



A Decentralized Web Application for Parking Spot Booking and Listing: Improving Convenience and Efficiency in Urban Areas

Sanket Shigaonkar¹, Abhishek Vetal^{1*}, Pranav Yeolekar¹, Suchita Walke²

Abstract

Finding a parking spot is becoming a frustrating task, especially during busy hours and with the increasing number of vehicles on the road. In the existing system, people have to manually drive and search for any vacant parking spot and if any spot is not available, they end up parking in the wrong zone. Not only is the time and fuel wasted on this task, but many people also end up paying fines for parking in no-parking zones. To solve this problem, we have proposed “ParkSpace”, which is a decentralized web application that connects people who want to rent their parking spot and drivers or car owners who are looking for a safe place to park. The platform provides individuals with a simple and low stress means to generate income from their property. The transaction between the entities will take place using cryptocurrency. Thus, transparency in the system is maintained and accessible to everyone free of cost. The main advantage of the application is that it saves time to find a spot by navigating the user to the nearest location where a spot is available, and all the transactions are stored on the blockchain network.

Keywords: Blockchain, parking, decentralized, Web3, dApp

INTRODUCTION

The process of urbanization has led to a significant challenge for citizens, as parking spaces in cities are limited. Constructing multi-level parking infrastructures is a possible solution to increase the number of parked cars in a given space. However, this approach requires a considerable investment in infrastructure and maintenance.

Motivation

*Author for Correspondence

Abhishek Vetal
E-mail: abhishekvive19compeng@student.mes.ac.in

¹Student, Department of Computer Engineering, Pillai HOC College of Engineering and Technology, Rasayani, Maharashtra, India

²Professor, Department of Computer Engineering, Pillai HOC College of Engineering and Technology, Rasayani, Maharashtra, India

Received Date: April 17, 2023

Accepted Date: April 24, 2023

Published Date: April 28, 2023

Citation: Sanket Shigaonkar, Abhishek Vetal, Pranav Yeolekar, Suchita Walke. A Decentralized Web Application for Parking Spot Booking and Listing: Improving Convenience and Efficiency in Urban Areas. Journal of Web Engineering & Technology. 2023; 10(1): 7–16p.

Alternatively, encouraging the use of public transportation such as buses, railcars, and subways can address the issue of parking shortages. Nevertheless, public transportation may not be adequately developed or popular, particularly in suburban areas of developing countries. By creating smarter parking lots that are user-friendly, individuals can save time and money, minimize gas wastage, and reduce emissions [1, 2]. Parking in city areas remains an ongoing struggle due to space constraints, leading individuals to travel significant distances in search of parking spots. Additionally, increasing residential apartments necessitate additional parking. In crowded residential areas, guest parking is difficult or unavailable, leading to roadside parking that further congests roads.

Parking demand is high on weekends near shopping malls, beaches, and parks, and on weekdays near official building premises for employees and clients. Such congestion leads to unnecessary time and money loss while searching for parking. Providing sufficient parking spaces can mitigate traffic congestion in cities. The introduction of smart parking solutions can make it easier for the public to park their vehicles securely within the vicinity to meet their daily needs.

Background

ParkSpace is a peer-to-peer decentralized web application that connects people who want to rent their parking spot and drivers or car owners who are looking for a safe place to park. With the platform, property owners can generate income from their property in a simple, low-stress manner. The transaction between the entities will take place using cryptocurrency. Thus, transparency of the system is maintained and accessible to everyone free of cost. ParkSpace is a game-changing platform that addresses the problem of parking scarcity in urban areas. As the number of vehicles on the road increases and parking spaces become less available, finding a safe spot can become increasingly difficult [3]. This is where ParkSpace comes in, providing a hassle-free solution for both parking spot owners and drivers. By leveraging blockchain technology, ParkSpace offers a secure and decentralized platform for parking transactions. The use of cryptocurrency ensures transparency, security, and anonymity in the transaction process, making it easier for users to trust the platform [4]. Additionally, the platform provides a way for people to monetize their unused parking spots, which can be a lucrative source of income. Moreover, ParkSpace is not just a service that benefits individuals, but can also have a positive impact on the environment. Urban air pollution and traffic congestion can be reduced through the platform by reducing the time drivers spend looking for parking spots. This makes it a win-win situation for both parking spot owners and drivers. Overall, ParkSpace is an innovative solution to the parking problem that benefits all parties involved. With its use of blockchain technology and cryptocurrency, the platform offers a secure, transparent, and easy-to-use platform for parking transactions [4]. It is a step towards a smarter, greener, and more efficient urban environment [5].

