End of Quarter Presentation

3/14/25

Introductions

SU Computer Science Capstone Team:

- Jules Hunter
- Joshua Baron
- Jessica Huang
- J Guadalupe Andrade

Sponsors:

- Dr. Peter Dauenhauer
- Daniel Nausner

Advisor:

Dr. Burkhard Englert

Organizations

Project Overview

The Microgrid Toolkit is designed to help plan for the long-term impact and sustainability of microgrid installations. The main feature of the tool is a customizable simulation that predicts energy production and consumption as well as socioeconomic impacts of a microgrid system. It is a web application that should enable system designers to optimize their plans for energy solutions in developing countries.

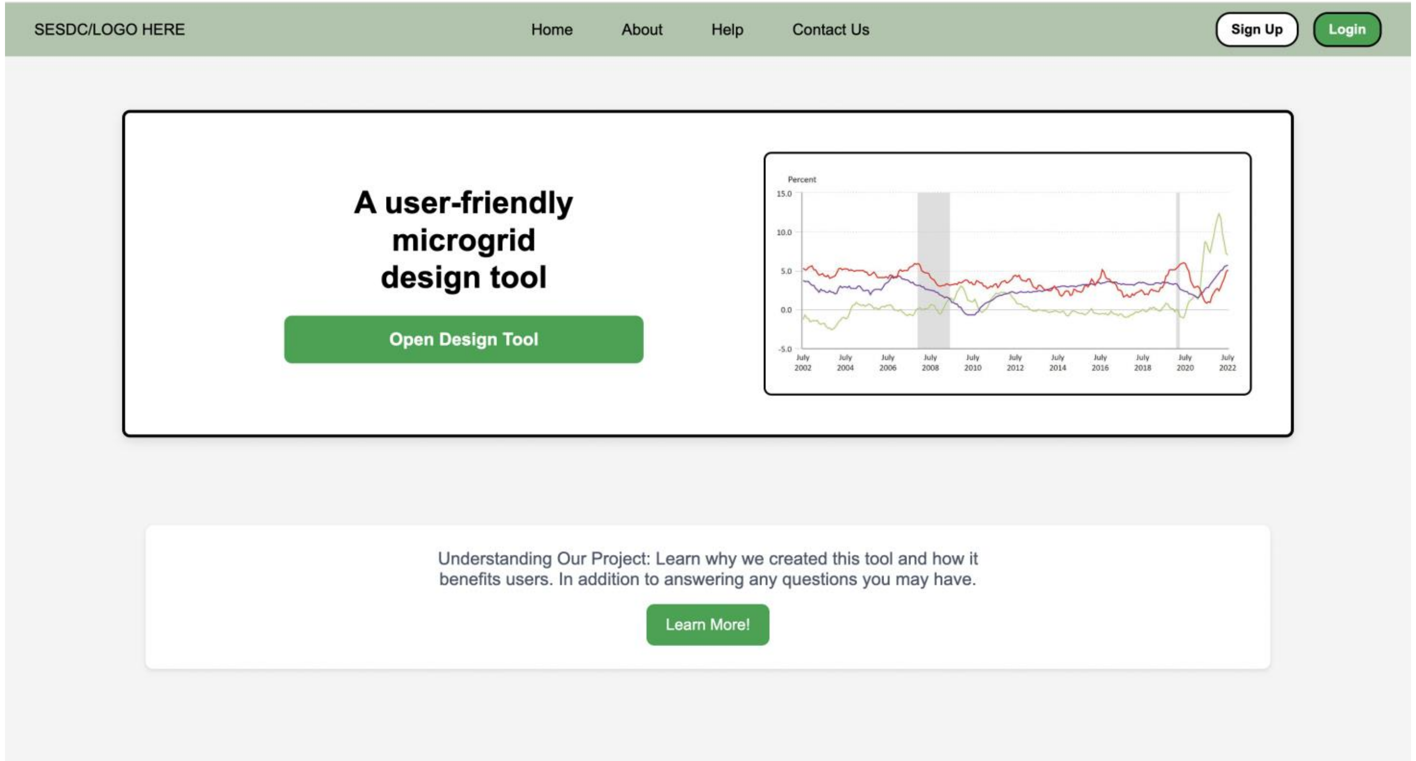
Project Goals

- Develop a user-friendly microgrid design tool for non-technical users.
  - Desktop & Mobile functionality
- Guide users through project creation, system input, simulations, analysis, and storage.
- Improve energy scripts
  - Easily query resource data such as solar irradiance, wind and temperature by location.
  - Consolidate and Improve performance of previous simulation scripts
- Simulation & Results Analysis
  - Generate Graphics and plots of project outputs
  - 20-year energy projections
  - Financial and impact analysis

Frontend - Jessica

- Design User Interface
- Implement User Interfaces from wireframes
  - Using React
- Ensures usability and accessibility

