

Revitalizing Solar Insights: A Dashboard for West Tennessee Solar Farm

Joshua Chamberlain
Martin, TN
jospcham@ut.utm.edu

Andy Lum
Lakeland, TN
andlum@ut.utm.edu

ABSTRACT

This project we constructed aims to be an interactive dashboard for displaying solar irradiance data collected at a photovoltaic power station. Given a recent push by the University of Tennessee Research Foundation toward revitalizing its use, the West Tennessee Solar Farm will serve as a template. This location is of particular interest due to its proximity to Blue Oval City (the site of the new Ford manufacturing plant, near Stanton, TN). With the farm's existing dashboard being non-functional, there is a demand for a solution, which we will achieve through MySQL, Python, Google Drive API, R-Shiny, Shinyapps.io, and Google Cloud Console. MySQL serves as our data hub, efficiently organizing solar energy data by sensor location. Python, coupled with the Google Drive API, simulates real-time data collection.

The core of the project is an R-Shiny dashboard offering real-time data visualization, interactive maps, detailed sensor information, and access to historical data and analysis. Users can select their desired time frames. Shinyapps.io hosts the dashboard, ensuring accessibility across diverse platforms, such as web browsers and various operating systems. This approach allows users from all major operating systems to access the dashboard, promoting widespread accessibility. To further fortify data security and enhance user convenience, Google Cloud Console safeguards our API information. Our dashboard incorporates an export function, enabling users to extract data. In addition, we constructed an easy-to-use webpage that is accessible across various major operating systems. This approach ensures that our project is widely available and caters to a diverse audience; thus, making valuable solar irradiance data easily accessible to all. This project aims to provide researchers, policymakers, and the public with real-time insights into solar irradiance data at the West Tennessee Solar Farm, supporting sustainable energy solutions.

1. INTRODUCTION

In the pursuit of sustainable energy solutions, our project ventures to unlock the vast potential of solar irradiance data. Our focus is on the West Tennessee Solar Farm, situated near Stanton, TN [7]. Its strategic proximity to Blue Oval City adds notable importance. The critical challenge we address is the current, non-functional existing dashboard for the solar farm, which has left vital solar irradiance data inaccessible for potential users. The West Tennessee Solar Farm offers the potential for abundant clean energy, but its potential remains unrealized in part due to the limitations of the existing dashboard. In this project, we aspire to bridge this data accessibility gap while opening up opportunities to inform and educate the general public about the farm's renewable energy potential.

In response to this need, our project emerges as a symbol of innovation and accessibility, seamlessly integrating an array of cutting-edge technologies. With MySQL as our data hub, we thoroughly organize solar energy data by sensor location.

Python, in collaboration with the Google Drive API, takes on the role of capturing and decoding real-time data with unwavering precision. However, the importance of our project lies in R-Shiny dashboard, an interactive interface that goes beyond the typical data visualization. It not only brings solar irradiance data to life in real-time but also empowers users with interactive maps, intricate sensor insights, information that is relatable to the lay-person, and a comprehensive historical archive.

2. MOTIVATION

The West Tennessee Solar Farm, located near Stanton, TN, represents a beacon of hope in the quest for clean and sustainable energy production. Nevertheless, despite its significant potential, the realization of this promise has remained elusive due to the operational constraints of the existing dashboard. Unfortunately, the dashboard has remained non-functional, thereby preventing users from accessing vital solar irradiance data. As a result, the solar farm's ability to significantly contribute to the renewable energy sector has been hampered.

This challenge serves as the stimulus for our proactive engagement with this project. Our project was further motivated by the valuable input we received from the CEO and President of the West Tennessee Solar Farm during a conference with Dr. Sims. When we presented our initial dashboard, they were highly impressed with the concept and functionality. Their constructive feedback propelled us to further refine and develop the final dashboard product that we proudly offer today.

Our interactive dashboard not only addresses the immediate needs of the solar farm but also extends its utility to education and research purposes. With this tool, we enable students, researchers, and the general public to gain insightful access to real-time solar irradiance data. This not only empowers them to better understand and appreciate the significance of renewable energy but also fosters a collaborative environment where valuable insights can be shared and applied. Whether for educational institutions or research facilities, our interactive dashboard serves as an invaluable resource for learning, innovation, and informed decision-making in the pursuit of sustainable energy solutions.

Unfortunately, the dashboard was not able to put onto the website due to legal constraints. The proof of concept for the dashboard will provide a template for other solar farms to be able to connect the citizens they power to the critical information needed. Furthermore, it educates the public on how the solar power process works. The educational and research benefits still provide multiple benefits to the community surround any solar farm.