LITERATURE SURVEY

An Overview of the Procedure

Building a P2P Marketplace

Designing a peer-to-peer marketplace requires a systematic approach to ensure that the platform meets the needs of both buyers and sellers. Below are some core methods that can help in the process [6]:

- *Identifying the target market:* The initial step in developing a peer-to-peer marketplace is to identify the target market. This involves gathering information on the demographics, preferences, and behaviors of potential buyers and sellers. Utilizing this data will assist in creating a user interface that is tailored to their specific requirements.
- *Conducting market research:* Research is essential to understand the market and developing a successful platform. It involves analyzing competitor platforms, identifying gaps in the market, and determining the features that are most important to users.
- *Building a strong brand:* A strong brand is critical in building trust and credibility among users. This includes creating a unique brand identity, establishing a consistent tone of voice, and developing a strong visual identity.
- *Developing a user-friendly interface:* The user interface is a crucial aspect of a peer-to-peer marketplace, as it directly impacts user experience. The design should be intuitive and easy to navigate, with clear categories and search filters to help users find what they are looking for quickly.
- *Providing robust security:* Security is paramount in a peer-to-peer marketplace. Users need to feel secure when using the platform, knowing that their personal and financial information is protected. Implementing encryption, two-factor authentication, and other security measures can help prevent fraud and maintain user confidence.
- *Implementing a transparent payment system:* Payment processing is a critical component of any peer-to-peer marketplace. It is essential to provide a transparent and reliable payment system that

protects both buyers and sellers. This can be achieved by integrating secure payment gateways and using escrow services to ensure that funds are released only when the transaction is complete.

- *Offering customer support:* Customer support is crucial in ensuring that users have a positive experience on the platform. Providing multiple channels of communication and a responsive customer support team can help address user concerns and build trust.

Technical Implementation

- *Choose a blockchain platform:* The first step is to choose a blockchain platform that supports the creation of DApps. Some popular platforms include Ethereum, EOS, and TRON.
- *Develop smart contracts:* Self-executing programs that are stored on the blockchain are known as smart contracts. They automate the execution of transactions and help to ensure the security and transparency of the DApp. Smart contracts can be developed using Solidity, a programming language used on the Ethereum blockchain.
- *Design the user interface:* The user interface is designed using standard web development tools like HTML, CSS, and JavaScript. The interface is responsible for presenting the information to the user and allowing them to interact with the DApp.
- *Implement the backend:* The backend of a DApp consists of a node that interacts with the blockchain. The node is responsible for executing smart contracts and storing data on the blockchain. This can be done using a node.js library like Web3.js.
- *Deploy the DApp:* The DApp is deployed on the blockchain using a smart contract. This is done using a blockchain-specific integrated development environment (IDE) like Remix.
- *Test and launch the DApp:* The DApp is tested for bugs and errors before being launched on the blockchain. Once launched, users can interact with the DApp using a compatible wallet like Metamask.

EXISTING SYSTEMS

The parking management system is a product or software that has been specifically designed to assist the parking business in managing its operations more efficiently [7]. It focuses on aiding individuals with their parking difficulties and incorporates several features that make it an excellent choice for managing parking spots. For example, the system can integrate with and control automated gates for entry and exit, and it also includes an automated ticketing system. When all these features are combined with other capabilities, such as statistical reporting, the parking management system can significantly increase the effectiveness of a company's parking lot.

Following were the drawbacks of the system:

- The expense associated with construction or installation is one of the main disadvantages of the parking management system. This is because the system requires various components to function, such as automated ticketing, statistical reporting, and other features, which can increase the overall cost significantly [8]. As a result, some organizations may not be able to afford the expense of implementing the system.
- Maintaining the parking management system is essential to ensure optimal functionality, despite its automated nature. Regular maintenance is required to identify and address any issues that may arise, ensuring the system is working seamlessly. Maintenance intervals are typically scheduled on a monthly basis to ensure any potential problems are identified early and resolved promptly [9].
- The parking management system may pose operational challenges for some individuals, as it may be unfamiliar to them. This lack of familiarity could result in difficulty utilizing the system, causing complications during parking [10].
- Like any machine, the parking management system is susceptible to breakdowns, which could result in vehicles being unable to access buildings or parked cars being unable to move. Additionally, malfunctions in the system could lead to cars being parked in the wrong place.

