2)+
         (1.07*((Planet_Density-5.51)/(Planet_Density+5.51))^2)+
          (0.7*((Escape_Velocity-11.186)/(Escape_Velocity+11.186))^2)+
         (5.58*((Temperature-288)/(Temperature+288))^2)))))
#ESI Custom
custom <- weighted %>% mutate(ESI_Custom=
(1-sqrt((1/9)*(((Stellar_Flux-1)/(Stellar_Flux+1))^2+
               ((Planet_Radius-1)/(Planet_Radius+1))^2+
               ((Gravity-9.82)/(Gravity+9.82))^2+
               ((Planet_Mass-1)/(Planet_Mass+1))^2+
               ((Temperature-288)/(Temperature+288))^2+
               ((Star_Temperature-5778)/(Star_Temperature+5778))^2+
               ((Star_Mass-1)/(Star_Mass+1))^2+
               ((Orbit_Period-365)/(Orbit_Period+365))^2+
               ((Planet_Density-5.51)/(Planet_Density+5.51))^2))))
#ESI Revised
revised <- custom %>% mutate(ESI_Revised=
(1-sqrt((1/9)*((3.2*((Stellar_Flux-1)/(Stellar_Flux+1))^2)+
               (0.57*((Planet_Radius-1)/(Planet_Radius+1))^2)+
               (4.75*((Gravity-9.82)/(Gravity+9.82))^2)+
               (0.2*((Planet_Mass-1)/(Planet_Mass+1))^2)+
               (10.58*((Temperature-288)/(Temperature+288))^2)+
               ((Star_Temperature-5778)/(Star_Temperature+5778))^2+
               ((Star_Mass-1)/(Star_Mass+1))^2+
               ((Orbit_Period-365)/(Orbit_Period+365))^2+
               (2.8*((Planet_Density-5.51)/(Planet_Density+5.51))^2)))))
```

Each of the SQL and R equations represent the structure of the original ESI equations. In the case of the Standard ESI formula, the "flux" and "radius" column names serve as stand ins for the F and R variables of the original Standard ESI equation.

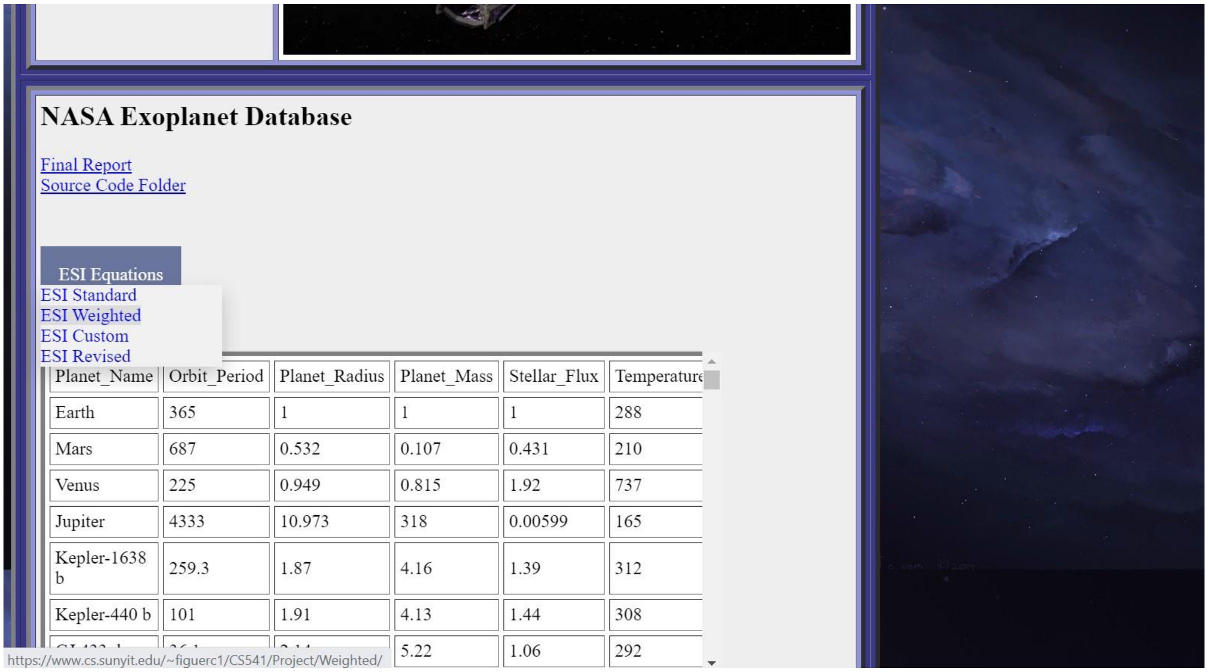
Standard ESI:

$$\mathrm{ESI}(F,R) = 1 - \sqrt{\frac{1}{2}\left[\left(\frac{F-Fe}{F+Fe}\right)^2 + \left(\frac{R-Re}{R+Re}\right)^2\right]}$$

