

Peer-to-Peer Optimal Solar Energy Trading using Proof-of-Authority Blockchain in Malaysia

Mohammad Rafaquat Alam, Teoh Jing Yang

Abstract—With Malaysia’s growing population and improved lifestyle, energy demand will only increase posing new challenges and problems to the energy sector. We propose a peer-to-peer (P2P) web-based solar energy trading web application with the aim of optimizing energy usage by homes, such that those with higher electricity demand, can trade energy instead of paying higher bills. Thus also providing a second income stream for prosumers for their production of the tradable renewable solar energy. We use Machine Learning to predict energy consumption and production. Based on these predictions, users can buy and sell renewable energy. To ensure maximum social welfare, we implement a Double Auction Mechanism (DAM) to optimally allocate tradable energy between prosumers where they are guaranteed a fiat reward. The transactions are stored on a Proof-of-Authority (PoA) Distributed Ledger. The back end is built using Go programming language due to its speed, easy syntax and its increasing popularity as a server side language. The Angular framework is used for the front end development with Typescript used for middleware service implementation. Energy forecasting and DAM are written using Python libraries. Moreover, the platform uses the MongoDB NoSQL cloud database to securely and reliably store data. The Locust stress testing tool is used for performance benchmarking.

Index Terms—Peer-to-Peer Trading, Solar Energy, Energy Forecasting, Double Auction, Blockchain, Frontend and Backend.

I. INTRODUCTION

AS access to electricity increases with rising number of appliances and electrical devices in households, it is only fair that the energy consumption for the average person throughout history has been increasing steadily [1], [2]. Grids in Malaysia are managed by Tenaga Nasional Berhad (TNB), with electricity flowing through main transmission lines from the generators all the way to distribution lines which in turn is forwarded to residential areas. Without a doubt, as the energy demand and consumption keeps increasing, the transmission lines will have to be upgraded eventually [3]. This will cost copious amounts of time and money. If nothing is done the transmission lines will become extremely congested and unable to fulfill energy demands. Another downfall of the energy sector in some developed countries is that houses with solar panels, do not utilize the generated surplus energy to its full potential. They do not benefit in any way from the surplus energy they generate as all of it is dissipated to the grid for free [4]. This reduces morale for the general public as there is no incentive to produce the extra energy. This demotivates them from renewable energy generation and usage.

We combat the aforementioned problems with our P2P energy trading web application, where one household producing electricity using solar panels can trade it with another, whilst

also guaranteeing prosumer rewards by trading their produced solar energy. We use energy forecasting algorithms to predict energy consumption and production of users. This is used to calculate the optimal energy that can be allocated to the prosumer (buyer) from other prosumers (bidders) using the DAM. The trading energy and fiat amount are stored as part of transactions in a block in a proof-of-authority distributed ledger. A portion of this fiat amount is also forwarded to TNB since the system relies on their grid network. So now, electricity is no longer pulled from the mains resulting the overall demand and stress on the transmission lines to be reduced [4], [5]. Consequently, massive infrastructure upgrade costs and resources can be avoided. Furthermore, an incentive is created for people to own solar panels and start trading on the P2P platform, thus increasing the usage of renewable energy leading to a cleaner and greener environment.

A. Organization of Paper

Rest of the paper is organized as follows. Section II elaborates the methods used to implement the solution and the justification behind them. Detailed description of the results from each method are illustrated in Section III. Section IV discusses the issues in the system and the conclusion is provided in Section V. User interface illustrations are shown in the Appendix.

II. METHODS

The workflow of the application is that first energy forecasting is done to allow users to make orders and bids. Then DAM is run for optimal energy allocation and finally the transaction is stored in the blockchain. This is discussed in detail in the sections below.

A. Energy Forecasting

1) *Why Energy Forecasting:* For our implemented web application, we are assuming that the solar energy a prosumer generates is stored in a battery and the energy balance is stored in a smart metre in their homes. Using the stream of energy data from the smart metre, we forecast the amount of energy consumed as well as produced for a given day. While making an order, prosumers are shown their energy consumption forecast as in Fig. 1 and the production forecast, Fig. 2, is shown while bidding.

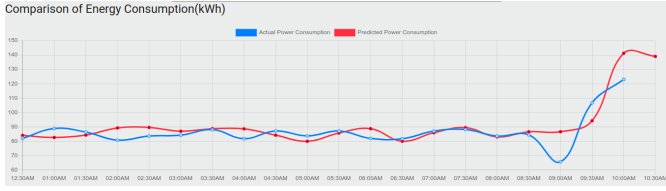


Fig. 1. Prosumer (buyer) energy consumption forecast for the next 30 minutes.