METHODOLOGY

Proposed System

The overview of a web application provides a visual representation of the entire constructed application (Figure 1). When a user first accesses the website, they will be presented with the Homepage, which will showcase various sections containing a Search Bar, which can take input from the user; nearby available spots, an option to list a parking spot and an option to find and book any user choice parking spot. Proceeding further, the user will see the various details about the spot like cost per hour, occupancy details and images of the spot [11]. Users will be prompted to connect to a cryptocurrency wallet. Connecting to the wallet will act as a login for the user's account. After that the user will make the payment using his wallet and after successful payment, the smart contract will get triggered and update the blockchain. User will now see his/her spot is booked successfully. All this is done via different technologies i.e., for UI for the user, React JS and Web3JS are used and for smart contracts we use an Ethereum Blockchain Platform. All the data of the user will be stored on the blocks and will be updated [12].

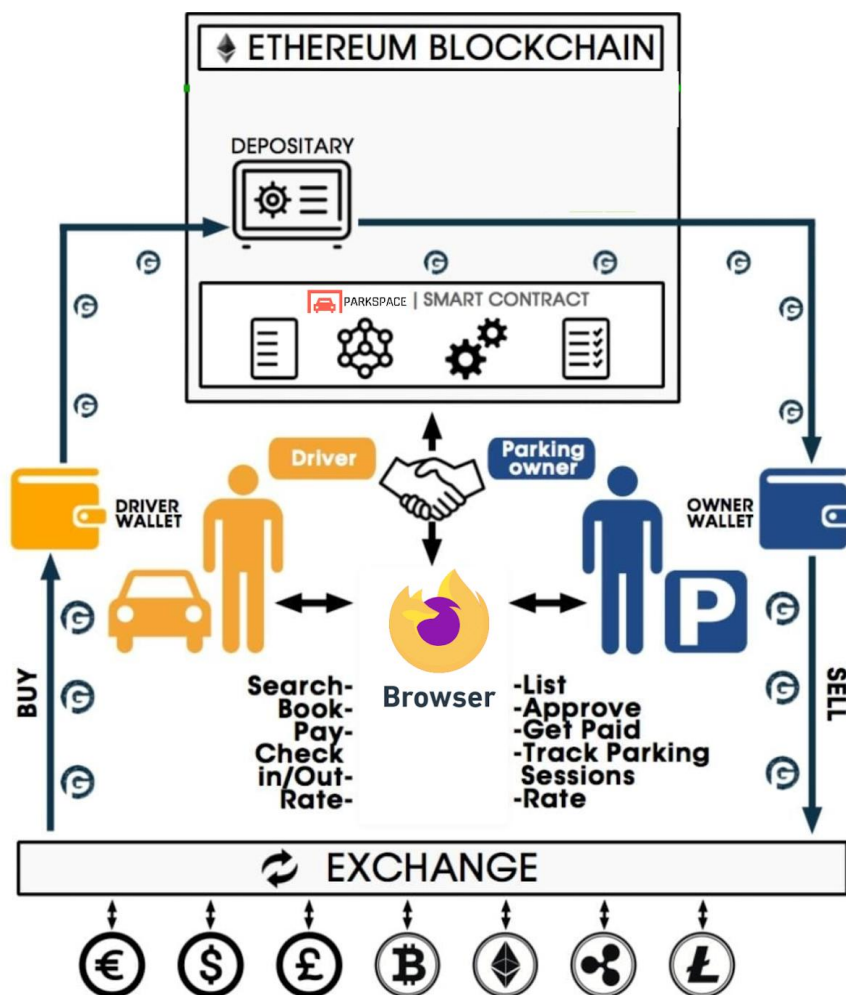


Figure 1. Overview of application.

Ethereum Blockchain

It is a decentralized platform that enables a peer-to-peer network to execute and verify smart contracts, which are applications with self-executing code. It allows participants to transact with each other without relying on a trusted central authority. The records of these transactions are secure, immutable, and distributed across the network, providing full transparency and ownership to

participants. Users create Ethereum accounts to send and receive transactions, which require a signature and a payment in Ether, the platform's native cryptocurrency, for processing on the network.

Smart Contracts

Smart contracts operate through code written on a blockchain, which contains conditional statements such as "if/when...then...". When these pre-established conditions are met and confirmed, a group of computers executes the corresponding actions, which can involve tasks like sending notifications, releasing funds, registering vehicles, or issuing tickets. Once the transaction is completed, the blockchain is updated with the results, which cannot be altered. Only authorized parties have access to view the results.

Cryptocurrency Wallet

Crypto wallets differ from regular wallets in that they do not store cryptocurrency but rather, the holdings exist on the blockchain and are only accessible using a private key. This key acts as proof of ownership and enables transactions [13]. Losing the private key means losing access to the funds. Therefore, it is crucial to keep a hardware wallet secure or utilize a reputable wallet provider such as Coinbase.