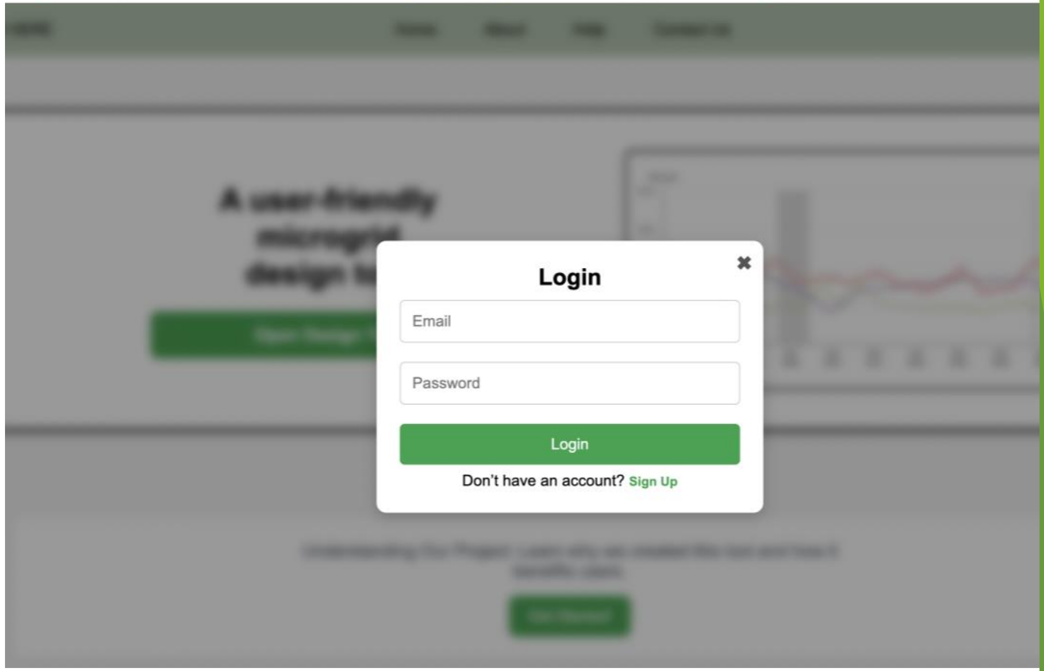
Homepage

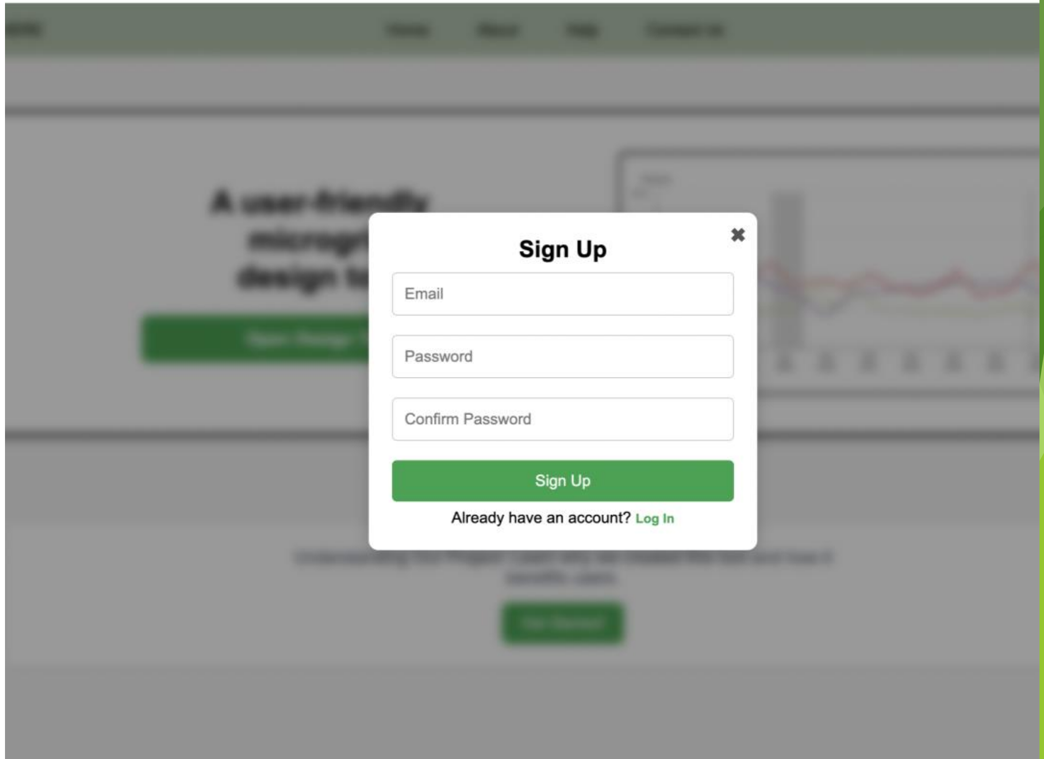


- Currently only Sign up + Login buttons are working
- Graph is clickable
  - Leads to tutorial
- Learn More! Leads to an in-depth manual