```
SELECT pname, (1-SQRT((1/2)*(POWER((flux-1)/(flux+1),2)+
              POWER((radius-1)/(radius+1),2))))
              as esistandard
FROM alldata
Group by pname;
```

VII. The Interface

The final output of NASA's ESI database analysis can be viewed from the project website itself. Each of the 4 ESI equation variants can be selected from the menu which emerges from the button "ESI Equations". The table "Data for Analysis" displays all the filtered exo-planet data that does not contain any null data.



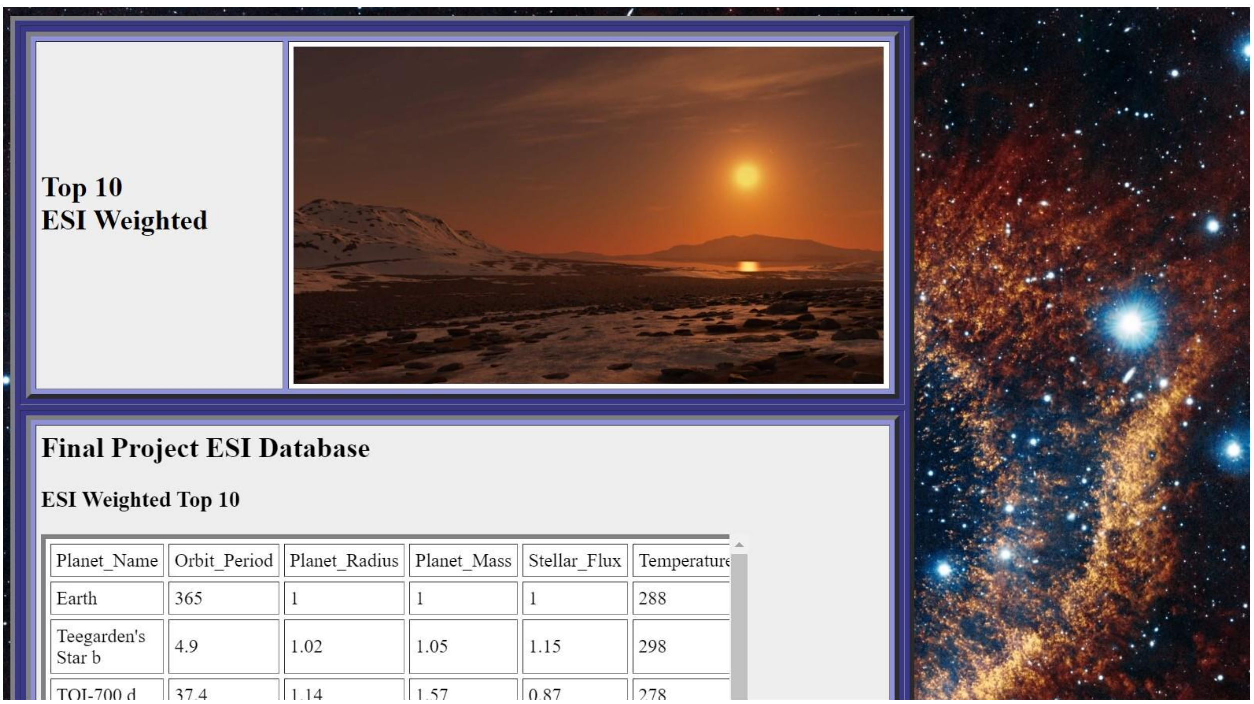
NASA Exoplanet Database

Final Report
Source Code Folder

ESI Equations
ESI Standard
ESI Weighted
ESI Custom
ESI Revised

Planet_Name	Orbit_Period	Planet_Radius	Planet_Mass	Stellar_Flux	Temperature
Earth	365	1	1	1	288
Mars	687	0.532	0.107	0.431	210
Venus	225	0.949	0.815	1.92	737
Jupiter	4333	10.973	318	0.00599	165
Kepler-1638 b	259.3	1.87	4.16	1.39	312
Kepler-440 b	101	1.91	4.13	1.44	308
[illegible]	[illegible]	[illegible]	5.22	1.06	292

https://www.cs.sunyit.edu/~figuerc1/CS541/Project/Weighted/



Top 10 ESI Weighted

Final Project ESI Database

ESI Weighted Top 10

Planet_Name	Orbit_Period	Planet_Radius	Planet_Mass	Stellar_Flux	Temperature
Earth	365	1	1	1	288
Teegarden's Star b	4.9	1.02	1.05	1.15	298
TOI-700 d	37.4	1.14	1.57	0.87	278

If you select the option "ESI Weighted", the website will redirect you a webpage titled "Top 10 ESI Weighted". The "ESI Weighted Top 10" table is the final output of the ESI Weighted equation. Earth serves as the basis for comparison to the other planets. The ESI values can be found by scrolling to the far left of the table.