By predicting the energy consumption, we can limit the prosumer (buyer) to a max energy value that they can order. This ensures that extremely large energy amounts, that even the grid cannot produce, are never ordered. Predicting the energy production limits the prosumer (bidder) to a max tradable energy value they can use to make a bid on an energy order. This ensures, that no prosumer makes a bid that exceeds the max amount they are expected to produce.

2) *How Energy Forecasting is done:* The energy forecasting uses the machine learning technique of triple exponential smoothing [10] with additive trend and seasonality where the latest two data points are used for seasonal period [9]. The training also includes trend damping to prevent the rise of unrealistic trends and account for seasonal irregularity [10]. Once training is done using all the data points upto the current time of the day, a prediction is generated for energy consumption/production for the next 30 minutes from the current time. The prediction is the maximum energy forecast for consumption or production depending on what feature the prosumer is making use of. In terms of implementation, we used the python pandas library for data processing and statmodels for its exponentialSmoothing library.

B. Optimal Energy Allocation

Here we discuss optimal energy allocation among prosumers (bidders) to supply energy to other prosumers (buyers). For more details regarding the equations in this section, please see [5] in references.

1) *Social Welfare Maximisation (SWM):* The SWM problem is the objective function that we hope to maximise to fulfill the energy request of prosumers by also maximising the energy supply from bidders whilst considering the cost to generate the energy. The amount of energy that would satisfy a prosumer (buyer) based on their smart metre data and energy consumption forecast is defined by the satisfaction function in [5] as:

$$U_i(E_i^n) = w_i \left[\ln \left(n \sum_{j=1}^J (e_{ij}^n - e_i^{n,min}) + 1 \right) \right] \quad (1)$$

Next, we need to consider the cost incurred for the bidding prosumer when they trade the produced energy as per the production forecast. The cost function is given in [5] as:

$$L_i(S_j^n) = c_1 \sum_{i=1}^I (s_j^n)^2 + c_2 \sum_{i=1}^I (s_{ji}^n) \quad (2)$$

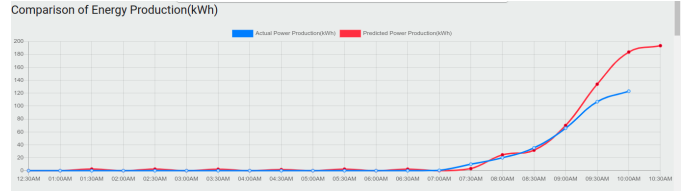


Fig. 2. Prosumer (bidder) energy production forecast for the next 30 minutes.

Now that we know how much energy would satisfy the prosumer as shown in Eq. (1) and the cost the bidder has to bear to produce and trade energy as in Eq. (2), we can determine the objective function for SWM as:

$$SWM : \max_{E_n, S_n} \sum_{i=1}^I U_i(E_i^n) - \sum_{j=1}^J L_j(S_j^n) \quad (3)$$

The SWM is subject to the following constraints.

$$\begin{aligned} \text{Subject to : } e_i^{n,min} &\leq \eta \sum_{j=1}^J e_{ij}^n \leq e_i^{n,max}, \forall i \in E, \\ \sum_{i=1}^I s_{ji}^n &\leq S_j^{n,max} \forall j \in Z, \\ \rho s_{ji}^n &= e_{ij}^n, \forall i \in E, \forall j \in Z, \\ e_{ij}^n &\geq 0, \forall i \in E, \forall j \in Z, . \end{aligned} \quad (4)$$

2) *Optimal Allocation Problem (OAP):* The OAP ensures no one prosumer can out bid everyone else, meaning all bidders are guaranteed to receive fiat reward as long as they bid on the order/request. Moreover TNB, as the grid providers, are also guaranteed to receive a reward since the system is designed to use their infrastructure to distribute the energy. The OAP is defined in [5] as :

$$OAP : \max_{E^n, S^n} \sum_{i=1}^I \sum_{j=1}^J [b_{ij}^n \ln e_{ij}^n - p_{ji}^n s_{ji}^n] \quad (5)$$

Both OAP and SWM have same subject to constraints as described in Eq. (4). Hence we carry out constraint relaxation through Lagrangian method [5] to find the local maximum [10]. So, energy receivable by a buyer, e_{ij}^n and energy tradable by a bidder, s_{ji}^n , after applying Lagrangian method is:

$$e_{ij}^n = \frac{b_{ij}^n [(n \sum_{j=1}^J e_{ij}^n - e_i^{n,min}) + 1]}{nw_i} \quad (6)$$

$$s_{ji}^n = 2c_1 s_{ji}^n + c_2 \quad (7)$$

where, c_1 and c_2 are cost factors for the bidder.