Front End:

For front-end, we have used the following technologies/languages:

- **HTML:** The Hyper Text Markup Language, or we can also say HTML, is a standard markup language. HTML is generally used to create the structure of the webpage. If we need to display the content on the web pages, HTML is the initial language that we have to use.
- **CSS:** Cascading Style Sheets is a style sheet language used for designing the web page and also used to describe the position of the elements of a document which is written in a markup language such as HTML or XML.
- **JavaScript:** JavaScript abbreviated as JS, is a programming language that is used to provide the functionalities to the web page, and it is generally used to create the more interactive websites which are able to handle the complex tasks or functions. 98% of websites use JavaScript on the client side to make the website more interactive.

Back End

The backend, which is also called a database is the part that is going to be connected to the front of the website. The backend is the part which is going to remain on the Ethereum network which is used to provide certain functions [14]. The backend is created using smart contracts which uses programming languages like solidity and also Serpent which is similar to python. Because of this code, the blockchain is able to provide the multiple characteristics like transparency, immutability, and distributed and decentralized system.

Ethereum Client: Ethereum is supported by a variety of client applications developed in languages such as C++, Go, Python, Haskell, and more. Having multiple implementations is important for the network's stability, as each one offers its own advantages. Once a client is synchronized with the blockchain and configured, it can be used for creating wallets and conducting transactions with Ether.

Interactive Console: To facilitate debugging, node configuration, and wallet usage via command line, installing the Go Ethereum JavaScript Console is the most straightforward method to obtain an interactive console. This allows for more dynamic changes to be made to the script.

Test Net: Developers may need to spend a significant amount of Ether to run smart contracts on the Ethereum main network. As an alternative, developers and learners typically use a local test network to avoid the need to download the entire blockchain and pay high fees to test their code. Setting up a test network using Geth is a simple solution as shown in Figure 2 [15].

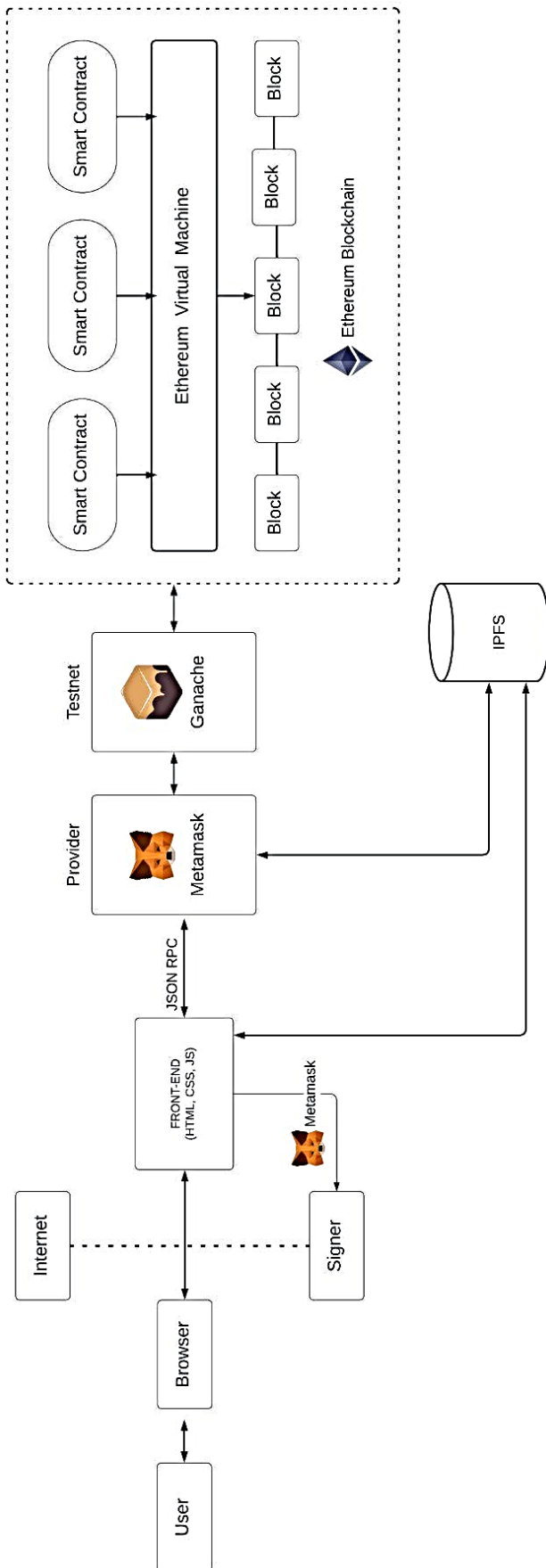


Figure 2. System architecture of the application.