3. KEY TERMS

As the project deals with the solar plant process, this report will contain many terms pertaining to the solar power process

of collecting energy and sending that power to the electrical grid. Some key terms you may need to know include:

1. Irradiance

- The energy per meter squared received from the sun in the form of electromagnetic radiation.

2. Solar Farm Efficiency Factor

- A measure of how effectively a solar farm converts received sunlight into usable electrical energy.

3. Angle of Incidence

- The angle at which sunlight strikes a surface, affecting the efficiency of solar energy absorption.

4. GGPlot

- A Grammar for Graphics Plot (GGPlot) is a plot that allows for customization of plots through the Grammar for Graphics principles.

4. TECHNOLOGIES USED

4.1 MySQL

The MySQL database was used to store information about irradiance collected from data sensors. The database schema included a main table called Irradiance that contained the following information:

1. The day of year the data was recorded in, ranging from 1 - 365 representing each day of the year.
2. The minute of the day that the data was recorded in, ranging from 1 - 1440 representing each minute of the day.
3. 10 separate columns containing the collected irradiance from each sensor in watts per meter squared ($\frac{W}{m^2}$).
4. The total irradiance of all sensors.

The reason the project used MySQL was because it is a lightweight database that is free. Furthermore, the user interface of mySQL workbench allowed for quick and easy queries and data visualization of the data table.

4.2 Python

The Python component of the data pipeline plays a pivotal role in facilitating the seamless flow of real-time information from the MySQL database to the R-Shiny dashboard. Utilizing the MySQL library[4], the Python program establishes a connection with the solar farm's database and initiates a systematic retrieval of data at regular one-minute intervals. This periodic extraction aligns with the continuous data collection carried out by the sensors.

The Python script's functionality extends to the formulation and execution of optimized queries, precisely tailored to fetch relevant data from the MySQL database. Subsequently, the retrieved data undergoes a meticulous transformation process, culminating in its storage within a CSV file hosted on Google Drive. This strategically positioned CSV file serves as a dynamic conduit, ensuring a consistent inflow of updated information to the R-Shiny dashboard. In essence, the Python program functions as a sophisticated data intermediary, bridging the gap between the solar farm's database and the real-time visualization presented on the dashboard.

4.3 R-Shiny

Employing a diverse array of packages, R-Shiny assumes a crucial role in executing intricate statistical analyses and generating visual representations of power production and irradiance within the solar farm data project. Within this framework, the code is intricately woven to design a user interface and seamlessly facilitate the conversion processes between irradiance and power production. The various R-Shiny packages not only facilitate statistical computations but also empower the system to dynamically plot and illustrate the nuanced relationship between power production and irradiance. This strategic integration of R-Shiny underscores its versatility and indispensability in the comprehensive functionality of the solar farm data dashboard, aligning cohesively with the technical intricacies of our project.

4.4 ShinyApps.io

ShinyApps.io serves as a pivotal platform for hosting our innovative solar farm data dashboard in the cloud. As a cloud-based service, ShinyApps.io enables us to make the dashboard publicly accessible, aligning with our project's goal to not only serve as a research tool but also as an educational resource. Operating on the Posit Cloud, which is our chosen primary server for R-Shiny programs, ShinyApps.io facilitates the deployment of our live dashboard with efficiency and reliability. This decision reflects our commitment to providing a seamless and widespread user experience, essential for both educational outreach and scientific exploration.

In operation, ShinyApps.io functions as a cloud-hosting service that allows R-Shiny applications, like our solar farm dashboard, to be deployed online. The Posit Cloud, integrated as the primary server, ensures the continuous availability and scalability of our dashboard.

ShinyApps.io works by allowing users to access and interact with the solar farm data remotely through a web interface. This cloud-based approach enables all the necessary dashboard features as well as making it available to as many users as possible.

4.5 Google Cloud Console

The utilization of the Google Cloud Console plays a pivotal role in fortifying the security measures implemented for the data transmission to the website. This platform functions as a robust shield, safeguarding sensitive information such as passwords and API keys during the interaction with both the database and the utilized APIs. Through meticulous encryption and secure protocols, the console ensures that these crucial authentication elements remain obfuscated, minimizing the risk of unauthorized access and potential vulnerabilities. By meticulously concealing the authentication credentials, the Google Cloud Console contributes significantly to the overall data integrity and confidentiality, aligning seamlessly with the stringent security standards requisite in the realm of computer science. This deliberate approach towards safeguarding the data transfer process underscores the commitment to maintaining a secure and resilient infrastructure for the seamless operation of the solar farm data dashboard.