3) *Double Auction Mechanism (DAM):* The DAM solves the OAP via Lagrange method. As shown in Fig. 3, the application runs the DAM every 30 minutes taking all closed requests, looping through them and applying Eq. (6). and Eq. (7). on each bid.

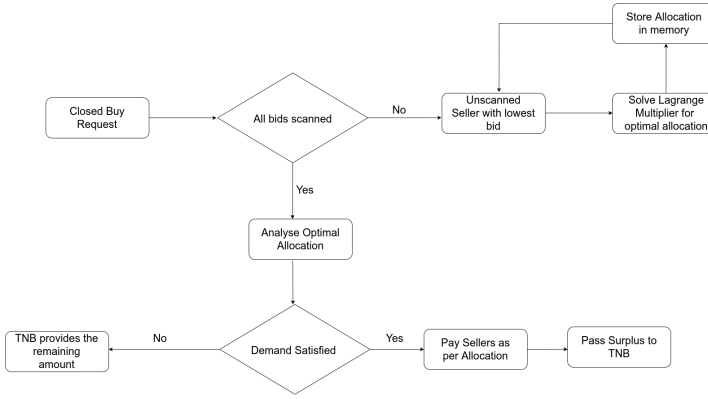


Fig. 3. Flowchart for double auction with demand and supply response.

The Lagrange scales down the bid energy amount by 10 to 15% on average to prevent one bidder to dominate. So, once all the bidders are considered, the system checks whether the total energy demand made by the prosumer in the order is satisfied or not [6]. If demand is satisfied, the remaining fiat amount that is not received by any bidder is transferred to TNB. If demand is not satisfied the surplus energy is taken from TNB [6]. The rate of energy is capped at RM 0.20 per kWh $\pm 20\%$ for all bidders as well as TNB. So, all bidders and TNB receive fiat currency as reward and buyer only pays for the amount of energy they need which is always fulfilled.

C. Blockchain

As shown in Fig. 4, the web application uses a Proof-of-Authority consensus mechanism [7], [8] for the blockchain.

1) Roles in the Blockchain:

- **Validator:** A validator can verify transactions, create new blocks, discard and add blocks. Since we are using a Proof-of-Authority consensus mechanism, the validators undergo a rigorous registration process where they need to reveal their identities [7]. Their reputation is at stake which means if they go against the interest of the prosumers, then their status as validators will be revoked and made known to the greater community [8].

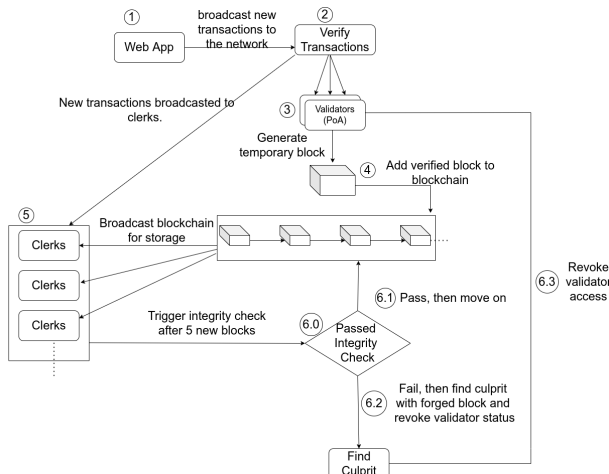


Fig. 4. PoA blockchain with integrity checks via clerks.

- **Clerk:** Clerks provide integrity check on the blockchain to ensure that validators do not conspire against the community. After the addition of every new block, they receive an updated local copy of the blockchain and user accounts which is used later to check the central one. Any normal node can be made a clerk and they do not need to be rigorously identified. Clerks also provide the benefit of recovering the blockchain in case a validator account is compromised since they have a 51% agreement on the valid blockchain.