```
<table ALIGN=LEFT BORDER=4 CELLSPACING=4 CELLPADDING=4 COLS=2 WIDTH="740" BGCOLOR="
<tr>
<td colspan=2 bgcolor=#EEEEEE>
<font color=black>

<H2>NASA Exoplanet Database</H2>
<a href="FinalReport.pdf">Final Report</a><br>
<a href="https://drive.google.com/drive/folders/1f08Qxob_xZy_Y4qbbz6hv8D9DvYytIka">

<H4> </H4>
<div class="dropdown">
  <buttom class="dropbtn">ESI Equations</button>
  <div class="dropdown-content">
    <a style="text-decoration:none;" href="Standard/">ESI Standard</a><br>
    <a style="text-decoration:none;" href="Weighted/">ESI Weighted</a><br>
    <a style="text-decoration:none;" href="Custom/">ESI Custom</a><br>
    <a style="text-decoration:none;" href="Revised/">ESI Revised</a><br>
  </div>
</div>
```

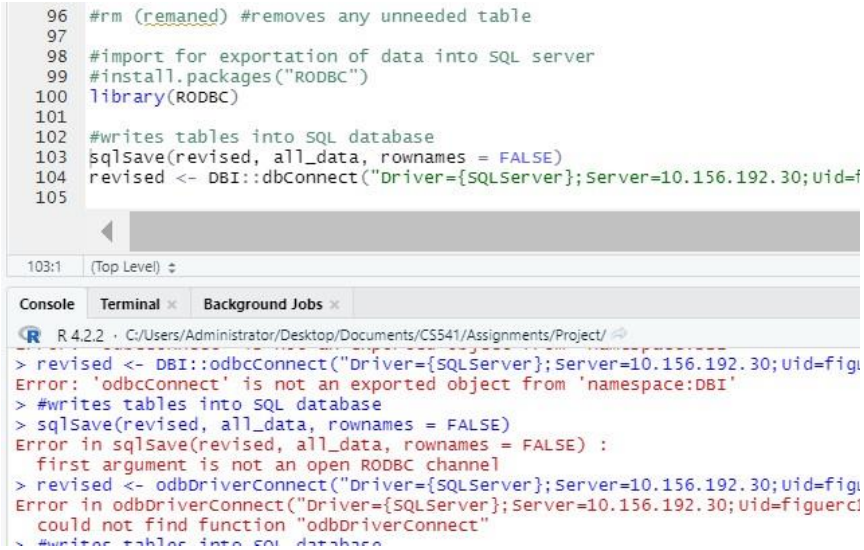
I created the “ESI Equations” menu with a combination of HTML links and CSS styles code. The user interface follows a simple selection of menu structure that serves as the bridge between the “Data for Analysis” and the 4 top 10 ESI tables. The CSS code itself is imbedded inside the PHP file which also contains all the HTML code.

```
<style>
  .dropbtn {
    background-color: #8F92DA;
    color: white;
    padding: 16px;
    border: none;
  }

