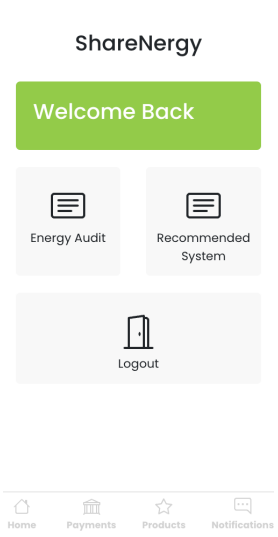
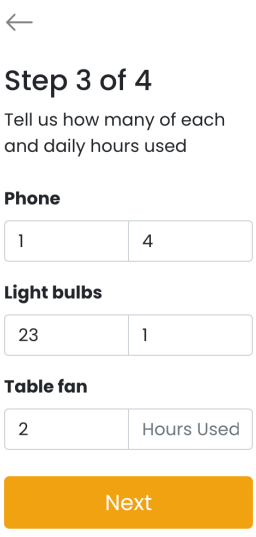
ShareNergy

ESE Senior Design Team 16
04/29/2022

Amine Soufaih	SSE	asoufaih@wharton.upenn.edu
Elijah Jasso	CIS	elijahj@seas.upenn.edu
Jonathan Jenkins	EE	jrjenk@seas.upenn.edu
Lisa Moshiro	CIS	lmoshiro@seas.upenn.edu
William Wang	SSE	wsquared@wharton.upenn.edu

Advisors
Sid Deliwala - deliwala@seas.upenn.edu
Jan Van der Spiegel - jan@seas.upenn.edu





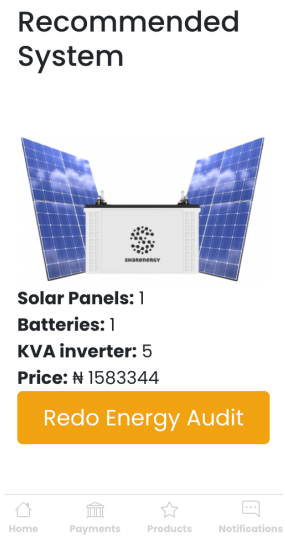


Table of Contents

Executive Summary

This project aims to create a platform for Nigerians to easily access renewable energy solutions and financing. We are partnered with a startup called ShareNergy based in Nigeria that aims to address the energy deficit in the country by providing easier access to solar. The platform will aggregate customers, suppliers, and financiers and make it easier for individuals and small businesses to access renewable energy solutions.

Our solution involves a web app where users can sign-up and take an energy audit questionnaire to receive a customized renewable energy system recommendation through a custom algorithm. The data is stored in our backend and users can modify their answers at any time to receive an up-to-date recommendation. From this recommendation, users can use one-click to contact ShareNergy regarding the recommendation and next steps. Overall, we hope our solution will help renewable energy play an important role in providing better access to energy for many Nigerians.

Overview and motivation of Project

Nigeria is facing a severe energy deficit. Only 50-60% of Nigerians are connected to the grid, which is inconsistent, only providing on average 4 hours of electricity per day, often with multi-day outages. As a result, many Nigerians and businesses rely on diesel generators to power their lives, spending $14 billion per year on generator fuel. These are expensive, and emit an estimated 29 million tons of CO2 per year. In addition, these generators can cause major long term health risks for those working near them. So, a major opportunity exists in Nigeria for renewable energy, but the current system makes it difficult for most people and small businesses to access solar panels and batteries.

ShareNergy is trying to change that by providing a one-stop-shop for Nigeria's small businesses to begin the process of installing solar. By providing a simple, easy to use platform, we simplify the process of finding solar suppliers and installers, and offer financing options to better fit within a normal budget. Through ShareNergy, accessing solar energy in Nigeria has never been easier.

Technical Description

Our solution with ShareNergy to help provide access to renewable energy sources to Nigerians involves a full-stack web-app with a user-friendly interface and backend for handling data storage and data/algorithm processing. To accomplish this we decided on using the Python Django Framework for the backend and database handling and a React frontend (with some Bootstrap styling). JSON Web Tokens (JWTs) are used for user authentication.