Sign Up/Login Page

Pop up format optimizes user experience





Progress so far - Jessica

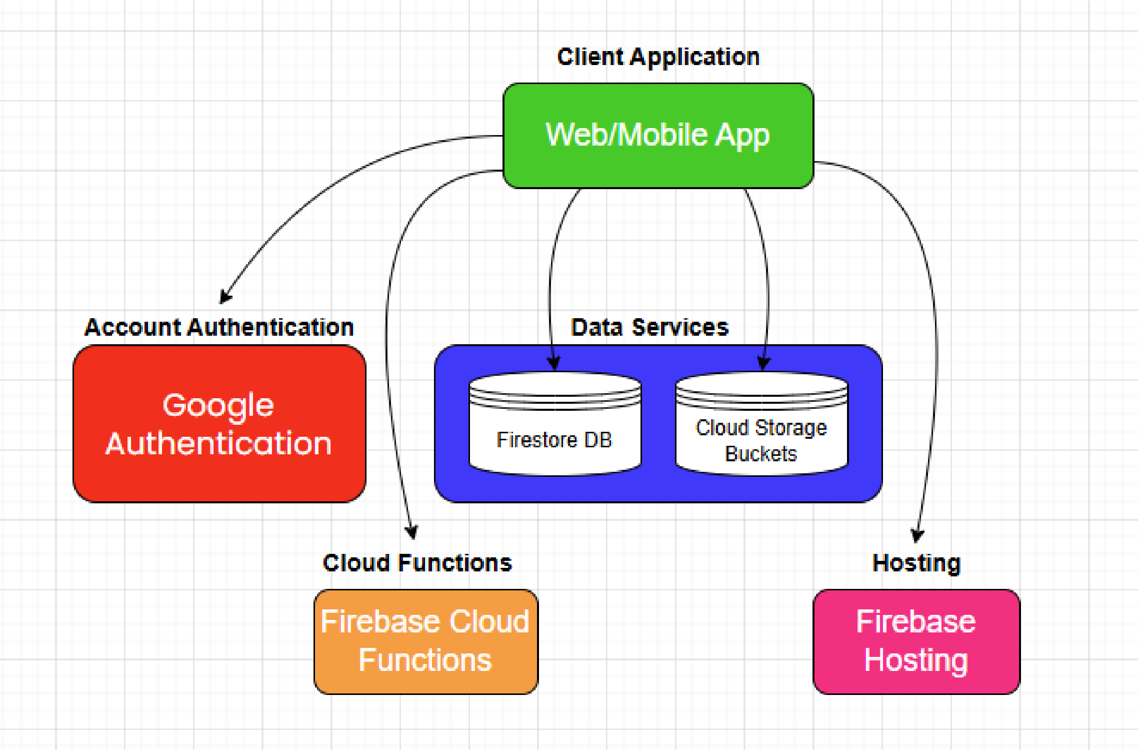
- Designed + Implemented the homepage
  - Translated wireframe into react code
- Connected sign up/login page to homepage
- Connected login page to test page

Future Plans - Jessica

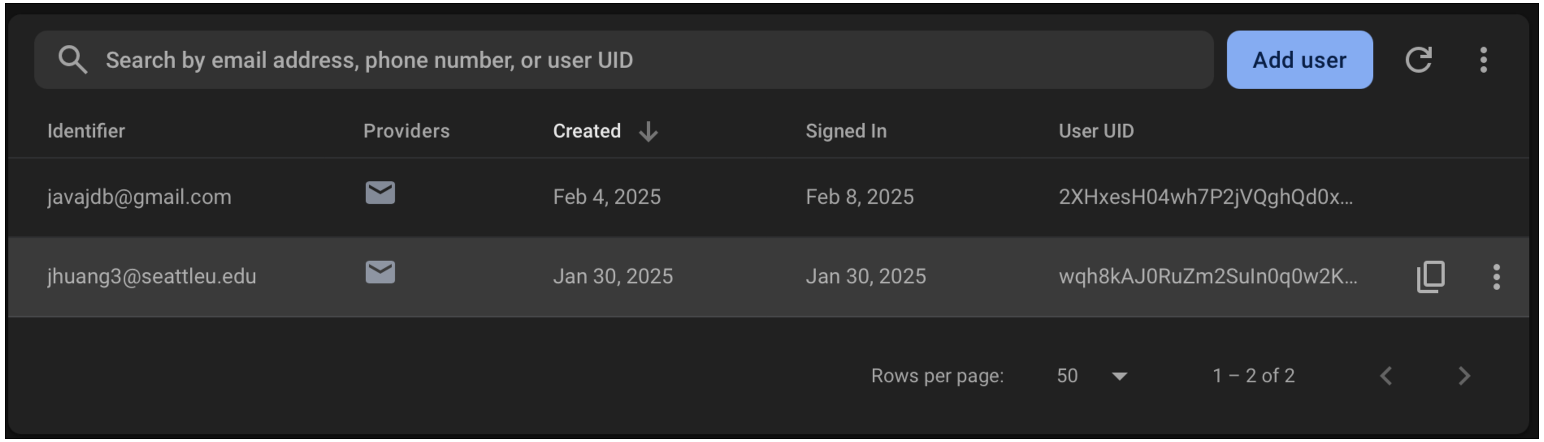
- Homepage for returning users
  - Includes quick access to project
- Create user manual
- Implement footer
- Create rest of pages
  - Project design
  - About