System Architecture

When a user accesses the web3 application, they will be presented with an interface created by Next.js, which acts as a web client. Next.js is a free and open-source development framework that is built on top of Node.js, providing web applications with React-based functionalities like server-side rendering and generating static websites. The workflow of the system is explained below:

1. User: There are two types of users in this application. One is the user who wants to list his parking spot for rental and others who want to book a spot for parking their vehicles. The user first connects his/her Metamask wallet, which has to have the sufficient balance needed for the transaction.
2. Then the user selects the date and time as per his needs and proceeds further on the rentals page. The user is then shown available spots to pick, then he selects a spot and can see the images of the spot and also the details about cost and occupancy.
3. After selecting the spot, he is then prompted for payment and after successful completion of payment using cryptocurrency, the blockchain gets updated via the smart contracts and finally the user is shown that his spot is booked successfully.
4. For the users wanting to list their spots they have to proceed step-by-step uploading the required documents, images, and the cost of renting their property.
5. Following all the steps, the user's property gets listed and updated on the application as well as the blockchain. User will get the amount for his property when somebody books his/her spot for parking, and thus the complete cycle of payment from the customer to the owner of the spot is completed.

RESULTS

The homepage consists of different components which are the connect wallet button which allows the user to login to the application; by-clicking it will open the MetaMask wallet and prompt access to the website as shown in Figure 3. After successfully connecting the wallet, the user can now search for spots nearby by selecting his preferred location and check-in check-out details as shown in Figure 4.

This page shows all the available parking spots nearby and the location on the map following the information taken from the homepage, users can book their preferred spot and pay the charges using the MetaMask wallet as shown Figure 5.

The Dashboard shows all the user information i.e., the booking history and also any spot they have listed any parking spot as shown in Figure 6.

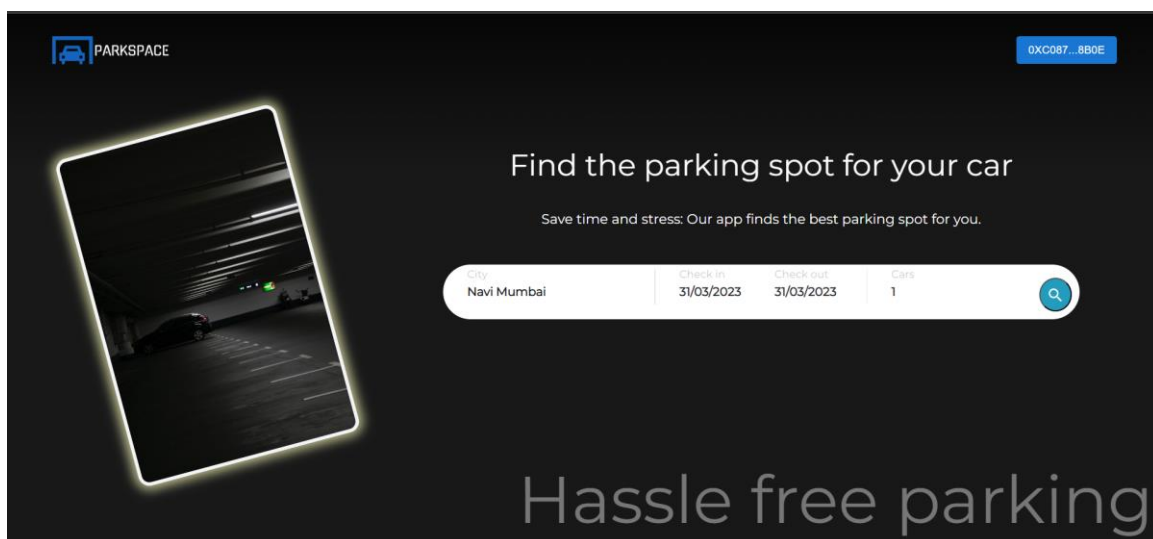


Figure 3. Homepage.