We chose these technologies specifically for multiple reasons. One large consideration was maintainability and scalability since this project is both for this ESE class and for an actual startup. Both Django and React are actively developed and used regularly in industry. Both are considered very scalable and used in both small projects and very large projects. Many people have the skills to upkeep Django and React codebases, or can easily learn how to do so, so in the future ShareNergy should have no trouble finding people to help work on this project.

Another consideration was making sure the technologies had a robust featureset. Django was a good choice for this because it has standard features of database management and user account management built-in, utilizing industry standards, ensuring best security and design practices for these features. Similarly, React creates an easier experience for working on a frontend, allowing easy creation, manipulation, and handling of HTML objects using Javascript.

Finally, a large portion of our decision to use a Python backend (Django) was because we decided our underlying matching algorithm should use Python. We considered using Javascript, but after considering the downsides of Javascript being slow, having some odd debugging quirks, and not having as many well-known math libraries, we decided to stick to Python. Python is good for our algorithm because it is easy to use, easy to understand, and has several math, data, and machine learning libraries that we could use if needed.

Our Django backend holds endpoints for our frontend to communicate with to create and log in to user accounts, store user account information, retrieve user details, and run our matching algorithm. The Django web framework uses Python for programming language, and an SQLite database, which is accessed also through Python syntax. Passwords are stored using the PBKDF2 algorithm with a SHA256 hash. This is standard with Django and is considered secure.

Our React frontend consists of five distinct pages. The sign-up page takes in user information and sends it to the backend to create a new account for the registering user. The login page takes in the email and password of an already registered user and receives a JWT token from the backend for authentication purposes. The energy audit page asks the user questions about their energy usage, the appliances they use, and how long they use their appliances each day. The system recommendation page shows the user their recommended system size–with number of solar panels, number of batteries, size of inverter, and estimated price–based on the user's responses to the energy audit. Additionally, from the energy audit page the user can contact ShareNergy directly through email to discuss the purchasing process. Finally, the dashboard page allows a logged-in user to easily access their system recommendation or edit their energy audit response.

The algorithm is written in Python and takes energy needs information from the frontend energy audit form and combines this information on appliances usage and renewable energy generation to suggest a suggested number of solar panels, number of batteries, and size of inverter. The suggested system size must fit a number of constraints based on the energy audit information provided by the user. Based on that information, plus industry averages for various appliances, we can calculate both the total energy that the solar panels and batteries need to

supply and the instantaneous power that they must be able to generate. In addition, the algorithm takes into account various types of energy loss and inefficiencies, and returns the optimal system size for the customer. Going forward, we hope to add in functionality to offer multiple options for customers, to refine our assumptions about appliance energy consumption and sunlight to create more accurate estimates, and to integrate our suggestions with supplier information about the types of panels, batteries, and inverters they have in stock.

Look at Figure 1.1 to see a diagram of our web-app.

Since we are performing computation on the backend of our web-app, we decided that we would need a scalable cloud provider to host our service. AWS Beanstalk seemed like a good choice for this as it has built services to tweak server specifics in order to create the best hosting environment for our application. We had other choices such as AWS EC2 combined with other services; however, we decided a more intuitive cloud product would be the best to pass off to ShareNergy. In addition, our team has the most experience with AWS which is why we chose this over a GCP or Azure product. This way ShareNergy will be able to run our app on the cloud without needing to spend extra time on maintenance. In order to deploy our app to AWS, we first started by researching deployment of the Django app to test the accessibility to our backend. After this documentation seemed clear, the React app would need to be linked in our backend files so it could be accessed through the backend. After this and deploying, the server could now recognize routes for clients and send frontend files accordingly. Cloud services charge for the time the app is deployed even if no one is accessing the app; therefore, the app was not deployed by our team as it would quickly consume our budget. We will provide this documentation to ShareNergy and allow them to use their company AWS accounts to deploy the app to the public.

Overall, both our ESE team and ShareNergy are satisfied with the product we developed as we were satisfying both this ESE class and the needs of the ShareNergy startup. Our project setup, which allows for good maintainability and scalability, will be expanded on next in the future by the ShareNergy team.

