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 [6]. 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. TECHNOLOGIES USED

2.1 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.

2.2 Python

The python program uses the mysql library[3] to connect to the database and retrieve data from the database. It gets the information at a rate of 1 minute, providing real time updates to the R-Shiny dashboard. It does this by creating a query that is then executed on the database.

2.3 R-Shiny

Through multiple packages, R-Shiny is used to perform statistical analyses and to plot both power production and irradiance. It also holds the code for the user interface and performs the conversion between irradiance and power production.

2.4 ShinyApps.io

ShinyApps.io is used to host the dashboard as a website on the cloud. It gives access to the dashboard for the public. This is essential since one of the main goals of the project is to make it an educational tool as well as a research tool. It uses the Posit Cloud, which is the main cloud server for R-Shiny programs.

2.5 Google Cloud Console

Google Cloud Console is used to secure the data being transferred to the website. It protects it by masking passwords and API keys to the database and the APIs being used.

3. 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:

3.1 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.[7][9]

3.2 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.[2]

3.3 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 the hourly cloud cover, hourly temperature, weather code, and the amount of shortwave UV radiation. The hourly variables were averaged for user convenience.[4]

3.4 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.[1]

3.5 Rsconnect

The rsconnect package connects the R-shiny application to shinyapps.io server. [5]

3.6 GGPlot2 and Plotly

GGPlot2 and Plotly allows for the interactivity in the plots. It allows the user to hover over the plot and see the minute the data was gathered and the irradiance at that time. It also allows null data to be absent from the dataset.

4. DASHBOARD FEATURES

4.1 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.[8]

4.2 Interactive Map

The map that shows the West Tennessee Solar Farm[6] 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. 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.[2]

4.3 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.

4.4 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.[7] 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.

4.5 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 calculated for the day, week, and month.

5. OTHER FEATURES

Along with the dashboard there are two other tabs.

5.1 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.

5.2 Frequently Asked Questions

The tab answers Frequently Asked Questions about solar farms and the solar power process.

6. FIGURES/CAPTIONS

Place figures and tables at the top or bottom of the appropriate column or columns, on the same page as the relevant text (see Figure **??**). A figure or table may extend across both columns to a maximum width of 17.78 cm (7 in.).

Captions should be Times New Roman or Times Roman 9-point bold. They should be numbered (e.g., "Table **??**" or "Figure **??**"), centered (if one line) otherwise justified, and placed beneath the figure or table. Please note that the words "Figure" and "Table" should be spelled out (e.g., "Figure" rather than "Fig.") wherever they occur. Figures, like Figure **??**, may span columns and all figures should also include alt text for improved accessibility. Papers and notes may use color figures, which are included in the page limit; the figures must be usable when printed in black-and-white in the proceedings.

The paper may be accompanied by a short video figure (we recommend staying within five minutes in length). However, the paper should stand on its own without the video figure, as the video may not be available to everyone who reads the paper.

6.1 Inserting Images

When possible, include a vector formatted graphic (i.e. PDF or EPS). When including bitmaps, use an image editing tool to resize the image at the appropriate printing resolution (usually 300 dpi).

7. QUOTATIONS

Quotations may be italicized when *"placed inline"*.

> Longer quotes, when placed in their own paragraph, need not be italicized or in quotation marks when indented.

8. LANGUAGE, STYLE, AND CONTENT

The written and spoken language of SIGCHI is English. Spelling and punctuation may use any dialect of English (e.g., British, Canadian, US, etc.) provided this is done consistently. Hyphenation is optional. To ensure suitability for an international audience, please pay attention to the following:

- Write in a straightforward style.
- Try to avoid long or complex sentence structures.
- Use common and basic vocabulary (e.g., use the word "unusual" rather than the word "arcane".
- Briefly define or explain all technical terms that may be unfamiliar to readers.
- Explain all acronyms the first time they are used in your text—e.g., "Digital Signal Processing (DSP)".
- Explain local references (e.g., not everyone knows all city names in a particular country).
- Explain "insider" comments. Ensure that your whole audience understands any reference whose meaning you do not describe (e.g., do not assume that everyone has used a Macintosh or a particular application).
- Explain colloquial language and puns. Understanding phrases like "red herring" may require a local knowledge of English. Humor and irony are difficult to translate.
- Use unambiguous forms for culturally localized concepts, such as times, dates, currencies, and numbers (e.g., "1–5–97" or "5/1/97" may mean 5 January or 1 May, and "seven o'clock" may mean 7:00 am or 19:00). For currencies, indicate equivalences: "Participants were paid ₩ 25,000, or roughly US $22."
- Be careful with the use of gender-specific pronouns (he, she) and other gendered words (chairman, manpower, man-months). Use inclusive language that is gender-neutral (e.g., she or he, they, s/he, chair, staff, staff-hours, person-years). See the *Guidelines for Bias-Free Writing* for further advice and examples regarding gender and other personal attributes . Be particularly aware of considerations around writing about people with disabilities.
- If possible, use the full (extended) alphabetic character set for names of persons, institutions, and places (e.g., Grønbæk, Lafreniére, Sánchez, Nguyễn, Universität, Weißenbach, Züllighoven, Århus, etc.). These characters are already included in most versions and variants of Times, Helvetica, and Arial fonts.

