Efficient Fuel Delivery at Your Fingertips: Developing a Seamless On-Demand Fuel Delivery App with Flutter

Navneet Mishra[1], Ritika Raghuwanshi[1], Naveen Kumar Maurya[2], and Indrajeet Kumar[1](✉)

[1] School of CSIT, Symbiosis University of Applied Sciences, Indore, India
indrajeet.kumar@suas.ac.in
[2] ECE, Vishnu Institute of Technology, Bhimavaram, Andhra Pradesh, India

Abstract. The increasing demand for fuel due to the growth of automobiles in the market has led to the need for on-demand fuel supply applications that depend on user orders and requirements. When a vehicle runs out of fuel, it can be a hassle for the owner to push the car or seek help to reach the nearest gas station. For older people and those who are medically ill, this task can be even more difficult. Additionally, people must go to gas stations to fill up generators. To address these issues, we introduce a new solution for vehicle refueling and emergency power supplies through the development of an on-demand fuel delivery application. This application provides door-to-door coverage and allows end users to choose the type of fuel they need, order it, and receive it with ease. The outcome of this research paper will be the development of a mobile application using Flutter framework that offers a range of functionalities catering to both customers and fuel station owners. The application aims to provide a convenient platform for customers to order fuel, locate nearby gas stations, and assist owners in efficiently managing orders and monitoring station availability. By utilizing Flutter, a cross-platform development framework, the application will be compatible with both Android and iOS devices, ensuring a broader reach and accessibility for users. Flutter's rich UI capabilities and native-like performance will enable the creation of a visually appealing and seamless user experience.

Keywords: Arduino · Flutter · LDR · Eye Blink Sensor · Ultrasonic Sensor

1 Introduction

Our app-based service, On-Demand Fuel Delivery Application that is built using flutter, is similar to what we other On-Demand Delivery services but we aim to provide fuel to other customer. Our goal is to establish a system where the user can request for the fuel to be deliver to his footsteps [1]. We aim to provide timely delivery of fuel to customers. In this modern fast paced world where demands are amplifying day-by-day, we aim to revolutionize of the least modernized area by introducing our On-Demand

P. Pareek et al. (Eds.): IC4S 2023, LNICST 537, pp. 134–147, 2024.
https://doi.org/10.1007/978-3-031-48891-7_11

Fuel Delivery Application built using Flutter which removes the existing constraints and gives an easy to use, safe, reliable way to meet the user demands [2–4]. On-Demand Fuel Delivery Application provides online fuel ordering services including an engaging and comprehensive online fuel ordering process such as ordering online, tracking order, and checking the fuel prices nearby. The future scalability of on-demand fuel services is enormous, and several end-users can be targeted. For example, 37% of India's electricity in urban areas is generated solely by diesel generator sets [5].

The paper [6] provides a review of various fuel management systems used in transportation, highlighting their importance in reducing fuel consumption and costs. The study covers the technical aspects of FMS, challenges in implementing FMS, and their potential benefits in the transportation industry. The research paper [7–9] examines the challenges of meeting the growing demand for road fuel in these countries driven by economic development and population growth. The study analyses factors affecting demand and the effectiveness of policy interventions like fuel taxes and subsidies. The paper suggests a more comprehensive approach is needed to promote sustainable transportation that considers technological innovations and behavioural change. It offers recommendations for policymakers to address these challenges.

The paper [10–13] examines the sector-wise demand for diesel and petrol in India, with a focus on key drivers of demand. The study provides insights for policymakers and industry stakeholders and contributes to a better understanding of the energy sector in India.

Fig. 1. Shows the utility of on demand fuel app.

In Fig. 1, we are presented with a demonstration of the practicality and effectiveness of the on-demand fuel app, which is developed using both the Flutter framework [14–16] and machine learning technology [17]. The figure illustrates how this innovative combination enables the app to deliver exceptional performance and enhanced user experiences. With the Flutter framework, the app achieves seamless multi-platform compatibility, allowing users to access its features effortlessly across mobile, web, and

desktop devices. This unified approach ensures that the app can reach a broader audience and cater to diverse user preferences. MIMO technology's ability to mitigate channel fading and reduce signal degradation can contribute to improved network connectivity and reliability for Flutter apps [18]. By leveraging spatial multiplexing and beamforming techniques, MIMO can ensure more robust communication between the app and backend servers, resulting in more stable data transmission and reduced packet loss [19–23].