Self-learning

In order to create the energy algorithm, our team had to learn how solar power is harnessed and what components make up a solar system. We also have had to learn how to use React libraries like Bootstrap in order to format the UI of our web app. In terms of technical knowledge, for the backend we had to learn how to use Django's password authentication in order to protect users' passwords. Learning about solar power and solar systems has been very interesting, and the technical aspects have not been too hard to learn as well.

In terms of the web application, CIS350 (Software Engineering) and CIS 550 (Databases) were very useful classes. In 350 we learned how to create a web app using Javascript which we are using for the frontend. We also learned how to work with a team in order to create a web app. In

CIS 550 we learned how to work with and send data to an SQL database. The team has taken various data science courses such as CIS545, ESE204, and ESE305 which have been very helpful in writing the algorithm, as the algorithm is written in Python and uses libraries such as Pandas to process the data.

Since we meet with Sharenergy on a weekly basis, we have been able to get a lot of very valuable feedback during our development process. In terms of frontend development, the CTO and design lead at the startup give feedback on the design of our app when we demo each week which we then incorporate. Another thing is ensuring ownership of our work. Since we are working with a startup, our advisor Sid advised us to make sure we get credit for the work we put in.

Ethical and Professional Responsibilities

There are a large number of ethical considerations we must take into account with our project as we are working to recommend energy solutions to Nigerians. There are many ethical dilemmas related to being the sole proprietors of energy needs and solutions for many individuals and small businesses. Assuming customers are dependent on us for their energy solutions, we must ensure high quality in our product and web application to avoid proper handling of sensitive information and important calculations.

The main consideration is the accuracy and consistency of our algorithm. When the final version of our product is built, we would be the only ones in the pipeline to recommend a system size using the customers' energy needs. With this huge task comes huge responsibility as we must ensure the quality of our recommendations. The customers are the main stakeholders in this process and an inaccurate calculation or recommendation could be devastating financially, physically, etc.

In order to facilitate high quality product recommendations, we were looking mainly at the ISO 9001 quality management guide to providing consistent high quality to our customers. We are engaging with ShareNergy to be customer focused, to have a careful approach to processes, and to aim for consistent improvement over time. In the future, we want to focus on the evidence-based decision making QMS principle found in ISO. Our calculations rely on assumptions made about the appliance use of power and we want to ensure that this is checked and updated regularly if necessary. Additionally, we want to ensure the solar cells and batteries recommended to the customers are reliable and provide the proper amount of energy that was promised to the customer.

Another concern that we are considering is our customers' privacy. We don't require very sensitive information in our web application thus far; however, we still want to be conscious of storing user information such as name, password, and energy needs. As mentioned above, Django has a built-in password authentication tool which is helpful for storing hashed passwords and checking logins without revealing real passwords in the database.

Meetings

As a team, we had multiple set weekly meetings throughout the semester. Our internal team meeting is on Monday at 8:30 where we plan the week ahead of us and split the work among all the members in our group. This meeting usually lasts about an hour. Our second meeting is on Thursdays at 7pm where we meet with the SharEnergy team based in Nigeria over Zoom. In this meeting, we share our progress with the founding team and ask them any questions we encountered as we are building the system. Besides our team, ShareEnergy CEO, CFO and engineering team are always present in the meeting to guide us through the process. Thirdly, we usually meet as subgroups within our team to make progress on our work. For example, Elijah and Lisa have been meeting regularly to build both the frontend and backend of our system while Will, Jack, and Amine would also meet separately to build the matching algorithm. Finally, We also met with our advisors once a month to update them about our progress and get their advice on our project.

Schedule with Milestones

Below is a table of our current intended deadlines for the spring semester.

	Lisa	**Elijah**	**Jack**	**Amine**	**Will**
Finalize AWS deployment strategy with shareNergy	02/01	02/01	02/01		
Look into potential ML pipeline opportunities				02/15	02/15
UX improvement and test deployment	02/28	02/28			
Implement necessary backend learning algorithms			03/15	03/15	03/15

Testing and cleaning	04/01	04/01	04/01	04/01	04/01
Deployment on AWS	04/15	04/15	04/15	04/15	04/15

In terms of our milestones for the fall semester, our team was able to accomplish everything we wanted besides deployment on AWS. This makes sense though as we want to thoroughly test our application before deployment as it will be widely accessible and representing ShareNergy. We were provided with a lot of useful data from ShareNergy which sped up the development process for the algorithm. For example, ShareNergy gave us a better idea of what appliances their customer base would be using and what types of those appliances were most popular. This allowed us to better tune the algorithm for their customers.