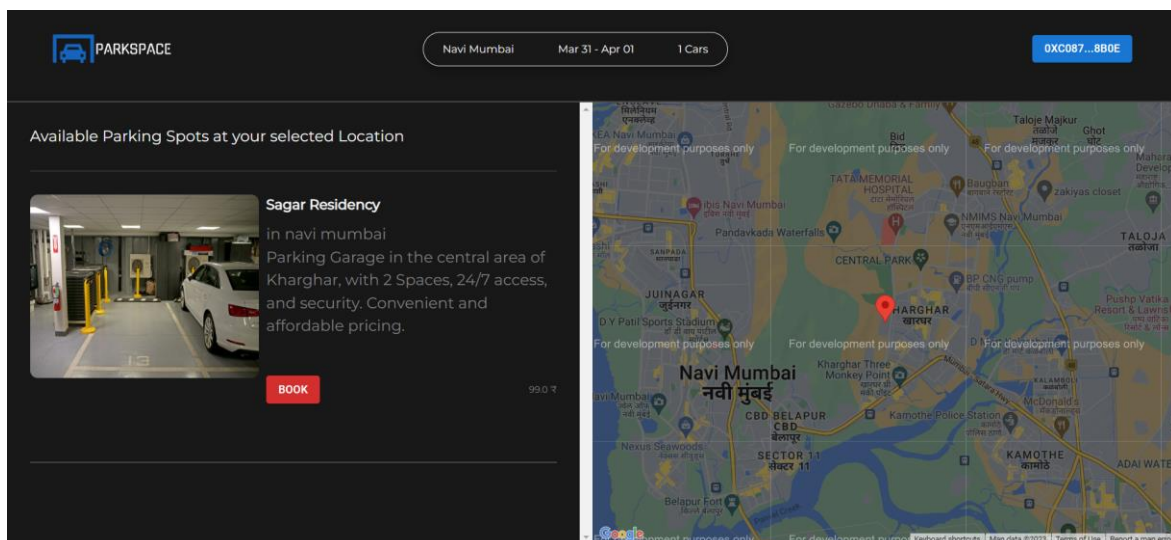


Figure 4. Booking Page.

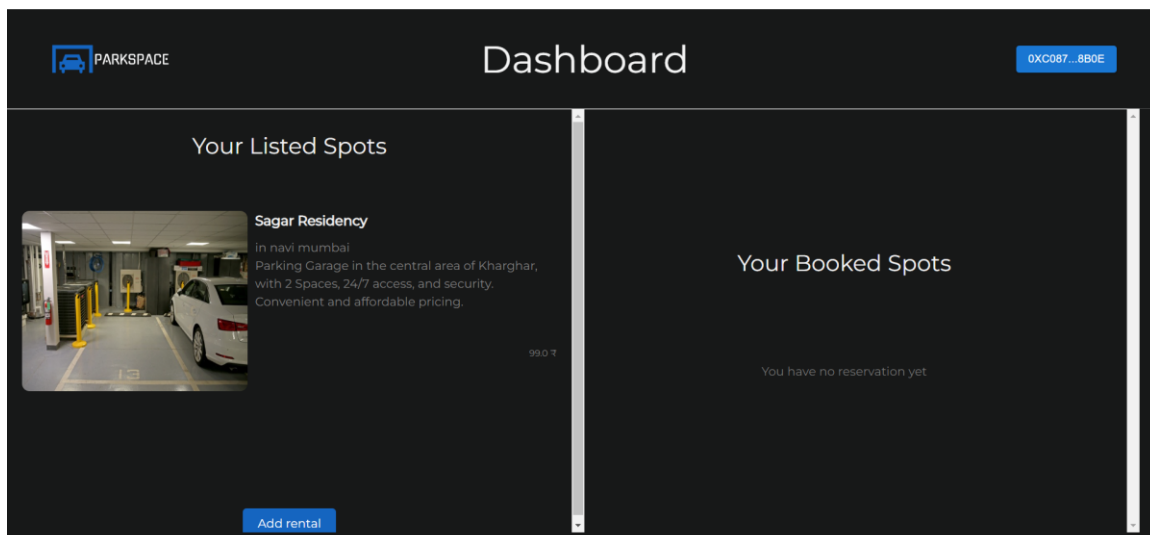


Figure 5. Dashboard (After listing user spot).