5. R PACKAGES

R packages play a pivotal role in bringing the solar farm data dashboard to life. R packages are collections of functions, documentation, and sample data bundled together for specific tasks or purposes. In the project, several key R packages have

been harnessed to create a robust and interactive dashboard, catering to the real-time needs of the solar farm data:

5.1 Shiny

Shiny is an R package used to bridge the gap between R and HTML. It was used to design the UI of the dashboard as well as connecting the data collected from the MySQL database to the R program. Finally, it allowed reactivity in the R program, making real time data visualization possible.[8][10]

5.2 Leaflet

Leaflet is used to provide the map and all of its functionalities. It uses the satellite photos from the USGS (United States Geological Survey) for the interactive picture of the solar farm. Leaflet also contributed to the popups and zoom features of the map, allowing the user to have easy access to information from each sensor.[3]

5.3 OpenMeteo

OpenMeteo is an API used for weather forecasting and historical data analysis. It allowed for forecasting the next day on the dashboard. It provided:

- Hourly Cloud Cover (%)
- Hourly Temperature (°F)
- Hourly Weather Codes
- Sum of Daily Shortwave Radiation ($\frac{W}{m^2}$)

The hourly variables were averaged for user convenience.[5]

5.4 Google Drive

The google drive package allows the program to connect to google drive files. It is used to connect to the csv file that contains all the processed data for the dashboard plots and sensor information.[2]

5.5 Rconnect

The rconnect package connects the R-shiny application to shinyapps.io server. [6]

5.6 GGPlot2 and Plotly

GGPlot2 and Plotly allows for the interactivity in the plots. It allows the user to do a variety of functionalities, live with the data. It fades out to refresh as well giving a visual indication when it is updating.

5.6.1 Plot Functionalities

1. Interactivity

The plots are able to detect the position of the mouse on the plot and determine what MINUTE value and the current irradiance at that value it is currently. It has a labeled text box follow the user's mouse in the following format: MINUTE: User's minute Irradiance: User's irradiance.

2. Picture a Certain Moment

There is a button in the top right of the plot that looks like a camera. This allows the user to take a picture of the plot and saves the picture as an PNG to the user's local computer.

3. Null Data Values

The plot will take null data values. This is important because solar panels have a variety of reasons to be down. For example, a solar panel could be down for maintenance. The plot should not show 0 because it is not collecting, thus

the plot is very helpful. Finally, this also provides a quick visual representation for the user.

GGPlot works by taking the base plot of R and adding layers onto the graph. Adding layers to the graph allow for more freedom in customization.

6. PROJECT GOALS

Using the technologies mentioned above and their functions, we want to implement:

6.1 Streamlined Data Pipeline

The optimal data pipeline would be: MySQL → Python → Google Drive CSV → Dashboard. Optimizing the data pipeline allows for real time reactive data. The MySQL database would store the data. Then the python script would then execute queries in order to retrieve the data and update the Google Drive CSV. The Google Drive CSV would hold the data in a format readable by the R-shiny dashboard. Finally, the dashboard reads the data and reacts accordingly.

6.2 Continuous Data Maintenance

A key objective of the project involves the ongoing monitoring of data, incorporating a visual mechanism to flag instances of "bad" data. An illustrative scenario is the need for solar panel sensors to undergo maintenance, during which the collected data should register as NULL rather than zero. This meticulous approach to data maintenance extends to the graphical representations, ensuring that both plots and maps accurately reflect periods of sensor downtime due to maintenance activities. By implementing this nuanced strategy, the project aims to sustain a high level of data integrity, providing a comprehensive and accurate depiction of the solar farm's operational status.

6.3 R-Shiny Dashboard

The dashboard will be user friendly and be able to be used for educational purposes. The features mentioned later in the paper will also be fully functional and reactive.

6.4 Cross-Platform Accessibility

The dashboard is meant to be used as an educational tool for students and interested parties. In order to help provide this information to as many people as possible, the dashboard is planned to be accessible on all popular operating systems. In particular, the dashboard will be available to Linux, MacOS, and Windows users.