  .dropdown {
    position: relative;
    display: inline-block;
  }

  .dropdown-content {
    display: none;
    position: absolute;
    background-color: #f1f1f1;
```

IIX. Challenges of Coding

Throughout the process of designing and building this database, I encountered a number of hurdles that necessitated a change to my original plan for the project. When I first became aware of the requirement for the use of large databases, I realized that I was going to need to utilize the R language to filter and create sub tables from the data. With the process of connecting directly to NASA’s exo-planet catalog being nigh impossible, I believed downloading the .csv file would be the simpler alternative.

The installation of the R language was a relatively simple process. There were no hurdles in the installation of packages or the finding of the necessary syntax for the coding itself. I did run onto a problem connecting my data from R studio directly to the Gaben server.



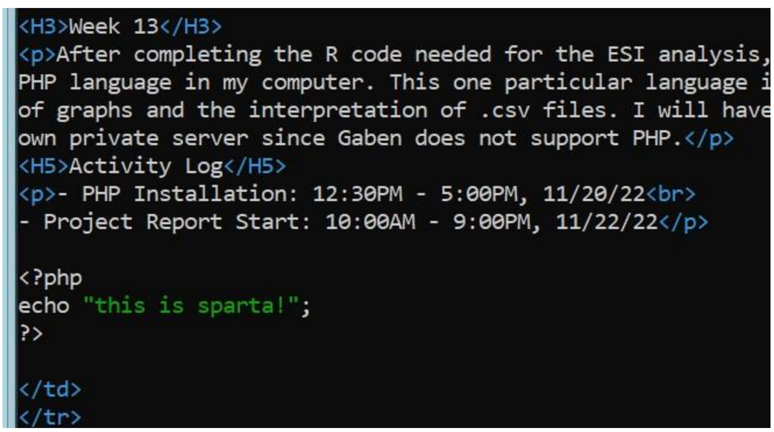
```
#rm (remaned) #removes any unneeded table

#import for exportation of data into SQL server
#install.packages("RODBC")
library(RODBC)

#writes tables into SQL database
sqlSave(revised, all_data, rownames = FALSE)
revised <- DBI::dbConnect("Driver={SQLServer};Server=10.156.192.30;Uid=f
```



```
> revised <- DBI::odbcConnect("Driver={SQLServer};Server=10.156.192.30;Uid=figu
Error: 'odbcConnect' is not an exported object from 'namespace:DBI'
> #writes tables into SQL database
> sqlSave(revised, all_data, rownames = FALSE)
Error in sqlSave(revised, all_data, rownames = FALSE) :
  first argument is not an open RODBC channel
> revised <- odbDriverConnect("Driver={SQLServer};Server=10.156.192.30;Uid=figu
Error in odbDriverConnect("Driver={SQLServer};Server=10.156.192.30;Uid=figuerc
  could not find function "odbDriverConnect"
```

I researched the proper syntax needed for the connection from R to a server. None of the possible syntax alternatives worked. The one conclusion that I gathered was there was a permissions setting in the server preventing the use of R code to access it. Since I am not the administrator of Gaben, I cannot correct this issue.

To work around this setback, I copied the .csv files into the CS541 project Gaben folder. From my research, I gathered that could use PHP to display the table data as graphs onto the HTML site. I first tested a sample PHP line “<?php echo “this is sparta!”; ?> on the log HTML file.



```
<H3>Week 13</H3>
<p>After completing the R code needed for the ESI analysis,
PHP language in my computer. This one particular language i
of graphs and the interpretation of .csv files. I will have
own private server since Gaben does not support PHP.</p>
<H5>Activity Log</H5>
<p>- PHP Installation: 12:30PM - 5:00PM, 11/20/22<br>
- Project Report Start: 10:00AM - 9:00PM, 11/22/22</p>

<?php
echo "this is sparta!";
?>

</td>
</tr>
```

Week 13

After completing the R code needed for the ESI analysis
installed the PHP language in my computer. This one pa
is neccesary for the use of graphs and the interpretation
have to setup the graphs on my own private server since
support PHP.

Activity Log

- PHP Installation: 12:30PM - 5:00PM, 11/20/22
- Project Report Start: 10:00AM - 9:00PM, 11/22/22

As you can see, the PHP echo statement “this is sparta!” did not display in the website. However, one crucial detail I did not take into account was the name of the webpage file. The “index.html” file had an .html extension. Once I renamed the file to “index.php”, that same test PHP code displayed on the project page.

```
:~/www/CS541/Project$ nano index.html
:~/www/CS541/Project$ ls
 index.html      Top_Revised.csv   Top_Weighted.csv
 Top_Custom.csv  Top_Standard.csv
:~/www/CS541/Project$ mv index.html index.php
:~/www/CS541/Project$ ls
 index.php       Top_Revised.csv   Top_Weighted.csv
 Top_Custom.csv  Top_Standard.csv
:~/www/CS541/Project$
```

```
<H2>Final Project ESI Database</H2>
<a href="FinalReport.pdf">Final Report</a><br>
<a href="https://drive.google.com/drive/folders/
<?php echo "This is sparta"; ?>
```

Final Project ESI Database

Final Report
Source Code Folder
This is sparta

The appearance of the test line “This is Sparta” confirms that the Gaben server supports PHP. With this knowledge, I was able to connect the .csv files directly to the .php webpage itself. The code below displays an example of a table generated from a .csv file. This code was sourced from Geeksforgeeks.org [10].