Last semester, we had a functioning skeleton of our app and we were able to take in user requests and provide basic recommendations from our algorithm. From what we discussed in our lightning talk, this looked like what we wanted our final product to be; however, we planned on making significant improvements over this semester.

As planned we greatly improved the UX/UI components of our web-app ensuring that it works on a wide variety of screen sizes and both on desktop and mobile devices. Also we improved navigation of the webapp and made the color themes more consistent and visually appealing. Users can now login and they are taken to a dashboard where they can easily see what other web-app features they can access and easily navigate to.

The AWS deployment was researched throughout the semester. We gathered documentation to first deploy the django app without the frontend. After further research it was determined the frontend would need to be embedded within the static files of the django app in order for the server to access it. At this point the entire app could be viewed by the public. More experimentation is still needed to fully deploy the app which will be handed off to ShareNergy in a few weeks.

One idea that our team had at the end of last semester was to create some feedback to our algorithm in order to improve product recommendation and overall user experience. The main issue with tackling this problem was data collection to create a learning model. ShareNergy is still in the early stages of their customer journey, and we simply didn't have access to enough data in order to make a meaningful improvement to our algorithm. Despite this, we were still able to achieve improvements to our user experience through a dashboard for returning users. This made the product more welcoming and useable for those who wanted to revisit their energy audit or access additional information about ShareNergy's products.

Discussion of Teamwork

Our team consists of 5 members and we have been sure to each work on significant portions of the project according to our own skills and the needs of the project. For communication we have a Facebook Messenger chat and a Slack channel with the ShareNergy startup, both of which we use often. Additionally, we have a Notion page where we keep track of tasks with information on who is responsible for each and their current progress. We meet at very least once every week as an ESE team and once more a week with the ShareNergy startup team. In these meetings we decide divisions of labor and additional meetings are planned as needed to complete those tasks.

We have done both group work and individual tasks, depending on which is most feasible and beneficial to the project. While we sometimes worked separately we made sure everyone was aware of our current progress and any roadblocks we had along the way.

For the current iteration of our project we subdivided our team into three main sections for work, one to work on the web app and database, one to work on the AWS hosting, and the other to work on the underlying algorithm for our project. We all kept up with each other, but also had additional meetings with our subteams.

The web-app was mainly done by Lisa Moshiro and Elijah Jasso. Lisa took charge of the backend of the web-app, using Python and the Django framework to create initial models of users and appliances and accepting requests from the client-side frontend. Elijah worked mainly on the frontend of the web-app, using Node.js, Javascript, and React to create the client-side, which handles frontend user authentication, asks users questions about their energy usage and needs, and communicates with the backend to retrieve energy audit results. Lisa and Elijah worked together to ensure that both of their portions could communicate with each other easily in a standardized manner.

The algorithm was mainly worked on by Amine, Jack, and William. Jack helped analyze the input data which consisted of energy usage for each appliance that was provided by shareNergy. Will and Amine wrote and tested the Python code for the algorithm, with Will working on the calculations for total energy consumption and power and Amine working on calculating the number of panels and batteries to recommend.

Budget and justification

Our project is a software-based solution with little need for capital at this stage. Our original budget consisted mostly of AWS credit in order to host our web application. This was part of our requests from the ESE department concerning project budgets. Aside from that, there weren't any expenses necessary to run the project. We also work closely with the SharEnergy team which has raised funding from investors so we have been able to get resources from them without having to spend our own budget. For example, the local SharEnergy team did an extensive market research on the viability of the product within their market segment and had to

pay for many external sources to get supporting data. We were able to access all that information for free since we're part of the larger team. Therefore, we were able to achieve cost-efficiency and run our project with very low costs unless necessary. As we continued with the project we didn't incur any additional costs. The bulk of our costs still come from hosting on AWS.