Furthermore, the integration of machine learning technology enhances the app's capabilities significantly. Machine learning algorithms enable the app to analyze user behavior, preferences, and patterns, leading to personalized fuel delivery recommendations and optimized service [24–26]. Through continuous learning and adaptation, the app can provide tailored and efficient fuel delivery solutions for each user. By combining the power of Flutter and machine learning, Fig. 1 highlights how the on-demand fuel app becomes a robust and cutting-edge solution that revolutionizes the way users access and receive fuel services. This depiction emphasizes the app's ability to stay at the forefront of technological advancements, providing a superior and user-centric experience [27–31].

The research offers valuable insights and recommendations for policymakers to promote sustainable development. The research paper [32–34] examines the factors driving oil demand in India, the impact of policy interventions on oil demand, and recommendations for promoting energy conservation and renewable energy sources. The study suggests that India's oil demand will continue to grow due to economic development and urbanization, and offers insights and recommendations for policymakers. The Ministry of Oil and Gas of India published a report on the "All India Survey on Diesel and Gasoline Demand by Sector" which outlines the consumption of diesel and gasoline in various sectors. The transportation sector is the largest consumer at 60%, followed by agriculture and industry. The report emphasizes the need to reduce dependency on fossil fuels and promote sustainable alternatives for a greener economy. During the forecast period of 2022–2032, the On-Demand fuel delivery market is likely to increase at a CAGR of 6.8%. In 2022, this market is expected to reach around $4.8 billion. The On-Demand fuel delivery market value will likely be $6.2 billion by 2026. On-demand application revenue is likely to generate $935 billion in 2023.

The rest of the paper is structured as follows: The next section provides essential background information to understand the proposed work, including a review of related studies conducted by other researchers and the areas where their research falls short. In Sect. 3, we delve into the system's design, explaining its architecture and components in detail. Moving on to Sect. 4, we showcase the practical implementation of the on-demand fuel app using Flutter, and to further clarify its operation, we present an illustrative example. This example vividly demonstrates how the system works in a real-life scenario. Finally, in Sect. 5, we conclude the study by offering a comprehensive summary of the findings, discussing the significance of the results, and highlighting potential avenues for future research.

2 Background

Flutter is a free and open-source user interface toolkit developed by Google. It empowers developers to create natively compiled applications for multiple platforms, including mobile, web, and desktop, all from a single codebase. Google initially unveiled Flutter at the Google I/O developer conference in May 2017, and since then, it has gained widespread acclaim among developers. This popularity can be attributed to its rapid development cycle, enabling faster iterations during the coding process, its excellent performance, ensuring smooth and efficient application execution, and its ability to cater to various platforms without the need for separate codebases, thereby streamlining the development workflow. Flutter uses a reactive programming model, where changes to the UI are automatically reflected in the app's state, and vice versa. This enables developers to build highly interactive and responsive apps with a smooth user experience. Flutter also comes with a rich set of customizable widgets and allows developers to create their own widgets or modify existing ones to suit their needs.The proposed system is divided into five distinct working units, each serving a specific purpose. The IR sensor hurdle detection unit is responsible for detecting obstacles using infrared technology. The ultrasonic hurdle detection unit utilizes ultrasonic waves to identify objects in the vehicle's vicinity. The automatic headlight unit ensures that the vehicle's headlights are activated or deactivated based on the ambient lighting conditions. The engine control unit utilizes the alcohol sensor to prevent the vehicle from starting if the driver is under the influence of alcohol. Lastly, the drowsiness detection unit utilizes the eyeblink sensor to monitor the driver's alertness level and provide timely alerts if signs of drowsiness are detected.

Flutter offers an incredibly useful feature known as "hot-reload," allowing developers to instantly view the changes they make to the code in real-time, without the need to restart the application. This feature significantly accelerates the development process and enhances efficiency, as developers can rapidly experiment with various adjustments and instantly see their impact.

Furthermore, Flutter provides a comprehensive set of tools to facilitate the development experience. One of these tools is Flutter Studio, a robust Integrated Development Environment (IDE) that empowers developers to create and design their applications efficiently. Additionally, Flutter offers command-line tools and plugins that seamlessly integrate with popular development environments like Android Studio and Visual Studio Code, making it even more convenient for developers to work with their preferred tools.