7. INITIAL MOCKUP

Title of Webpage (West Tennessee Solar Farm Dashboard)			
Map With Sensor Markers and Pop-ups		Daily Panel Button	Weekly Panel Button
		Monthly Panel Button	
		Plot of Irradiance Values over time of current sensor selected (Hourly, Daily, and Weekly)	
		Plot of Power Consumption over Time of current sensor selected (Hourly, Daily, and Weekly)	
Date Selector - A drop down calendar for users to select timelines-Daily, Weekly, or Monthly-for data plotting, synchronized with their choice		Data Export Button	

7.1 Initial Features

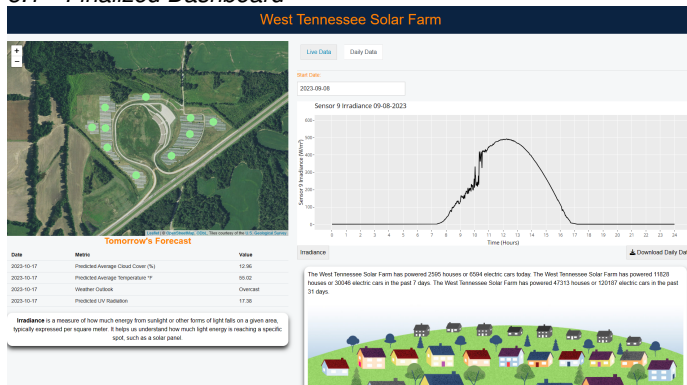
The initial features of the first mockup includes:

- A title for the webpage
- A map with sensor markers and Popups that display the current irradiance
- An irradiance plot that shows data in a selected time frame reactively
- A power production plot that takes the irradiance and converts it into power production reactively.
- A data export button that allows the user to download data for a specified time frame
- A date selector that allows the user to specify the time of the data shown
- A button that allows the user to switch between plotting daily, weekly, and monthly data plots.

8. FINALIZED DASHBOARD FEATURES

Throughout the design process many features have either been added or removed.

8.1 Finalized Dashboard



8.2 Reactive Plots

The dashboard's main feature is the reactive plots that show the current irradiance level or power production. The plots are able to change from live plot to daily plots. Also within the daily plots, the plots can switch between irradiance and power production. It does this using a formula gathered through scientific study.[9]

This works by making two vectors, the MINUTE and the current irradiance or power production. Then it takes the vectors and plots it using MINUTE as the x-axis and the irradiance or power production as the y-axis. It takes the scatter plot and connects the data values to create a line using the geom_line function. Next, it takes the adjusts the x and y axes and adjust them to the specified parameters. Finally, it converts the ggplot to a plotly so it can be properly displayed on the dashboard.

8.3 Interactive Map

The map that shows the West Tennessee Solar Farm[7] from an ariel view. It has clickable sensors that pop-up with the current irradiance value at that time. The sensors change colors when one is abnormally low or abnormally high. It is colored based on a red to green gradient.⁷ The map also has the capability to zoom in and out. All of the interactivity and reactivity within the map was made using the Leaflet package.[3]

The leaflet map works similar to the reactive plots. It is based on the base map provided by USGS. It adds layers onto it based on the parameters set. The two layers added onto the map were the markers and the popup.

8.4 Weather Forecast Table

The dashboard has a weather forecast for the next day. It gives the cloud cover, temperature, weather code, and total predicted shortwave ultraviolet radiation. This allows not only users to know the predicted forecast, it also allows the West Tennessee Solar Farm to plan storage based on how much power the farm will have to supplement the main power supply.

OpenMeteo works by taking the parameters given in its program, such as location, and cloud cover. It then takes the parameters and retrieves the data from its database. Next it takes the data and puts it into a readable JSON (JavaScript Object Notation) file. Finally, R-shiny takes the JSON file and uses the tableOutput function to create a readable table on the dashboard.

Although, OpenMeteo provides much more information, to keep the information concise for the readability to the user the decision was made to only provide the most prominent factors affect the total irradiance and power production.

8.5 Exporting Daily Data

The dashboard has a button to download the specific days data the user is looking at as a csv. This is achieved using the downloadHandler in Shiny.[8]

Pressing the button will pop up with a "Save As" screen on the users local computer. By default it will name the file in the format, mm-dd-yyyy.csv, where the date is the current days plot the user is looking at.

8.6 Educational Features

The dashboard has two educational sections. The first section is a description of what irradiance is. The second section contains reactive text that converts the total power production to more relatable units: amount of houses powered and how many electric cars are being powered. This is calculated for the day, week, and month.

The two educational section provides not only an educational benefit, but it also eases the transition of newcomers to understand the information being presented to them.

9. OTHER FEATURES

Along with the dashboard there are two other tabs.