2) Blockchain pipeline:

- **Signing Smart Contracts:** After the DAM is run every 30 minutes, the new pool of transactions are broadcasted over the network to all the validators. Those that receives the transaction pool first, will verify each transaction. They check whether the buyer has sufficient balance or not. If so, the transaction is marked as verified and made part of a temporary block. If a buyer has insufficient balance then the transaction is marked invalid. Once all the transactions have been checked, and added to the temporary block, they are broadcasted to all the remaining validators. These validators use their local copy of user accounts to verify each transaction in the temporary block. They then use the nonce of the temporary block to hash the transactions. If this hash matches with that of the temporary block for all validators, then the temporary one is added permanently to the blockchain. The last validator who checks the hash finalizes/signs the smart contract.
- **Discarding Blocks:** If there is a validator who does not find a match for the hash, then their local copies of user accounts and blockchain is updated. Then the check is done again. If the hash fails to match a second time, then that block is discarded permanently. The non-match signifies that a transaction was manipulated in the central blockchain so the discarding is justified.
- **Integrity Check:** After the formation of 5 new blocks, an integrity check is triggered where the clerks verify each transaction in the latest permanent block in the central blockchain using their local copy of user accounts. Next, they use the nonce of the latest block from central blockchain to hash the transactions from the latest block in their local blockchain. If this hash matches that in the central blockchain for more than 50 percent of the clerks, then there is no issue but if the match is less than that, then an integrity check is issued. This goes through the local blockchain copy of each validator and compares the hash of the latest block. The validator(s) whose hash has a mismatch is then flagged since they may have tempered the central blockchain. In this case the validator access may be revoked thus safeguarding public interest.

III. RESULTS

We now discuss the results that we got from trying to fulfill the project promises in terms of social welfare by generating income, suitability of the application design and performance of the application and blockchain. The income generated by prosumers and TNB are shown in Table. 1.

TABLE I

Potential Prosumer income		
Date	Sold Energy (kWh)	Income (RM)
20-04-2022	53.45	15.00
24-04-2022	52.55	14.80
30-04-2022	23.55	7.55
Potential TNB income		
Date	Sold Energy (kWh)	Income (RM)
20-04-2022	125.33	30.45
21-04-2022	247.54	52.45
23-04-2022	165.63	42.33
24-04-2022	325.67	77.82
30-04-2022	220.05	50.05

A. Energy Trading As A Source Of Income

As mentioned earlier, the DAM ensures that all bidders make some fiat income as a reward as long as they are bidding on buyer requests. This serves as an incentive for the public to invest in equipment to generate electricity using solar panels. Since TNB manages the grid infrastructure for us, the application ensures that they receive some fiat reward as well for their participation in the trading process. The TNB account is an admin account. They have control over all aspects of the blockchain and can add new validators and clerks. As admin, they can also access the main central database. No matter what bids are made on an energy request, TNB always provides a fraction of the energy demand and hence this income is always guaranteed. If a request has no bids, then the full demand is fulfilled by TNB where they receive the full price of the energy. Income charts are also available for all users - validators, clerks, normal and admin, TNB, from their profile page in the application.

B. Blockchain Management

The blockchain exists in two formats. One of them is the centralised one, stored in the database and the other is a local copy stored in the validators' and clerks' device as a .JSON file.

1) *Blockchain on the frontend of web application:* As shown in Fig. 5, the central blockchain is available to all users from the blockchain page of the web application which shows the index, hash, nonce and the previous hash for each block. In addition to these, when the info icon is clicked, users can see some general information regarding the transactions in each block: the buyer, total energy and fiat traded and the income TNB generates from this transaction. This is shown in Fig. 6.

Index	Hash	Previous Hash	Nonce
0	00366b81436deffad74e7c50961abc81858a15e319676287da522724e33		6
1	08f0cbac423e303e50f03f84f5ba1ee2d5d0c1dcf4734355fa7cd55daa1353a5	00366b81436deffad74e7c50961abc81858a15e319676287da522724e33	5
2	05b4d60f384a4925aa9a51cfff4d3b4316afa6d544e6b47795892b59313615862	08f0cbac423e303e50f03f84f5ba1ee2d5d0c1dcf4734355fa7cd55daa1353a5	9
3	0b0ba7519fc3208cf682f68b3f1db26a01891b27ce66e1c6e2d9afce383a022	05b4d60f384a4925aa9a51cfff4d3b4316afa6d544e6b47795892b59313615862	1
4	0426dc5f6376c517ca006fec1ae102ebba7764656623629474f78c02e1	0b0ba7519fc3208cf682f68b3f1db26a01891b27ce66e1c6e2d9afce383a022	15

Fig. 5. Index, Hash, Previous Hash and Nonce on frontend.