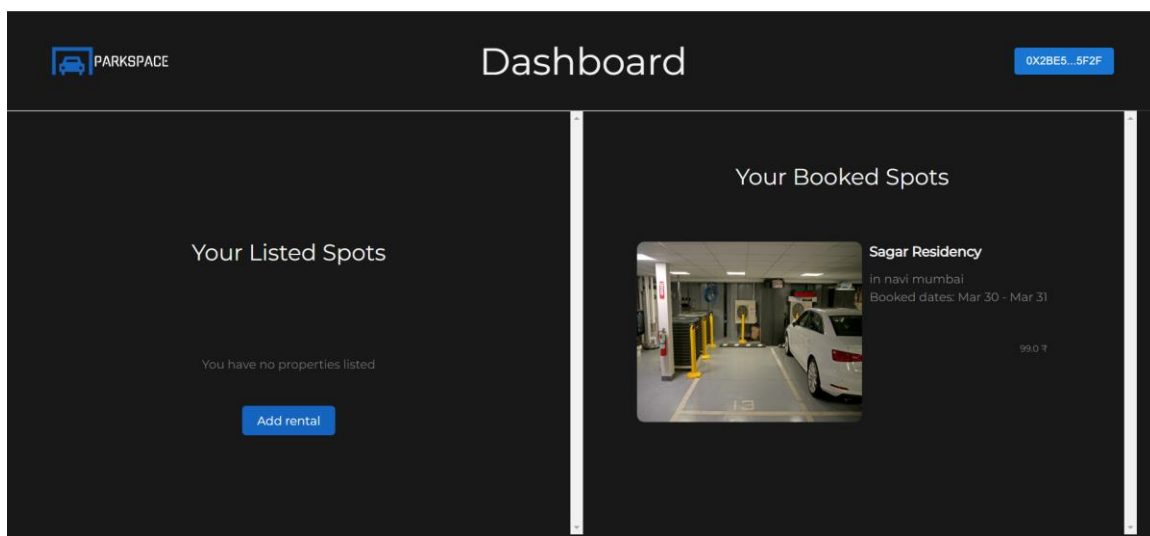


Figure 6. Dashboard (After booking view).

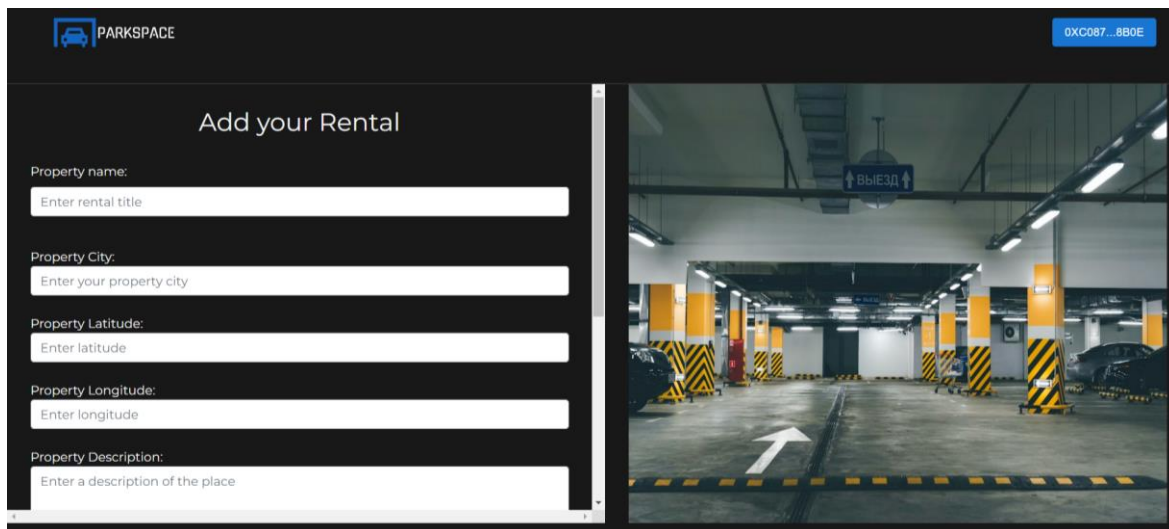


Figure 7. Add listing page.

The add listing page takes in input from the user about the parking spot he/she is listing for rent. Information like address, map coordinates, description, and image of the property. After filling in all the information, to add the listing certain gas fees are charged and confirmation of the same is done via MetaMask as shown in Figure 7.

CONCLUSION

In this project, the proposed system decreases the time, cost, and difficulty of finding parking spaces in urban areas. Traffic congestion can be reduced. Private parking lot owners can make a profit from renting out available parking spaces. Transparency and integrity are maintained. The parking spot is also being utilized for making profits instead of just remaining a depreciating asset. The navigation to the nearest parking is helpful in reducing time for finding the spot and fuel consumption.

In conclusion, the parking spot booking and listing service aims to solve the issue of the shortage of available parking spots and the inconvenience of finding one in busy urban areas. The platform provides a solution that is beneficial to both drivers and property owners, making it easier for drivers to find and book a parking spot, and providing an additional source of income for property owners by allowing them to rent out their unused parking spaces.