Josh - Backend

- Google Firebase (Google Cloud)
  - Google Authentication
  - Google Firestore
  - Google Functions
  - Google Storage
  - Google Hosting



User Login and Registration



- Firebase Authentication
  - Email and Password
  - Secure Sign In

Data Storage

gs://sesdc-function-test.firebasestorage.app > 2XHxesH04wh7... > 3JYybrzbMZ8N...

Upload file

Name	Size	Type	Last modified
_empty	—	application/octet-stream	Feb 6, 2025
diesel_plot_plot.png	20.63 KB	image/png	Feb 7, 2025
energy_plot.png	49.41 KB	image/png	Feb 7, 2025
load_plot.png	51.43 KB	image/png	Feb 7, 2025
net_energy_plot.png	71.07 KB	image/png	Feb 7, 2025
test_app_wind.csv	125.49 KB	text/csv	Feb 7, 2025
wind_plot.png	42.14 KB	image/png	Feb 7, 2025

- Firestore
  - No SQL database
- Cloud Buckets
  - Stores User Project Data
  - Easier to Store Project Data in Cloud Buckets
  - Cost Efficiency

Running the Simulation

Function		Trigger
run_simulation us-central1	HTTP	Request https://run-simulation-2e75hqar4q-uc.a.run.app
fetch_solar_data_function us-central1	HTTP	Request https://fetch-solar-data-function-2e75hqar4q-uc.a.run.app

- Google Cloud Functions
  - Run Backend Simulation Scripts

Progress so Far - Josh

- User Login and Registration features
- Project Selection Page
  - Add Projects
  - Delete Projects
  - Share Projects
- Connected Database to Store User Information and Project Information
- Connected Backend Scripts to Website

Future Plans - Josh

- Password Recovery
- Security Features
- Website Hosting

Integration of API Data for Solar & Wind Energy - Jose

- Solar, Temperature, Wind Data API Selection
- Query, Gather, Process Data
- User Interface Implementation
- Simulation Calculation Implementation