Standards and Compliance

The main standards we used for this project were related to protecting user credentials and all personally identifiable information. For general compliance, we researched some industry standards such as ISO/IEC 29100 and ISO/IEC 27001 which outline generic processes to ensure that user information is not accessible to the public. These standards also include algorithms and methods for protecting very sensitive data such as passwords, emails, etc. To comply with this, our team used Django's PBKDF2 algorithm with a password hashing protocol utilizing the sha256 algorithm with salts. Overall, our team believes we have achieved a state of the art system for protecting user privacy which will be the first step to building customer confidence in the product and drawing them back over time.

Market Opportunity, Size, and Growth

Sharenergy is targeting an attractive market opportunity within the Nigerian energy sector. As the most populous country in Africa with over 200 million people, Nigeria is a strategic market full of lucrative opportunities in the energy transition space. This opportunity is mostly apparent in the small and micro enterprise market where a majority of business owners still rely on non-renewable energy sources. There are over 40m micro enterprises which account for almost all of the MSMEs in the country. There are different customer segments targeted include wholesale/retail trade, tailor shop, make-up artists, remote workers, cyber cafes, hair salons, etc. Sharenergy has narrowed its target customer base to better position its product offering and provide the best service to its customers.

The Nigerian micro enterprise energy consumption is a sizable market with ample opportunity for disruption given the falling cost of renewable energy resources over time. With the help of the Sharenergy team, we calculated the market size based on current fuel consumption by micro enterprises. We broke down our calculation based on the cost of fuel per generator, number of generators and the average cost of a generator. Our estimation of the total market size amounts to over $15bn that Sharenergy could tap into in the next 5 years.

Competition and Business Model

Sharenergy offers access to solar energy as a substitute for gasoline generators that are used all throughout the country. These generators are relatively cheap but they require burning fuel which is expensive and bad for the environment. There are little alternatives to fuel generators and solar energy is still a very nascent solution in this market. While there are several solar players for utility scale projects, there are almost no players that target micro projects such as

hair salons and convenience stores. Sharenergy is well positioned to become a competitive player and acquire sizable market share.

In the short term, ShareNergy's first source of revenue will be through a commission for generating sales to renewable energy equipment suppliers. Second, ShareNergy will receive a facilitation fee from banks for brokering relationships with clients. In the long term, ShareNergy hopes to expand to offer its own hardware and financing in-house. In this case, we would receive income from selling the equipment and from interest payments on financing.

Work Done Since Last Semester

A lot of work went into the project this semester that built upon the foundations we laid in the Fall semester.

On the frontend many things were added. Work was done on creating dedicated signup and login pages which use RESTful communication with the backend to properly sign up and login

Of course some of the most obvious changes happened with the frontend styling. Colors were made to be more cohesive with the ShareNergy logo. Pages were made to be more adaptive to different screen resolutions on both mobile and desktop devices. Now the web app both looks better and operates better no matter what device is being used.

We also made some changes to the backend. Last semester, customers could fill out the algorithm and it would save on their web browser. However, they could not go back to see their past process or logging on other computers as we did not have accounts. To remedy this, we first added signup and login capabilities to the backend using JWT authentication and routes to enable these actions. We also enabled users to save their data via another endpoint to the backend. To do this we changed the user model to take in a JSON field of form data that would be edited each time the user edited the form. Finally, we made some algorithm updates, including building in price estimation and accounting for existing solar panels or batteries owned by the customer.

Additionally, we experimented with cloud deployment and discussed the process with ShareNergy. Since Django and react are two separate frameworks, there is some tweaking that needs to be done to fully deploy our app on AWS so the server knows how to handle requests. We plan on putting together some docs for ShareNergy when we hand off the web application so they can make this app accessible to their customers.

Discussion and Conclusion

This project has provided our team a great learning opportunity for software development as well as strategic problem solving for business development. We have created a product that will help ShareNergy scale and acquire customers faster. After automating a big chunk of their operations, we are hoping ShareNergy will be able to focus more time on customer acquisition and making partnerships with manufacturers and financial institutions to grow the business. We faced many challenges while building this product as the features and next steps were always limited by the information and needs of ShareNergy. This became relevant about halfway through the year when we realized the overhead that was associated with recommending financing options to customers. However, we were still able to build in some extra features and focus on perfecting the user experience.