9. ACCESSIBILITY

The Executive Council of SIGCHI has committed to making SIGCHI conferences more inclusive for researchers, practitioners, and educators with disabilities. As a part of this goal, the all authors are asked to work on improving the accessibility of their submissions. Specifically, we encourage authors to carry out the following five steps:

1. Add alternative text to all figures
2. Mark table headings
3. Add tags to the PDF
4. Verify the default language
5. Set the tab order to "Use Document Structure"

For more information and links to instructions and resources, please see: `http://chi2016.acm.org/accessibility`. The `\hyperref` package allows you to create well tagged PDF files, please see the preamble of this template for an example.

10. PAGE NUMBERING, HEADERS AND FOOTERS

Your final submission should not contain footer or header information at the top or bottom of each page. Specifically, your final submission should not include page numbers. Initial submissions may include page numbers, but these must be removed for camera-ready. Page numbers will be added to the PDF when the proceedings are assembled.

11. PRODUCING AND TESTING PDF FILES

We recommend that you produce a PDF version of your submission well before the final deadline. Your PDF file must be ACM DL Compliant. The requirements for an ACM Compliant PDF are available at: `http://www.scomminc.com/pp/acmsig/ACM-DL-pdfs-requirements.htm`.

Test your PDF file by viewing or printing it with the same software we will use when we receive it, Adobe Acrobat Reader Version 10. This is widely available at no cost. Note that most reviewers will use a North American/European version of Acrobat reader, so please check your PDF accordingly.

12. CONCLUSION

It is important that you write for the SIGCHI audience. Please read previous years' proceedings to understand the writing style and conventions that successful authors have used. It is particularly important that you state clearly what you have done, not merely what you plan to do, and explain how your work is different from previously published work, i.e., the unique contribution that your work makes to the field. Please consider what the reader will learn from your submission, and how they will find your work useful. If you write with these questions in mind, your work is more likely to be successful, both in being accepted into the conference, and in influencing the work of our field.

13. ACKNOWLEDGMENTS

Sample text: We thank all the volunteers, and all publications support and staff, who wrote and provided helpful comments on previous versions of this document. Authors 1, 2, and 3 gratefully acknowledge the grant from NSF (#1234–2012–ABC). *This whole paragraph is just an example.*

14. REFERENCES FORMAT

Your references should be published materials accessible to the public. Internal technical reports may be cited only if they are easily accessible and may be obtained by any reader for a nominal fee. Proprietary information may not be cited. Private communications should be acknowledged in the main text, not referenced (e.g., [Golovchinsky, personal communication]). References must be the same font size as other body text. References should be in alphabetical order by last name of first author. Use a numbered list of references at the end of the article, ordered alphabetically by last name of first author, and referenced by numbers in brackets. For papers from conference proceedings, include the title of the paper and the name of the conference. Do not include the location of the conference or the exact date; do include the page numbers if available.

Citing something[**?**]. Citing something else.[**?**]

REFERENCES

1. Google Drive Package in R. https://googledrive.tidyverse.org/, 2023.
2. Leaflet for R - Introduction. https://rstudio.github.io/leaflet/, 2023.
3. MySQL for Python. https://dev.mysql.com/doc/connector-python/en/connector-python-coding.html, 2023.
4. Openmeteo API. https://open-meteo.com, 2023.
5. Rsconnect Package in R. https://rstudio.github.io/rsconnect/, 2023.
6. West Tennessee Solar Farm. https://solarfarm.tennessee.edu/about/, 2023.
7. Winston Chang. R-Shiny Documentation. https://www.rdocumentation.org/packages/shiny/versions/1.7.5.1, 2023.
8. 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.
9. Hadley Wickham. *Mastering Shiny: Interactive Apps, Reports, and Dashboards Powered by R*. O'Reilly, 2021.