Progress so Far - Jose

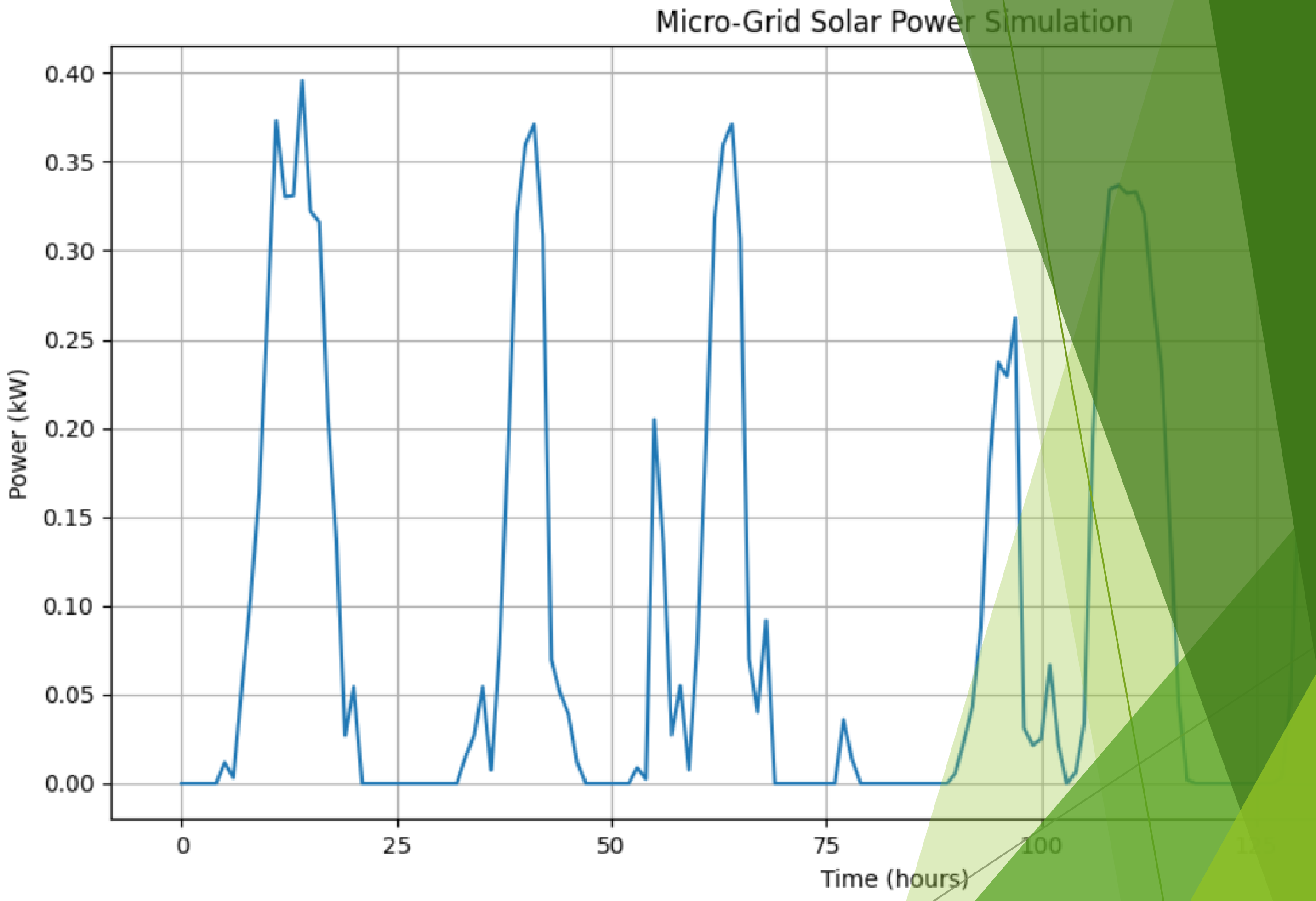
- API Found (NSRDB: National Solar Radiation Database)
- Data has been retrieved & calculated
  - Zambia
- The data has been structured in CSV format
- Method created to fit the UI

Future Plans - Jose

- Ensure data is useful & reliable
- Reduce data wait time
- Include areas outside of Africa
- Allow for user selection of displayed CSV data

Jules - Scripts

- Consolidating and improving performance of legacy code
  - Solar
  - Wind
  - Diesel
  - Load
  - Net Energy
- Energy Storage System (batteries, curtailment)



Solar

- Using a panel Output Power formula
  - Other options considered: PVWatts Model, Machine Learning
  - Accounts for losses due to temperature, wiring, aging, dust, converter
- Hourly data default

Solar Equation

$$P_{\text{out}} = \left(\frac{I}{STC_{\text{irr}}}\right) \times \left[P_{\text{rated}} + P_{\text{rated}} \times \left(\frac{\text{coef}}{100}\right) \times (T_{\text{cell}} - STC_{\text{temp}})\right] \times \prod_i (1 - \text{losses}_i)$$