Overall, our team learned how to delegate tasks for a large project in order to meet our milestones. This was emphasized as we were working with people outside of Penn. We were required to delineate everyone's strengths in order to delegate our tasks efficiently. In addition to this we believe we improved our ability to communicate progress to our peers and professors in this class. This involved giving necessary information and avoiding fluff while presenting this information in a relevant order. We learned a ton throughout the year about what it means to work with others on engineering projects and how to communicate effectively with others.

Appendices

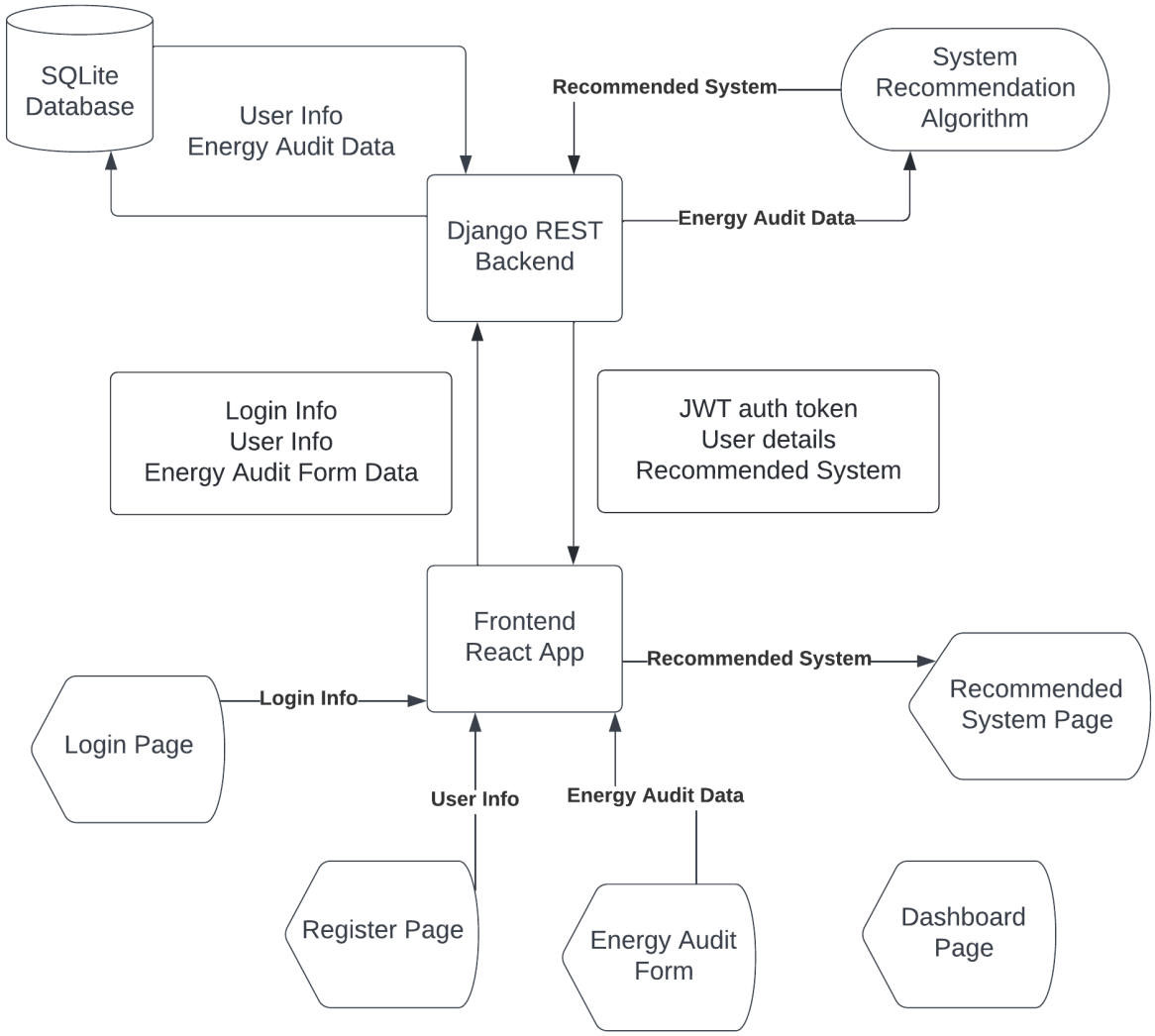


Figure 1.1 Web app and Database Design Diagram