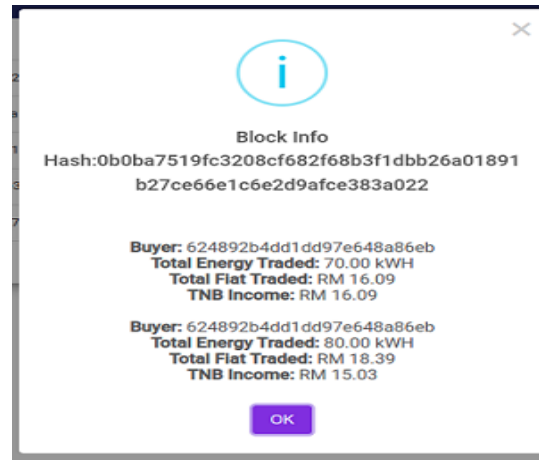


Fig. 6. Transaction information in the block.

2) *Blockchain on the backend of web application:* As mentioned before, the backend stores the blockchain in a central database as shown in Fig. 7. This is used as the primary blockchain throughout the system. The frontend uses the data available in this blockchain.

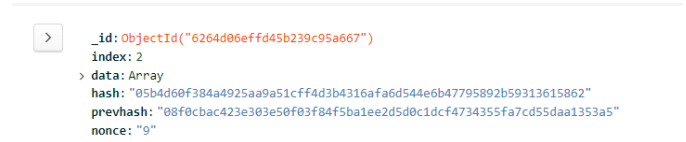


Fig. 7. Blockchain as stored centrally using mongodb.

The other blockchain format is a local copy stored in the validators' and clerks' device as a .JSON file and this is shown in Fig. 8. As discussed before, the application uses this local copy to verify transactions and check the hash in the central database before finalizing the blocks. The app always updates the local copy using the central one when the verification and integrity checks are done. This means there is no chance for clerks and validators to edit the local copy and manipulate the central blockchain.



Fig. 8. Snippet of a block stored in the local blockchain file.

C. Web Application Testing

1) *Load Testing:* When the platform is eventually released to the public, it has to serve lots of users concurrently without any issues. We used a swarming tool called Locust to simulate such scenarios. The load testing simulates 100 normal users concurrently using the system. Each user first

visits the homepage, creates an order, makes a bid, visits the marketplace and finally the blockchain page. Such sequence ensures real user behaviour when the system is deployed. Fig. 9 and Fig. 10 respectively show the system performance when handling concurrent user requests and system responses. From both graphs, we can see that there are no failures meaning the system works as intended under normal loads.

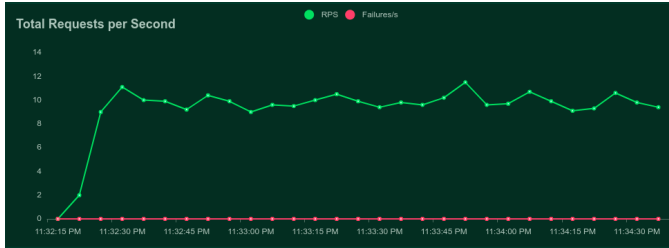


Fig. 9. Handling concurrent requests on backend servers.

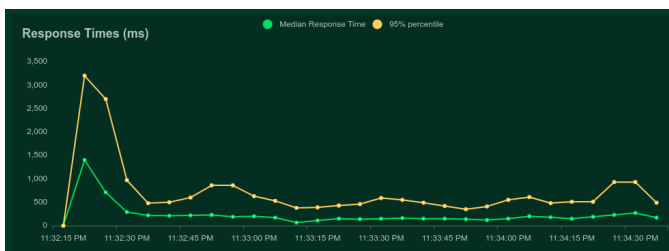


Fig. 10. System responding to concurrent user requests.

2) *Stress Testing*: Stress testing is done to find out at what capacity the system fails. This will help us determine which upgraded database version to use during deployment since the current one is using the free cluster service from mongodb. It also shows us how gracefully the system fails. The results from stress testing are shown in Fig. 11 and Fig. 12 respectively. In both cases the request failures and response time started to increase when we hit the 1,000 user threshold.