In conclusion, Flutter proves to be a highly versatile and powerful toolkit that caters to developers' needs, enabling them to produce top-notch, cross-platform applications with ease. Its combination of hot-reload and feature-rich development tools fosters a smooth and productive development environment, ultimately resulting in high-quality, responsive, and engaging applications for various platforms. Its popularity is only expected to grow in the coming years as more developers discover its potential and adopt it for our paper.

3 System Model

Flutter is a powerful and versatile open-source framework developed by Google for creating cross-platform mobile applications. It allows developers to build high-quality, natively compiled apps for both Android and iOS platforms using a single codebase. Flutter employs the Dart programming language, which is known for its simplicity and ease of learning. One of the key strengths of Flutter is its fast and hot reload feature, enabling developers to see instant updates on the app as they make changes to the code. This significantly speeds up the development process and facilitates rapid prototyping and iterative improvements.

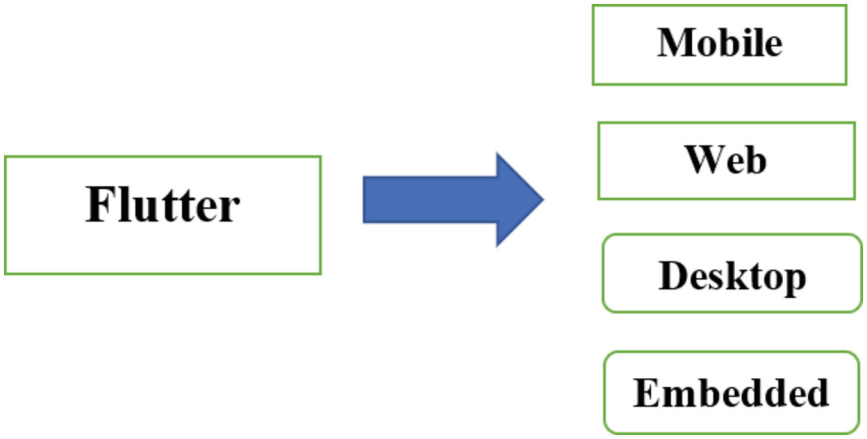

Fig. 2. Empowering Seamless User Interface on Multiple Devices using Versatile UI Framework

For startups, the ability to seamlessly connect with users across various platforms such as mobile, web, and desktop through a unified app empowers them to effectively reach their entire audience right from the beginning. This approach eliminates any constraints imposed by technical limitations, ensuring a broader accessibility to their products and services. By catering to multiple platforms with just one codebase, startups can allocate their resources more efficiently and avoid the complexities associated with managing separate codebases. Not only does this streamline development efforts, but it also leads to cost savings and faster time-to-market.

Similarly, for larger organizations, offering a consistent user experience to all users, regardless of their device or platform, streamlines the development process and reduces overall complexity (Fig. 2). By maintaining a single codebase, these organizations can focus on improving the quality of the user experience and iterating on their app more effectively. This unified approach enables them to strategically allocate their development team's efforts, resulting in higher user satisfaction.

In both cases, leveraging a single app codebase for multiple platforms provides a competitive advantage, allowing businesses to remain agile and responsive to user needs. This flexibility in reaching a diverse user base enhances user engagement and opens up new opportunities for growth and success in today's interconnected digital landscape.

These widgets are customizable and can be combined to build complex UI layouts with ease. Due to its native-like performance, Flutter delivers smooth and fluid user experiences, which are crucial for mobile applications. It achieves this by compiling the Dart code directly into native ARM machine code, eliminating the need for a bridge between the application and the platform's native components. Another advantage of Flutter is its strong community support and an ever-growing ecosystem of packages and plugins. These resources provide developers with a wide range of functionalities and integrations, making it easier to implement various features in their apps. With Flutter, developers can also create beautiful animations and stunning visuals, enhancing the overall user experience. It supports 2D and 3D graphics rendering, allowing for eye-catching visual elements. Furthermore, Flutter's single codebase approach simplifies maintenance and reduces development costs, as developers do not need to manage separate codebases for different platforms.