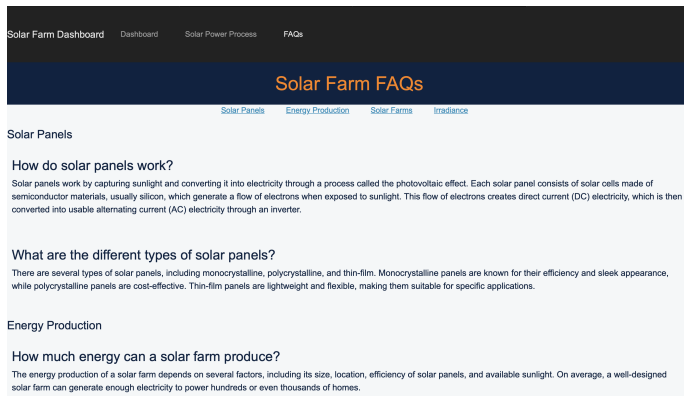
9.1 Solar Power Process

The tab explains in a simple format how the solar panels collect sunlight and convert it to power. It uses the fade in CSS animation to keep users engaged. It is meant to serve as an education and research tool for the public outreach and awareness. Using the UTM colors alternately, the tab looks like this:



9.2 Frequently Asked Questions

The tab answers Frequently Asked Questions about solar farms and the solar power process. The tab also has clickable buttons that using CSS automatically scrolls down to the appropriate category of questions. The tab looks like this:



10. POWER CONVERSION FORMULA

The data presented on the R-Shiny dashboard undergoes a crucial transformation to convert irradiance measurements, expressed in $\frac{W}{m^2}$, into a more meaningful metric: total power production, measured in kWh . This conversion is intricate, considering various environmental factors that influence solar energy generation. The formula employed for this conversion, as derived from a comprehensive study published in the *Renewable and Sustainable Energy Reviews* in 2022[9], intricately accounts for the nuanced interplay of factors such as the angle of incidence, the orientation of the solar panels, potential interference, and the solar farm's efficiency factor.

The empirical formula obtained from the study is articulated as follows: Power Production $\approx -0.038 + 0.0058 \times$ Current Irradiance. The nuanced nature of this formula necessitates a comprehensive adjustment to ensure that resulting power production values do not fall below zero, particularly for lower irradiance values. To address this, a safeguard mechanism is implemented in the form of the following R code: `max(0, -0.038 + 0.0058 * Current Irradiance)`. This ensures that the converted power production values remain non-negative, reflecting a realistic and feasible representation of solar energy output, accounting for the intricacies of panel orientation, solar radiation angles, and the inherent efficiency of the solar farm.

11. CHALLENGES

As our first experience with R-shiny was the dashboard, there were many challenges while implementing features and dealing with the interactivity between the MySQL database, the python script, the Google Drive CSV, and the R-shiny dashboard.

11.1 Streamlined Data Pipeline

As stated in the project goals, the data pipeline had to be optimized in order to have the most accurate reactive data on the dashboard. This was optimized through the python module `mysql`[4]. Using the module, after connecting to the MySQL database, the program executes a query to get the next minute of data after a specified amount of time.

11.2 Error Handling

As stated in the project goals, error handling provides a visual cue to the user for abnormal data or abnormal circumstances. Although making the visual cue a singular color would have been easier, the plan was to not only have a visual cue, but to further the user experience by having the color represent the intensity of the circumstance. The eventual solution was to have a color gradient to representing the intensity of the problem.

All the intensities are based on the differences between each of the sensors. The sensors will a muted green, showing less color to the user representing a lower irradiance value.

The reason for having the color based on the differences between each of the sensors is because during the night, all sensors produce little energy. If the intensity was based on the current irradiance value, it would create a false negative for all sensors while the sun is not visible.

11.3 Webpage Development

The dashboard is meant to be an educational tool, so the dashboard needed to be as accessible as possible. Typically, there are three different options:

11.3.1 ShinyApps.io

ShinyApps.io is a web platform leveraging cloud technology for the efficient publication of Shiny apps. This service streamlines the deployment process, providing users with a publicly accessible link for their applications. Notably, *ShinyApps.io* integrates seamlessly with R-Studio, enhancing the development and deployment workflow within the R programming environment. Furthermore, the platform offers a free tier, allowing users to host their apps for up to 25 hours per month. This resource allocation makes *ShinyApps.io* a practical and cost-effective solution.[1]

11.3.2 Shiny Server

A shiny server is a web server for the explicit use of shiny applications. Although this allows for more customization in how the app the published, it also requires having an enterprise backend database, and it is not free as well.[1]

11.3.3 Shiny Proxy

A shiny proxy server is used for enterprise applications. It takes a user accessing the shiny application and creates a separate session for each user. Although this is the best option in terms of scalability, it requires a lot of RAM as well as requiring lots of money to keep the server running.[1]

After carefully considering the pros and cons of each option, it was ultimately decided that the project was to be deployed

using shinyapps.io. The decision was made due to the ease of setting up as well as meeting our financial and accessibility needs.