```
<H3>Data for Analysis</H3>

<div style="width:600;height:300px;overflow:auto;">
<?php
echo "<html><body><center><table>\n\n";
$file = fopen("All_Data.csv", "r");

while (($data = fgetcsv($file)) !== false)
{
  echo "<tr>";
  foreach ($data as $i)
  {
    echo "<td>" . htmlspecialchars($i)
    . "</td>";
  }
  echo "</tr> \n";
}

fclose($file);
echo "\n</table></center></body></html>";
?>
</div>
```

Data for Analysis

Planet_Name	Orbit_Period	Planet_Radius	Planet_Mass	Stellar_Flux	Temperature	Star_Te
Earth	365	1	1	1	288	5778
Mars	687	0.532	0.107	0.431	210	5778
Venus	225	0.949	0.815	1.92	737	5778
Jupiter	4333	10.973	318	0.00599	165	5778
Kepler-1638 b	259.3	1.87	4.16	1.39	312	5710
Kepler-440 b	101	1.91	4.13	1.44	308	3813
GJ 433 d	36.1	2.14	5.22	1.06	292	3600
Kepler-1653 b	140.3	2.17	5.34	1.04	291	4771
GJ 832 c	35.7	2.19	5.4	0.99	286	3657
Kepler-705 b	56.1	2.11	5.09	0.77	269	3593

The Gaben server's usability of PHP code facilitated the visualization of the R tables in the website. I had some initial project plans that involved a direct link between the SQL database and the user interface. However, the possibility of data corruption made me reconsider SQL as the choice for the primary data storage system.

One example of data corruption happened in Homework 5 with the "Option" table. I created the blank option table as usual. When I made the attempt to input the data into the table, I was greeted by an error message. I made sure to type the name of the table correctly. I could input data into every other table. But for some unknown reason, the system did not recognize the Option table.

```
+-------------------+
| book              |
| book_authors      |
| book_copies       |
| book_loans        |
| borrower          |
| car               |
| course            |
| department        |
| dependent         |
| dept_locations    |
| employee          |
| grade_report      |
| library_branch    |
| option            |
| option2           |
```

```
| price       | varchar(10) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql> describe option;
ERROR 1064 (42000): You have an error in your SQL syntax; chec
 for the right syntax to use near 'option' at line 1
mysql>
```

As noted in the images above, the “describe” command does not work on a corrupted table. Under normal circumstances, the “describe” command should print out the contents of the table. As a consequence, I created a new table “Option2” to store the data I needed.

```
mysql> describe option2
    -> ;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| serial_no   | varchar(25) | NO   | PRI |         |       |
| option_name | varchar(20) | NO   | PRI |         |       |
| price       | varchar(10) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

The thought of losing any of the project’s data was too much to bear. Although this project could be done entirely in the SQL language, because of the possibility of data corruption, I opted to perform the initial analysis in R before doing the SQL analysis. That way, I could have an offline .csv backup of every table I would create using SQL.

IX. Conclusion

The creation of the ESI database began with an idea, the idea to create a more reliable way to find habitable Earth-like planets outside the Sol system. The initial research of the ESI equation led me to uncover new factors that affect the habitability of planets. With the discovery of these factors, I have derived the Revised ESI formula, which was a bi-product of the other three ESI variants.

NASA's original database was comprised of roughly 33,000 rows. The implementation of the R language has allowed me to narrow down the data to a more manageable size of 313. I then utilized SQL to create 4 sub tables as a means to compare the results of the ESI formulas. These tables are readily available in my Final Project site. I may have endured a number of setbacks in this project. But I believe the work I have come up with should reflect the original purpose of this database.

References

[1] https://exoplanets.nasa.gov/discovery/exoplanet-catalog/

[2] https://phl.upr.edu/projects/habitable-exoplanets-catalog

[3] https://phl.upr.edu/projects/earth-similarity-index-esi

[4] https://sciencing.com/measure-density-planet-5557152.html

[5] https://www.omnicalculator.com/physics/escape-velocity

[6] https://planetcalc.com/1758/

[7] https://interestingengineering.com/science/earth-similarity-index-where-could-we-live-besides-earth

[8] http://www.imatheq.com/imatheq/com/imatheq/math-equation-editor.html

[9] https://www.upi.com/Science_News/2017/02/08/NASA-Red-dwarf-habitable-zones-may-not-be-so-habitable/2881486579322/

[10] https://www.geeksforgeeks.org/how-to-display-data-from-csv-file-using-php/

[11] https://www.w3schools.com/howto/howto_css_dropdown.asp

[12] https://popsql.com/learn-sql/sql-server/how-to-import-a-csv-in-sql-server

[13] https://stackoverflow.com/questions/7367750/average-of-multiple-columns