Our On-Demand Fuel Delivery Application mainly consists of four modules:

3.1 Register Module

The registration module of our On-Demand Fuel Delivery Application requires users as well as the fuel station to enter their credentials and login first. To register as a user, you will be required to provide specific personal details, including your full name, contact number, email address, chosen username, and a secure password. On the other hand, gas stations are also required to register by submitting their pertinent information, which includes the gas station's name, contact number, email address, their chosen username, a password, and the physical location of the gas station.

3.2 Information Module

The gas station is obligated to furnish essential details regarding fuel availability, the pricing of various fuel types, the range of fuels they offer, and the services they provide. As fuel is a crucial element for any vehicle, its price experiences daily fluctuations, and these prices may also vary based on the gas station's location. Consequently, it becomes the gas station's responsibility to ensure that fuel prices are updated on a daily basis to accurately reflect the current market rates. This timely updating ensures transparency and facilitates customers in making informed decisions about fuel purchases.

3.3 Order Fuel Module

Once a user registers with the application and gains access to its services, they can conveniently order fuel from their location based on their specific requirements. To utilize the application's features, users are prompted to enter their credentials for authentication.

To place a fuel order, users must first locate a nearby gas station using the application's built-in features. Upon finding a suitable gas station, they can proceed to check the availability of fuel and the range of services offered by that particular station. After verifying the availability of the required fuel and desired services, users can then proceed to place their fuel order according to their needs and preferences. This streamlined process ensures that users have easy access to fuel services while being well-informed about the options available to them.

3.4 Track Order Module

After a user places an order for fuel through the application, they gain the ability to track the progress of the order. The user can monitor various stages, such as whether the order has been accepted by the gas station and whether the fuel delivery has been made. To stay informed about these updates, it is crucial for the gas station to actively participate in the process.

Upon receiving an order, the gas station is responsible for either approving or rejecting it. Subsequently, the gas station is required to regularly update the order status to keep the user informed about any changes in the order's progress. By actively participating in this process, the gas station ensures effective communication with the user, enabling them to know the current status of their fuel order and anticipate the delivery accordingly.

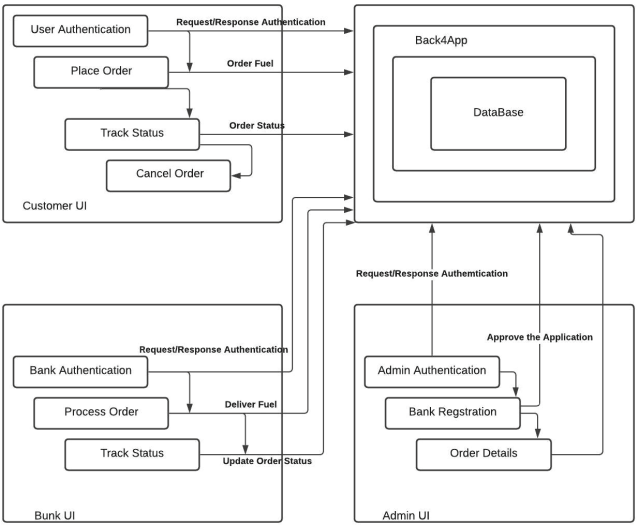

Fig. 3. In this figure we can see the entire architecture and the flow of the entire application:

The Customer UI includes the interactive environment developed for the customer. The entire section contains all the section present directly to the customer as shown in Fig. 3. First Section in the Customer UI is the User Authentication section in which the user must enter their credentials through the Request/Response Authentication in which one module sends request to other module and waits for a response. The user credentials are sent to the Database of the Application for Authentication. Once the response is received from the Database Section the process is processed accordingly. Once User Authentication is completed the customer then can move forward to the Place Order Section which enables the customer to be able to Place Order for the Fuel Delivery. Once the user has placed the order the record is stored in the database. Once the user has placed the order, they will be able to track their order status where they can see the current status of their order and can track the location of the delivery. The customer

would also have access to Cancel Order where the customer can cancel their order and once the cancel request is completed the process the aborted and an entry is made in the database regarding the entire case.