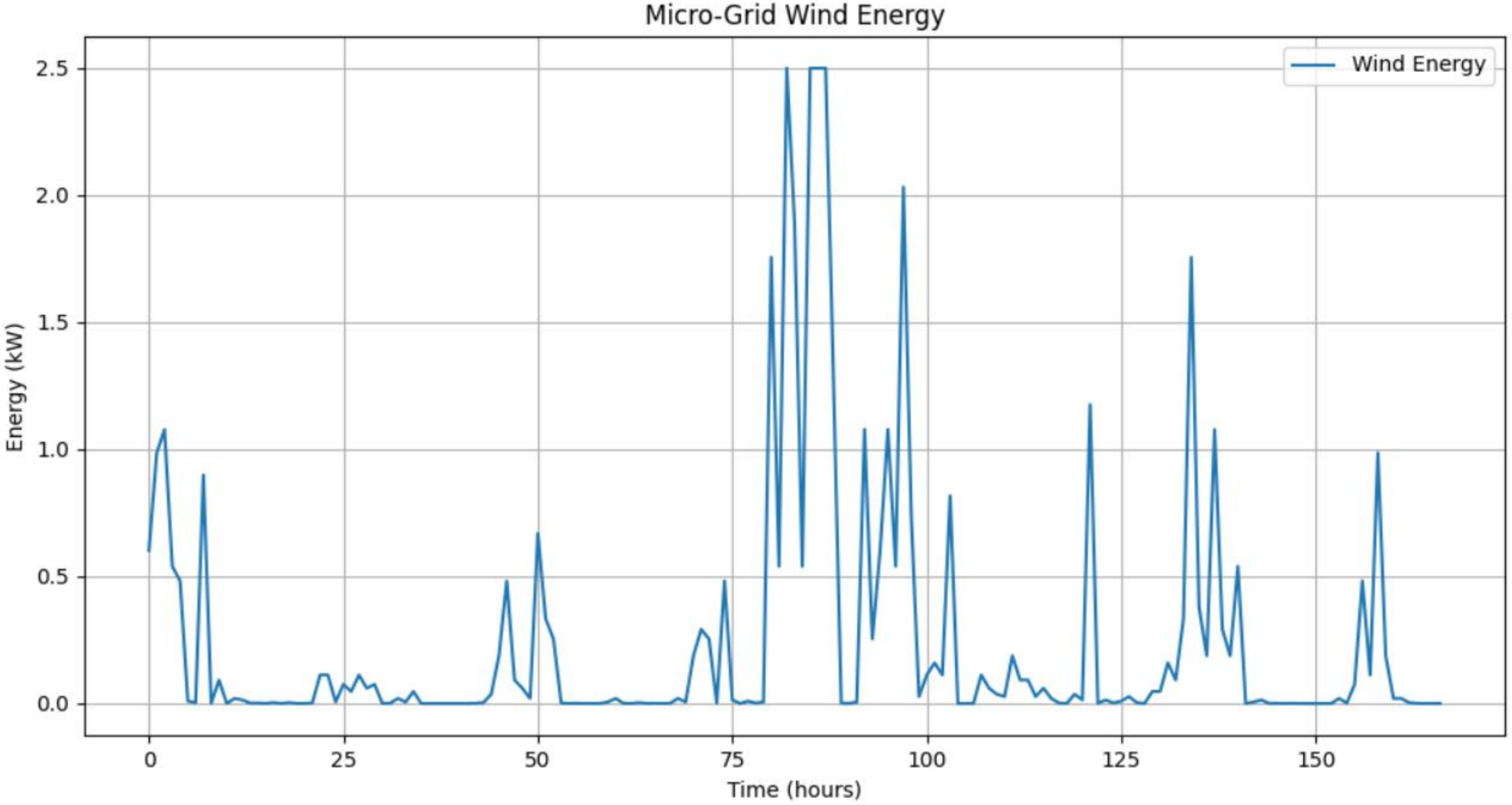
- P_{out} = Solar power output (W)
- I = Solar irradiance (W/m^2)
- STC_{irr} = 1000 W/m^2 (Standard Test Conditions irradiance)
- P_{rated} = Rated panel power (e.g., 680 W)
- coef = Temperature coefficient (e.g., -0.45)
- $T_{\text{cell}} = T_{\text{ambient}} + 0.0563 \times I$ (Cell temperature adjusted for irradiance)
- STC_{temp} = 25°C (Standard Test Conditions temperature)

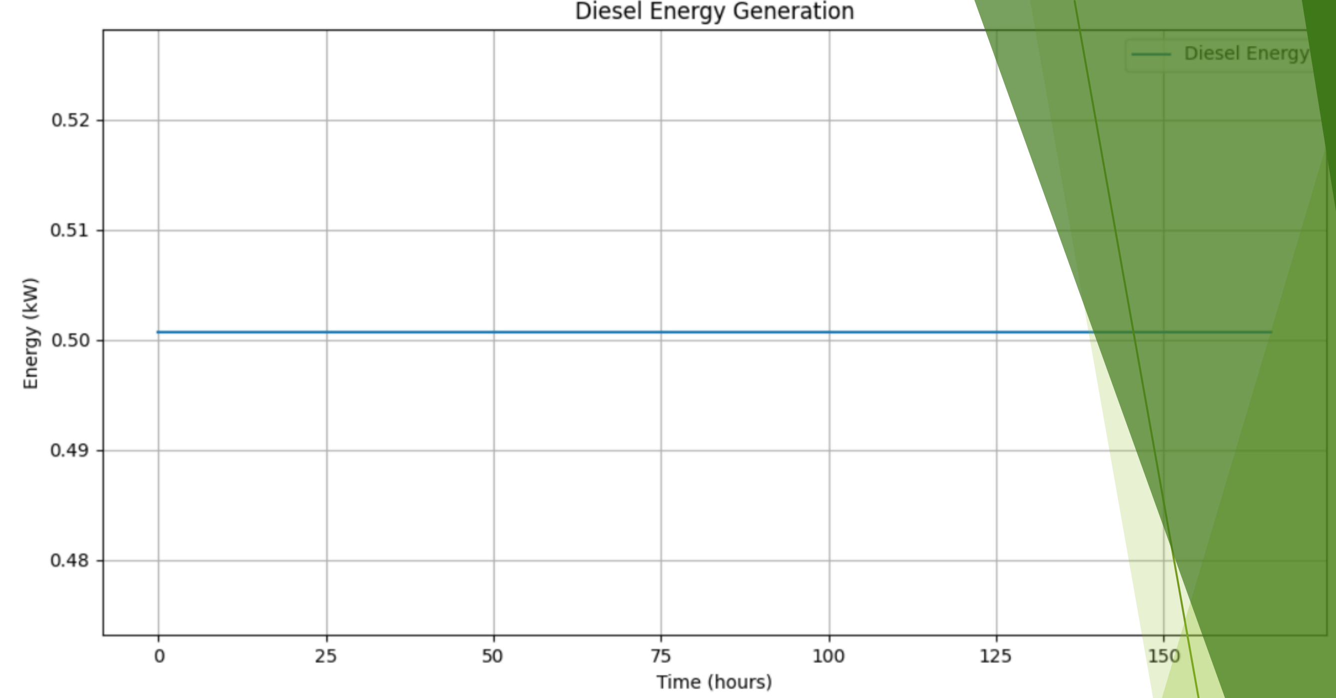
- $\prod_i (1 - \text{losses}_i)$ accounts for efficiency losses:
  - Wiring losses (10%)
  - Module mismatch losses (10%)
  - Module aging losses (8%)
  - Dust/dirt losses (11%)
  - Converter losses (5%)