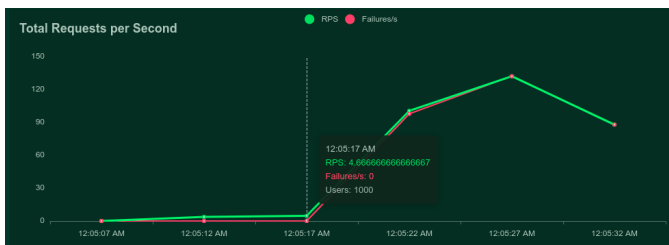


Fig. 11. Request handling by backend servers under high stress.

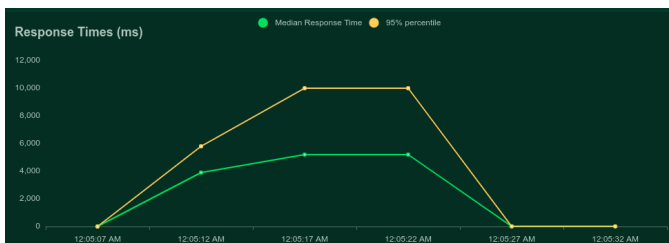


Fig. 12. Response handling by backend servers under high stress.

3) *Blockchain Performance*: As the web application is still in prototype phase, the hashing of blocks were kept to the lowest difficulty, that is each block is considered hashed, when there is a nonce that returns one zero at the very start of the hash string. With this level of difficulty, we tested the block hashing time with increasing transaction size as shown in Fig. 13. Since the hashing algorithm only increments the nonce and checks for zero character at the start of the hash string, there is no specific observation. However, in the real world scenario, as block size increases with increasing transaction sizes, the difficulty would need to be increased and hence the hashing time will increase as well. With a load test of 100 users, we observed that blocks with 50 transactions can be generated as fast as possible while accepting a lot of requests without failing. So, every block can store up to 50 transactions.

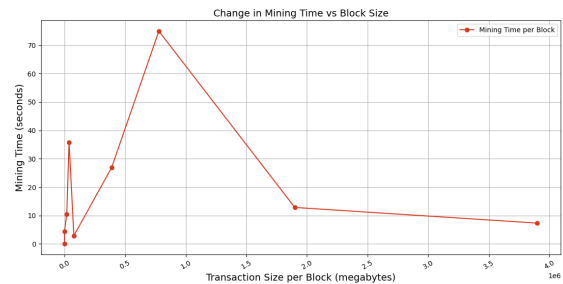


Fig. 13. Time taken to mine a block with increasing block size.

D. Source Code

The web application is developed over a period of 6 months from December 2021 to May 2022. The code is made available at: https://gitlab.com/mohammad_rafaquat_alam/imdc_p2p_energy_trading.

IV. DISCUSSION

The system hashes and stores the private and public keys of all the users in the database. Even though the hash is not reversible, this is not good software development practice. Attackers can launch a dictionary attack with a known hash table and therefore obtain the real private key. So, further work needs to be done to write a mechanism to store and extract the private key of users from the user devices as opposed to the database. As shown in the results section, the free mongodb database clusters that are used are not suitable for large scale use. The system crashes when 1,000 users are using the system concurrently. To fix this, funding is required to upgrade to a better and reliable mongodb service. The underlying database code would remain the same so this fix is very simple.

V. CONCLUSION

Based on all the previous sections, it is clear that the blockchain technology is here to stay and keep penetrating even more industries. P2P optimal solar energy trading has a huge potential to make ripples through the energy industry. In conclusion, team MY157 have successfully implemented a peer to peer optimal solar energy trading platform using blockchain technology.

VI. APPENDIX

This section illustrates the user interface of the web application in context to the previous sections.

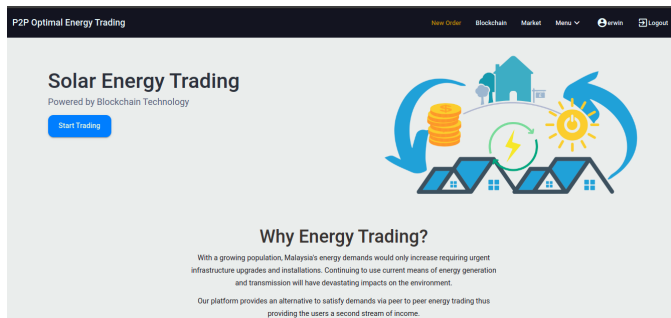


Fig. 14. How the user is greeted to the site.