The Bunk UI will handle all the transaction related process where we intend to hire a third party for safer, faster and reliable transactions. Bunk is the placeholder for the third-party service we intend to utilize in our application. The Bunk UI contains Bank Authentication, Process Order, Track status. First the Bank Authentication will include verification of the bank details the customer provides with their respective bank, once the verification is completed the service will be providing the customer with number of options for the transaction that their bank provides for smoother user experience. If bank details are stored in the database the service will request the details from the database module and wait for the response and if the details are not present the process is proceeded as mentioned above and the user bank details are stored in the database. Once the Authentication and the transaction is completed then the order is processed forward and a message is sent to the database regarding the process and on successful user transaction the customer is send the details of the transaction. In the Track Status section, the Bunk UI constantly monitors and sends the updates to the database. If the order is cancelled the Bunk UI starts the process of returning the money to the customer by requesting the details from the database.

The Admin UI contains all Admin tools providing several productivity features. The architecture focuses on the three section of the Admin UI namely Admin Authentication, Bank Registration and Order Details. The admin module manages the entire system, and only authorized personnel can access it. The admin can view all registered users and fuel stations, track orders, and oversee payments. The admin module also provides analytical reports, including sales reports and customer insights, to help the fuel delivery service to make informed decisions.

4 Implementation

Our On-Demand Fuel Delivery Application has been skillfully developed and put into action, allowing users to access and request fuel delivery services conveniently and efficiently. Through our implementation, we have ensured a seamless user experience, enabling customers to order fuel at their convenience and have it delivered promptly to their desired location. Our innovative application built using cutting-edge Flutter technology has proven to be a game-changer, revolutionizing the way fuel is delivered to users, and enhancing the overall experience of fuel procurement. The implementation of our On-Demand Fuel Delivery Application will involve the following stages:

4.1 Design Phase

During this stage, the user interface, features, and functionalities of the application will be designed. We will use Flutter to build the user interface, and we will incorporate Material Design principles to ensure a clean, modern look.

4.2 Development Phase

This stage involves the actual coding of the application. We will use the Flutter framework to develop the application, and we will use Firebase for user authentication and database management. Google Maps API will be integrated for displaying the nearby gas stations and tracking the fuel delivery.

4.3 Testing Phase

During this phase, the application will undergo thorough testing to verify its proper functionality and alignment with user requirements. We will conduct two essential types of testing: unit testing and integration testing. Unit testing involves evaluating individual components of the application in isolation to ensure their proper operation. Each component will be rigorously tested to verify that it performs as intended and meets its specified functionalities.

Integration testing, on the other hand, assesses the interaction and compatibility between different components of the application. We will examine how these components work together harmoniously to deliver the desired features and functionalities as a cohesive whole. By performing both unit and integration testing, we aim to ensure that the application is reliable, robust, and capable of fulfilling user expectations. This meticulous testing process enables us to identify and address any potential issues or discrepancies before the application is launched to end-users, ensuring a smooth and satisfactory user experience.

4.4 Deployment Phase

After the completion of development and rigorous testing, the application will be ready for deployment. It will be made available to the public through the Google Play Store for Android users and the Apple App Store for iOS users. Once deployed, users can easily discover the application on the respective app stores and download it onto their smartphones. They can then proceed to install the application, granting them access to its full functionality and features. By making the application accessible on these widely-used platforms, it ensures a broad reach to users across both Android and iOS ecosystems. This approach maximizes the application's exposure and availability, allowing a diverse user base to benefit from its offerings.

5 Results

The following figures illustrates the key screens that the user will interact with while using Our On-Demand Application built using Flutter. Flutter is Google's SDK for crafting attractive, User-friendly environment for mobile, web, and desktop from a single codebase.

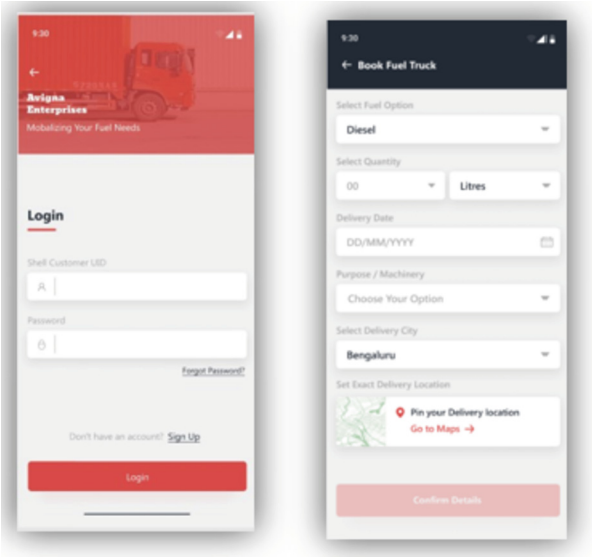

Fig. 4. Illustrating Login (Left) and Home (Right)