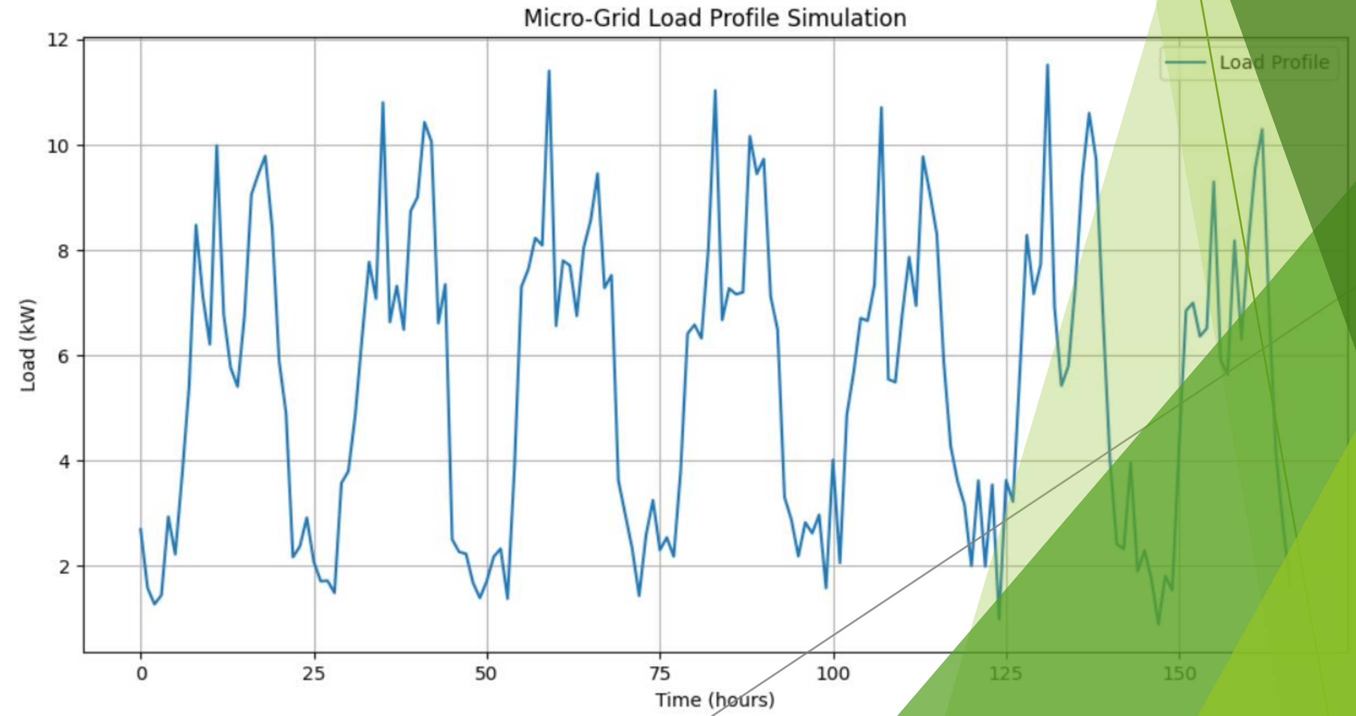
$$P_{\text{wind}} = \begin{cases} 0, & V < V_{\text{cut-in}} \text{ or } V > V_{\text{cut-out}} \\ P_{\text{rated}} \times \left(\frac{V - V_{\text{cut-in}}}{V_{\text{rated}} - V_{\text{cut-in}}} \right)^3, & V_{\text{cut-in}} \leq V < V_{\text{rated}} \\ P_{\text{rated}}, & V_{\text{rated}} \leq V \leq V_{\text{cut-out}} \end{cases}$$