12. FUTURE WORK

While the current dashboard boasts extensive functionality, there exist opportunities for enhancements and additions in the future.

1. Predictive Analysis

Implementing machine learning algorithms to predict power production for the upcoming day could significantly benefit the project. This predictive capability would enable a more refined optimization of the energy balance between supplying the West Tennessee Solar Farm and contributing excess power to the main grid. Maximizing supplemental power has broader implications, potentially influencing economic factors by affecting power prices based on supply and demand dynamics.

2. Push Notifications

Given the substantial impact of the West Tennessee Solar Farm on local communities, timely notifications are crucial. Introducing push notifications for scenarios where data is null or exhibits unusually low values (deviating beyond two standard deviations from the norm) would allow affected parties to prepare promptly. For instance, the Ford factory employees in Blue Oval City, relying on the solar farm's power supplementation, would benefit from immediate awareness in case of critical information.

3. On-boarding Video

Recognizing the dual role of the dashboard as both an educational and research tool, ensuring user comprehension is essential. Many users may lack familiarity with solar farm operations, posing a potential barrier to understanding the dashboard's features. Incorporating an on-boarding video into the interface would provide newcomers with a guided introduction, facilitating a better understanding of all available features.

13. CONCLUSION

In conclusion, our project represents more than just a solution to an immediate challenge. It embodies a valuable educational resource, catering to individuals keen on delving into the intricacies of solar farm operations and the utilization of renewable energy sources. With our interactive dashboard, learners can gain a comprehensive understanding of the solar farm process, from real-time data access to interactive map exploration, and even insights tailored to the layperson.

Additionally, our work extends its reach to benefit researchers by providing them with unfettered access to public data essential for their research. This research-friendly feature allows academics and scientists to harness the potential of solar irradiance data, thereby contributing to the field's growth and development.

By presenting our innovative and accessible solution, we've not just stated our intentions but have demonstrated a tangible contribution to the realm of renewable energy. Readers will discover a tool that bridges the gap between renewable energy potential and reality, making clean energy more accessible, understandable, and usable. Our project aims to be a valuable resource for those looking to learn and research in this area, offering a unique blend of functionality that brings the West

Tennessee Solar Farm and its data to life for the benefit of education and research alike.

14. ACKNOWLEDGMENTS

We would like to thank all of our partners for helping us make this project possible. Thank you to:

- Dr. Sims
- West Tennessee Solar Farm
- University of Tennessee Research Foundation
- University of Tennessee at Martin Department of Computer Science and
- University of Tennessee at Martin Department of Mathematics and Statistics

REFERENCES

1. How to deploy a stable shiny app with minimal effort? <https://medium.com/@akpradhn/how-to-deploy-a-shiny-app-at-zero-cost-94e8897df34d>, 2019.
2. Google Drive Package in R. <https://googledrive.tidyverse.org/>, 2023.
3. Leaflet for R - Introduction. <https://rstudio.github.io/leaflet/>, 2023.
4. MySQL for Python. <https://dev.mysql.com/doc/connector-python/en/connector-python-coding.html>, 2023.
5. Openmeteo API. <https://open-meteo.com>, 2023.
6. Rsconnect Package in R. <https://rstudio.github.io/rsconnect/>, 2023.
7. West Tennessee Solar Farm. <https://solarfarm.tennessee.edu/about/>, 2023.
8. Winston Chang. R-Shiny Documentation. <https://www.rdocumentation.org/packages/shiny/versions/1.7.5.1>, 2023.
9. Wenting Wang, Dazhi Yang, Nantian Huang, Chao Lyu, Gang Zhang, and Xueying Han. Irradiance-to-power conversion based on physical model chain: An application on the optimal configuration of multi-energy microgrid in cold climate. *Renewable and Sustainable Energy Reviews*, 161:112356, 06 2022.
10. Hadley Wickham. *Mastering Shiny: Interactive Apps, Reports, and Dashboards Powered by R*. O'Reilly, 2021.