Figure 4 Login Page where registered users can input their pre-existing credentials stored in the database and login to their accounts. User can use the forgot password feature incase the user forgets their password. Users will be able to reset their password through email verification. Unregistered users can Sign Up in-order to use the Application. Main Activity screen is displayed when the user has successfully has verified his credentials and has login to the application. In this screen the user can click on book fuel truck and proceed to the next step of the process.

Figure 5 Place Order Activity screen provides user with a lot of options starting with fuel option where user can choose the type of fuel he wants to be delivered, user can also choose the quantity of the fuel and the measurement units. User must provide the delivery date and purpose of his request for the fuel. The user then must provide his exact location and once the user has input the details, he can proceed by clicking on the Confirm Details.

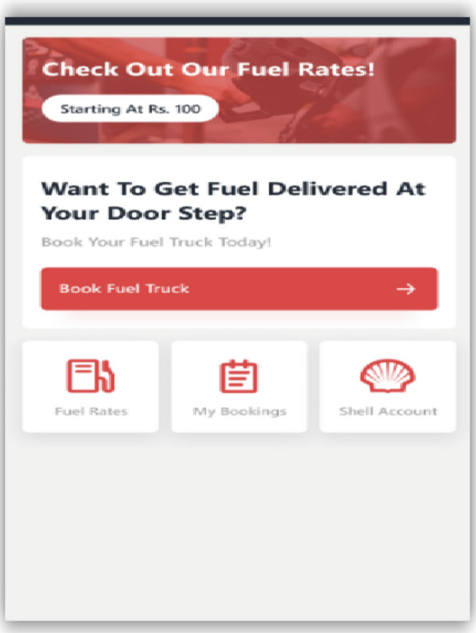

Fig. 5. Illustrating the order activity screen provides user with a lot of options starting with fuel option.

6 Challenges and Opportunities

The On-Demand Fuel Delivery Application faces several challenges and opportunities. One of the challenges is building a robust and reliable delivery system. Fuel delivery requires special equipment and trained personnel, and the application must ensure that the delivery is safe, efficient, and timely. The application must also comply with regulatory requirements, such as fuel transportation regulations, to ensure the safety of the delivery personnel and users.

Another challenge is the adoption of the application by fuel stations and customers. The application must convince fuel stations to partner with the service, and customers must be willing to use the application instead of visiting gas stations physically. The application must also be user-friendly, reliable, and provide a seamless experience to ensure customer satisfaction.

The On-Demand Fuel Delivery Application also presents several opportunities. The first opportunity is providing convenience to users. Users can avoid the hassle of visiting gas stations and waiting in queues by ordering fuel through the application. The application can also provide users with real-time fuel prices, promotions, and discounts, helping them save time and money.

7 Conclusion

Our On-demand fuel delivery application has the potential to revolutionize the traditional fuelling process and provide customers with a convenient, time-saving, and efficient service. By leveraging technology and innovation, the application can address the challenges and limitations of traditional fuelling methods, such as long queues, time-consuming commutes, and limited availability of fuel stations. Our On-Demand fuel delivery application represents an innovative and promising solution to the challenges and limitations of traditional fuelling methods. By continuously improving and innovating the application, we can further enhance its benefits and promote a more sustainable and efficient fuelling system for the future. Overall, the On-Demand Fuel Delivery Application developed using Flutter will help meet the growing demand for fuel while promoting sustainability and creating a more convenient and reliable fuel delivery system. In summary, Flutter is a powerful and popular framework for mobile app development that offers a wide range of benefits, including fast development, native performance, expressive UI design, and a strong community support system. Its versatility and efficiency make it an excellent choice for building modern and feature-rich mobile applications.