Wind

- Wind data is first converted from km/h to m/s
- Converts power output to kW

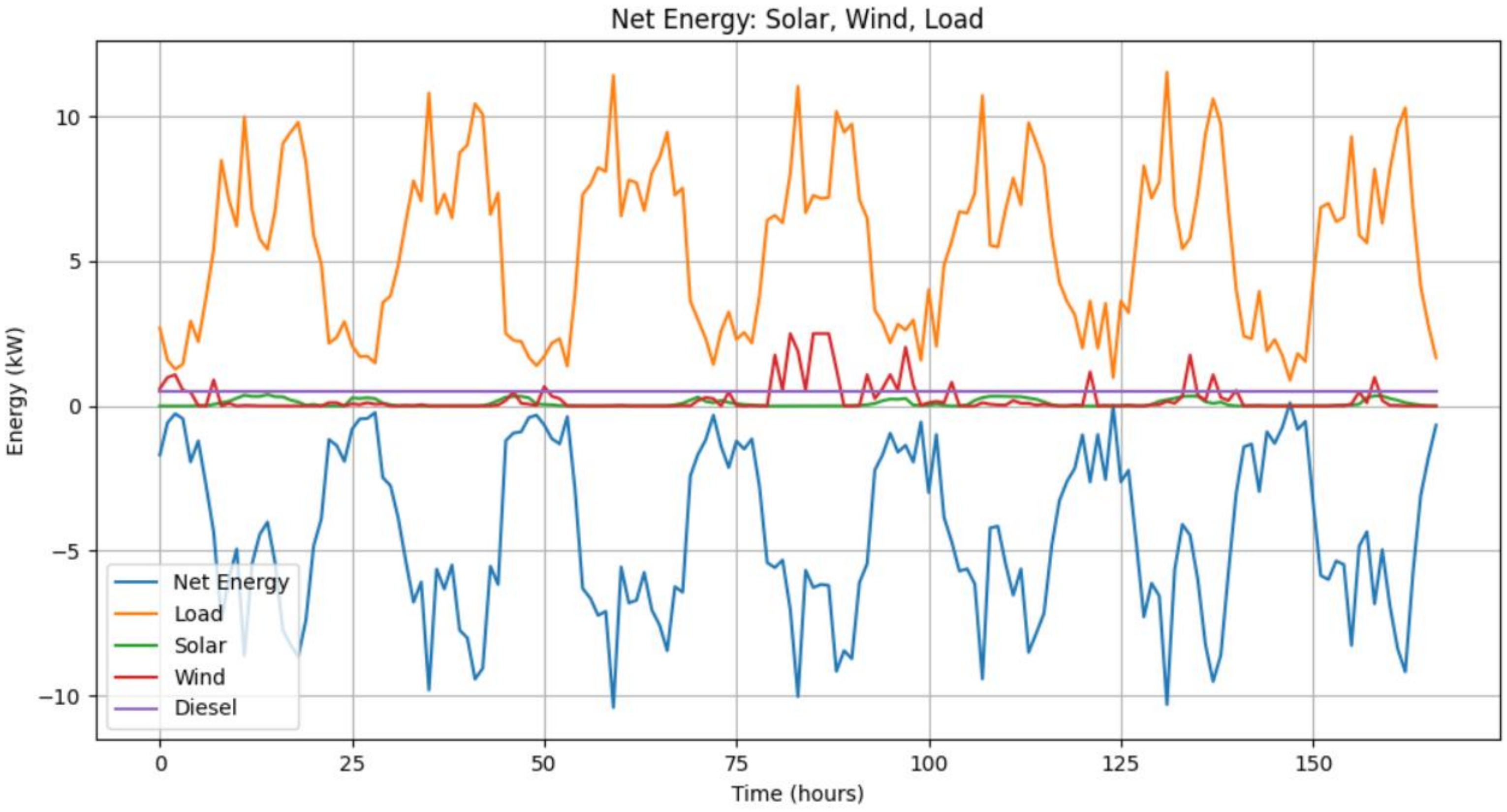






Diesel & Load

- Reads fuel consumption and generator output from dataset, and applies losses
  - Losses: engine efficiency, generator efficiency, transmission, converter
- Load: aggregates 5 loads



Net Energy – Without Batteries

Progress so far - Jules

Calculations implemented:

- Solar Power Output
- Load Profile
- Wind Energy Generation
- Diesel Energy Generation
- Net Energy Balance

Matplotlib used for plotting graphs

All graphs saved and displayed in the web app

Future Plans - Jules

BATTERY STORAGE MODELING

USER INPUT & WEB UI ENHANCEMENTS

INTERACTIVE GRAPHS

OPTIMIZING ENERGY MIX FOR DIFFERENT DEMAND SCENARIOS

Q & A Time!
Questions?