Our platform will be user-friendly, secure, and accessible through multiple devices, ensuring a seamless experience for users. We believe that our solution will not only help to ease the shortage of parking spots, but also improve the overall parking experience for drivers, making it less of a hassle and more convenient.

In summary, the parking spot booking and listing service is a practical solution that has the potential to make a significant impact in busy urban areas. By providing a platform for drivers to find and book parking spots and for property owners to rent out their unused spaces, we aim to improve the overall parking experience and provide a solution that benefits everyone involved.

ACKNOWLEDGEMENT

Collaboration among team members is crucial for the successful completion of a technically demanding project such as this. We express our heartfelt appreciation to those who played a pivotal role in bringing this project to fruition. Our external guide, internal guide, and Supervisor, Ms. Suchita Walke, as well as our Project Coordinator, Ms. Rajashree Gadhave, provided valuable inputs on documentation and project timeline. Additionally, we extend our gratitude to the Head of Department, Ms. Rohini Bhosale, for granting us the privilege of being a part of this remarkable journey.

REFERENCES

1. INRIX. (2017). Searching for Parking Costs Americans \$73 Billion a Year. [Online]. Inrix. Available from: <https://inrix.com/press-releases/parking-pain-us/>
2. Li T, Song S, Yang Y. Driving restrictions, traffic speeds and carbon emissions: Evidence from high-frequency data. *China Econ Rev.* 2022 Aug 1; 74: 101811.
3. Zhang MY, Zhou LP, Cai WY. Smart Parking System Based on Wireless Sensor Network for Large Parking Lots. *Applied Mechanics and Materials (AMM).* 2012; 195–196: 889–894. <https://doi.org/10.4028/www.scientific.net/amm.195-196.889>
4. Araya D. (2015). Interview: Michel Bauwens on peer-to-peer economics and its role in reshaping our world. [Online]. *Futurism.*
5. Miles J. Pushing the boundaries. *Research World.* 2018 May; 2018(70): 51–1. [Online] Available from: <https://library.oapen.org/bitstream/handle/20.500.12657/30986/640674.pdf?sequence=1#page=50>
6. Geng Y, Cassandras CG. New “Smart Parking” System Based on Resource Allocation and Reservations. *IEEE Trans Intell Transp Syst.* 2013 Sep; 14(3): 1129–1139. doi: 10.1109/TITS.2013.2252428.
7. Jennath HS, Adarsh S, Chandran NV, Ananthan R, Sabir A, Asharaf S. Parkchain: A Blockchain Powered Parking Solution for Smart Cities. *Front Blockchain.* 2019; 2: 6. doi: 10.3389/fbloc.2019.00006
8. Hongwei Wang, Wenbo He. A Reservation-based Smart Parking System. 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHP). 2011; 690–695. doi: 10.1109/INFCOMW.2011.5928901.
9. Tang V, Zheng Y, Cao J. An intelligent car park management system based on wireless sensor networks. in Proc. of the 1st International Symposium on Pervasive Computing and Applications, Urumchi, Xinjiang, P.R. China. 2006 Aug; 65–70.
10. Panayappan R, Trivedi J, Studer A, Perrig A. VANET-based approach for parking space availability. In Proc of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2007), Montreal, Qu´ebec, Canada. 2007 Sep; 75–76.
11. Chinrungrueng J, Sunantachaikul U, Triamlumlerd S. Smart parking: an application of optical wireless sensor network. In Proc. Of the 2007 International Symposium on Applications and the Internet Workshops (SAINTW’07), Hiroshima, Japan. 2007 Jan; 66–69.
12. Bi Y, Sun L, Zhu H, Yan T, Luo Z. A parking management system based on wireless sensor network. *Acta Automat Sin.* 2006 Nov; 32(6): 38–45.
13. Basu P, Little T. Networked Parking Spaces: Architecture and Applications. *Proceedings of IEEE Vehicular Technology Conference.* 2002 Sep; 1153–1157.
14. Caliskan M, Graupner D, Mauve M. Decentralized Discovery of Free Parking Places. *Proceedings of ACM International Workshop on Vehicular Ad Hoc Networks.* 2006 Sep; 30–39.
15. Caliskan M, Barthels A, Scheuermann B, Mauve M. Predicting Parking Lot Occupancy in Vehicular Ad Hoc Networks. *Proceedings of IEEE Vehicular Technology Conference.* 2007 Apr; 277–281.