References

1. Ahmed, A.A.I., Mohammed, S.A.E., Satte, M.A.M.H.: Fuel management system. In: 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan, pp. 1–7 (2017)
2. Chandrasiri, S.: Demand for road-fuel in a small developing economy: the case of Sri Lanka. Energy Policy **34**(14), 1833–1840 (2006)
3. Nielsen India Private Limited: All India Study on Sectoral Demand of Diesel & Petrol. Ministry of Petroleum and Natural Gas (2013)
4. Rivera-González, L., Bolonio, D., García-López, G.A., Alvarez, M.: Long-term forecast of energy and fuels demand toward a sustainable road transport sector in Ecuador (2016–2035): a LEAP model application. Energies **12**(20), 3849 (2019). https://doi.org/10.3390/en12203849
5. Agarwal, P.: India's Petroleum Demand: Empirical Estimations and Projections for the Future. Institute of Economic Growth (IEG) University (2012)
6. Rabinovich, A., Azuri, Y., Shtilman, L.: Assessment of fuel delivery system of a high-performance UAV engine. J. Propul. Power **34**(4), 880–888 (2018)
7. Gao, H., Liu, J., Huang, Q.: Fault diagnosis of fuel delivery system for diesel engine based on dynamic Bayesian network. J. Mech. Sci. Technol. **33**(5), 2245–2253 (2019)
8. Huang, K., Xie, S., Wang, X., Sun, L.: Design and simulation of a fuel delivery system for a variable compression ratio engine. Energies **13**(22), 6029 (2020)
9. Wang, J., Liu, J., Huang, Q.: Design of a fuel delivery system for high-speed diesel engine based on digital simulation technology. Int. J. Automot. Technol. **22**(3), 1045–1056 (2021)
10. Williams, T.M., Pearson, J.M.: Fuel Delivery Systems for Gasoline Direct Injection Engines. SAE Technical Paper, 2018–01–0312 (2018)
11. Manh, N.P., Jeong, H.G.: Modeling and control of a fuel delivery system for gasoline engines. Energies **10**(8), 1221 (2017)
12. Kuo, Y.S., Chen, W.L.: Design and optimization of a fuel delivery system for a diesel engine using CFD simulation and RSM methodology. Energies **9**(11), 918 (2016)

13. Sharpe, R.G., de Bruin, T.: Fuel delivery system modeling for high-pressure common rail diesel engines. J. Eng. Gas Turbines Power **136**(6), 061505 (2014)
14. Ameen, S.Y., Mohammed, D.Y.: Developing cross-platform library using flutter. Eur. J. Eng. Technol. Res. **7**(2), 18–21 (2022)
15. Wiriasto, G.W., Aji, R.W.S., Budiman, D.F.: Design and development of attendance system application using android-based flutter. In: 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE), pp. 1–6 (2020)
16. Kavitha, M., Srinivas, P.V.V.S., Kalyampudi, P.S.L., Srinivasulu, S.: Machine learning techniques for anomaly detection in smart healthcare. In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp. 1350–1356 (2021). https://doi.org/10.1109/ICIRCA51532.2021.9544795
17. Vadrevu, P.K., Veeramanickam, M.R.M., Adusumalli, S.K., Bunga, S.K.: Sign language recognition for needy people using machine learning model. In: Intelligent Computing and Applications: Proceedings of ICDIC, pp. 227–233 (2020). Singapore: Springer Nature Singapore, 2022
18. Kumar, I., Mishra, M.K., Mishra, R.K.: Performance analysis of NOMA downlink for next-generation 5G network with statistical channel state information. Ingénierie des Systèmes d'Information **26**(4), 417–423 (2021). https://doi.org/10.18280/isi.260410
19. Shankar, R., Kumar, I., Mishra, R.K.: Pairwise error probability analysis of dual hop relaying network over time selective Nakagami-m fading channel with imperfect CSI and node mobility. Traitement du Signal **36**(3), 281–295 (2019). https://doi.org/10.18280/ts.360312
20. Kumar, I., Kumar, A., Mishra, R.K.: Performance analysis of cooperative NOMA system for defense application with relay selection in a hostile environment. The Journal of Defense Modeling and Simulation (2022). doi:https://doi.org/10.1177/15485129221079721
21. Ashish, I.K., Mishra, R.K.: Performance analysis for wireless non-orthogonal multiple access downlink systems. In: 2020 International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET), Patna, India, pp. 1–6 (2020). https://doi.org/10.1109/ICEFEET49149.2020.9186987
22. Maurya, N.K., Kumari, S., Pareek, P., Singh, L.: Graphene-based frequency agile isolation enhancement mechanism for MIMO antenna in terahertz regime. Nano Communication Networks, p. 100436 (2023)
23. Maurya, N.K., Bhattacharya, R.: CPW-fed dual-band compact Yagi-type pattern diversity antenna for LTE and WiFi. Progress In Electromagnetics Research C **107**, 183–201 (2021)
24. Maurya, N.K., Bhattacharya, R.: Design of compact dual-polarized multiband MIMO antenna using near-field for IoT. AEU-International Journal of Electronics and Communications **117**, 153091 (2020)
25. Kumar, I., Mishra, R.K.: An investigation of spectral efficiency in linear MRC and MMSE detectors with perfect and imperfect CSI for massive MIMO systems. Traitement du Signal **38**(2), 495–501 (2021). https://doi.org/10.18280/ts.380229
26. Kumar, I., Mishra, R.K.: An efficient ICI mitigation technique for MIMO-OFDM system in time-varying channels. Mathematical Modelling of Engineering Problems **7**(1), 79–86 (2020). https://doi.org/10.18280/mmep.070110.
27. Valarmathi, B., et al.: Price estimation of used cars using machine learning algorithms. In: International Conference on Cognitive Computing and Cyber Physical Systems, pp. 26–41 (2022). Springer Nature Switzerland, Cham
28. Biorn-Hansen, A., Rieger, C., et al.: An empirical investigation of performance overhead in cross-platform mobile development frameworks. In: Empirical Software Engineering **25**, pp. 299730240 (2020). Springer
29. Kumar, I., Sachan, V., Shankar, R., Mishra, R.K.: An investigation of wireless S-DF hybrid satellite terrestrial relaying network over time selective fading channel. Traitement du Signal **35**(2), 103–120 (2018). https://doi.org/10.3166/TS.35.103-120