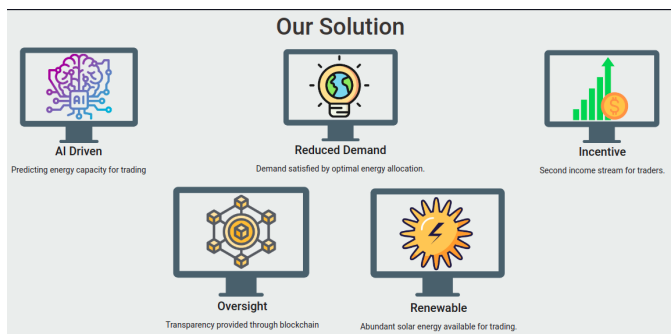


Fig. 15. Key features of the web app.

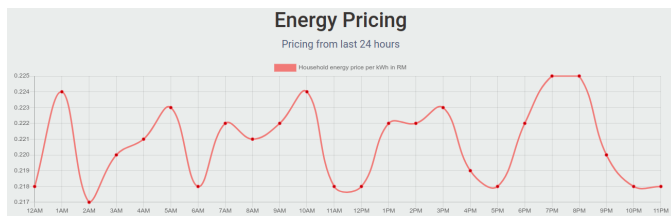


Fig. 16. Pricing chart for per kWh of energy.

Fig. 17. Form to make an energy order.

Buyer	Energy Amount (kWh)	Fiat Amount (RM)	Request Id	Elapsed Time	Make a bid
(raf) 6235eada93c9762da8b7e54	145	33.40043478260869	62ced7be2007b942992edf	5 Min 19 S	Make a bid

Fig. 18. The open energy request for bidding.

Fig. 19. Form to make a bid on the open energy request.

Buyer	Energy Amount (kWh)	Fiat Amount (RM)	Request Id	Request Status
(WZH) 6288476c510a2d46ddc38ab0	5	1.1515217391304347	628d23185f1894d2f7019757	Closed
(raf) 6235eada93c9762da8b7e54	120	27.584347826086965	62886d26cc751aa60de31a9b	Closed
(WZH) 6288476c510a2d46ddc38ab0	59	13.56230434782609	62886de73cc006b726414d4	Closed
(WZH) 6288476c510a2d46ddc38ab0	33	7.585955652173915	6288a1537ce51a250344656	Closed
(James) 6288a1877ce51a250344657	1	0.23000000000000004	628b4499d4311e0086342c67	Closed

Fig. 20. List of all closed requests.

VII. ACKNOWLEDGEMENT

The team is immensely grateful for the efforts of our supervisor Dr. Tan Wen Shan, for believing in our capabilities and for supporting and trusting our decision making, until the very end. We would also like to thank Kareem Osama Mahmoud Elsayed Ahmed for mentoring the team on Angular and Golang as well as Wong Zheng Hao and Roland Chua Siew Ong for providing feedback for the web app. Lastly, we would like to thank IMDC for giving us the opportunity to compete and build this project.

REFERENCES

- [1] G. "U.S. energy facts explained - consumption and production - U.S. Energy Information Administration (EIA)", Eia.gov, 2021.
- [2] H. Ritchie and M. Roser, "Energy", Our World in Data, 2021.
- [3] "Outline of Malaysia's electricity system - Power Engineering International", Power Engineering International, 2021.
- [4] J. Yang, A. Paudel, H. B. Gooi, and H. D. Nguyen, "A proof-of-stake public blockchain based pricing scheme for peer-to-peer energy trading," Applied Energy, vol. 298, p. 117154, 2021.
- [5] S. Aggarwal and N. Kumar, "A consortium blockchain-based energy trading for demand response management in vehicle-to-grid," IEEE Transactions on Vehicular Technology, vol. 70, no. 9, pp. 9480-9494, 2021.
- [6] H. Haggi and W. Sun, "Multi-Round Double Auction-Enabled Peer-to-Peer Energy Exchange in Active Distribution Networks", IEEE Transactions on Smart Grid, Vol. 12, no. 5, 4403
- [7] "Proof of Authority Explained", Binance Academy, 2020
- [8] "Proof of Authority: consensus model with Identity at Stake", POA Network
- [9] "Time-series Forecasting - Complete Tutorial", Analytics Vidhya
- [10] "A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python", Machine Learning Mastery
- [11] "Lagrange Multiplier Approach with Inequality Constraints", Machine Learning Mastery