30. Kumar, I., Sachan, V., Shankar, R., Mishra, R.K.: Performance Analysis of Multi-User Massive MIMO Systems with Perfect and Imperfect CSI. Procedia Computer Science **167**, pp. 1452–1461 (2020), ISSN 1877–0509. https://doi.org/10.1016/j.procs.2020.03.356
31. Gupta, N., Kumar, I., Rathod, I., Sharma, S.S.P.M.: Sustainable Production Systems with ai and Emerging Technologies: A Moderator-Mediation Analysis. **12**(Special Issue-8), 2819–2832 (2023). https://doi.org/10.48047/ecb/2023.12.si8.200
32. Arb, G.I., Al-Majdi, K.: A freights status management system based on dart and flutter programming language. Journal of Physics: Conference Series **1530**(1). IOP Publishing (2020)
33. Pareek, P., Maurya, N.K., Singh, L., Gupta, N., Reis, M.J.C.S.: Study of smart city compatible monolithic quantum well photodetector. In: International Conference on Cognitive Computing and Cyber Physical Systems, pp. 215–224 (2022). Springer Nature Switzerland, Cham
34. Li, L., et al.: CiD: automating the detection of API-related compatibility issues in Android apps. In: 27th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), pp. 153–163 (2018)
35. Sharma, S.S.P.M., Ravishankar Kamath, H., Siva Brahmaiah Rama, V.: Modelling of cloud based online access system for solar charge controller International Journal of Engineering & Technology **7**(2.21), 58–61 (2018)
36. Shalinee Gupta, S.S.P.M., Sharma, B.: Design and Development of an Intelligent Aqua Monitoring System using Cloud Based Online Access Control Systems International Journal of Recent Technology and Engineering (IJRTE) **8**(4) (2019). ISSN: 2277–3878
37. Ravishankar Kamath, H., Sharma, S.S.P.M., Siva Brahmaiah Rama, V.: PWM based solar charge controller using IoT International Journal of Engineering & Technology **7**(2.7), 284–288 (2018)
38. Ravishankar Kamath, H., Siva Brahmaiah Rama, V., Sharma, S.S.P.M.: Street Light Monitoring Using IOT International Journal of Engineering & Technology **7**(2.7), 1008–1012 (2018)
39. Sharma, S.S.P.M., Kumar, A., Meena, B. K.: An Intelligent Solar Based Farm Monitoring using Cloud Based Online Access Control Systems International Journal of Recent Technology and Engineering (IJRTE) **8**(3) (2019).ISSN: 2277–3878