International Research Journal of Education and Technology
Peer Reviewed Journal
ISSN 2581-7795

SOCIAL MEDIA APP CLONING USING MERN Stack

Karthiga M[1], Karthick J[2] , Vignesh K[3]

[1] *Assistant Professor, Dept. of Electronics and Communication Engineering*
[2] *UG Scholar, Dept. of Electronics and Communication Engineering*
[3] *UG Scholar, Dept. of Agricultural Engineering*
[1,2,3] *Bannari Amman Institute of Technology*

Abstract - This short tutorial shows the development process of a clone of the popular social media platform Instagram using the MERN (MongoDB, Express.js, React.js, Node.js) cluster. This project focuses on a better understanding of the key components and functions of building a large and effective social media network.The development process starts with setting up the backend using Node.js and Express.js, which is the basis for performing server-side tasks. MongoDB is a NoSQL database that stores and manages user data, posts, comments and other related information.Towards the end, the powerful JavaScript library React.js was used to create interactive and efficient applications. Items such as user profiles, feeds, photo uploads, likes, comments and notifications are created and used as the main functions of Instagram.User authentication and security play an important role in all social applications. User authentication in this project uses JSON Web Tokens (JWT) and bcrypt for secure password hashing. This ensures that user accounts and information are protected.Also, the project has other features such as following and unfollowing users, direct messaging, a search function, and a search section to increase user interaction and interest. Code modularisation, scalability, and performance are emphasized throughout the entire development process. The simplicity and stability of the MERN group enables the integration of different products and efficient operation of large data sets. In summary, cloning Instagram using the MERN group gives insight into the complex process of creating a social app from scratch. This work includes back-end and front-end development, customer authentication, data management, and many other things that make for a great customer experience. By understanding the underlying structure and context, developers can use this knowledge to build their own social apps or modify existing apps to meet specific needs.

Keywords—MERN, Database, Frontend, Backend, Social media App, Features clone and special feature, High security

1.INTRODUCTION

In recent years, social media platforms have changed the way people connect, share, and interact online. Among these platforms, Instagram has become one of the most popular and influential, offering a unique combination of photo and video sharing, discussion and content discovery. The success of Instagram has inspired many developers and entrepreneurs to explore the process of cloning the platform by creating their own custom ad apps or deepening their understanding of model design.

Clone Instagram includes copying its main functions and functions while sharing personalization and personalization. By doing such projects, developers gain insight into the complexity of large-scale social applications, gain knowledge of techniques and tools, and gain hands-on experience in developing complex systems. The

MERN (MongoDB, Express.js, React.js, Node.js) stack is a popular choice for full web development and provides a solid foundation for building capacity and usability. Benefits of Instagram clones. MongoDB is used as a database to store user data, reports, comments and other related information. Node.js acts as a runtime while Express.js helps to build a powerful and flexible server. React.js is a JavaScript library that can create dynamic and interactive front-end interfaces. The process of cloning

Instagram includes many important features and features such as user authentication, user information, photo and video uploads, posts, likes, comments found, direct messages, job search and reports. Everything needs to be carefully planned, implemented and integrated to ensure a good user experience. From the beginning of the journey to clone

Instagram, developers have a deep understanding of the challenges and decisions involved in building a social media platform. They can explore the intricacies of user interaction, data management, capacity building, performance and security applications. Finally, the information obtained by copying Instagram can be used to create new builds, tweak social apps or enhance existing platforms with unique and enhanced features.

In this project, we will delve deeper into the process of cloning Instagram using the MERN group, providing step-by-step instructions and information on how to use the

various features. By reading the following pages, readers can better understand the challenges of creating social media apps and gain the skills needed to develop their own social media apps.

2.LITERATURE SURVEY

Sr. no	Site Name	Language	Limitation
[1]	Twitter	Rubyon,Rails, Scala, Java, C,Python	The limit for messages are 1000 per day.2400 tweets per day
[2]	WhatsApp	HTML5,C,C++,Java, PhP, Erlang ,Mnesia DB,SQLite database	You need to share your number in case you wantto add someone and communicate.
[3]	Insta-gram	HTML,JQuery,Python, Django,ReactNative	The daily limit is 200 a day. 10 follows and unfollows per hour would keep your account safe and saves your account from beingsuspended.
[4]	Tele-gram	C++,C#,Swift, Java, NodeJS, AngularJS,	You can't make videocalls on Telegram. You are limited to 50channels maximum.
[5]	Face-book	Hack, PHP (HHVM), Python, C++, Java, Erlang, D, XHP, Haskell, MariaDB, MySQL, HBase, Cassandra.	Limit the number of adsa page can run at once; privacy risk-often data breach frequently; Promotional way of advertising is best: the usual banner ads and pop-up ads make the users to ignore them.

Explore research papers, articles, and case studies that delve into the development process of social media applications. This should include discussions on architecture, features, scalability, and user engagement strategies. Pay attention to insights related to user data handling and security concerns. Investigate literature on various authentication mechanisms employed in social media apps. This includes detailed explanations of OAuth, OpenID Connect, SAML, and their implementations. Analyze the strengths and weaknesses of each method in terms of user security and ease of integration.[6]

DUIN J. VAN et al 2019 proposed that The affordance provided by Instagram and its Instagram Stories, and the purpose of this goal is to show that the contemporary ephemeral image is employed in favor of the platform's strategy to guide the users towards sharing their images at a high frequency. Arguably, this strategy can be related to the company's business model, and it feeds into the assumption that Instagram Stories is balancing the participation of its users with a strategy to increase profits. Through the design of the platform, Instagram steers the user towards this need of using the photo as a means of communication. JASON G MILES et al 2020 proposed that Instagram is the first social media site that was born mobile" and has become the key to kicking your mobile marketing into high gear. Instagram provides an opportunity for you to bring your company into the new mobile revolution without complexity or drama. [7]

3. METHODOLOGY

3.1 Problem Identification

➢ Social media app cloning often involves replicating the core features and functionalities of existing platforms. However, accurately identifying and recreating the unique value propositions that distinguish the original app from competitors can be challenging. Without such distinctions, the cloned app may struggle to attract and retain users in a saturated market. While users appreciate the familiarity of a cloned app resembling a popular social platform, there's a fine line between familiarity and innovation. Striking the right balance between replicating what users love and introducing innovative elements that make the clone stand out is a complex task.

3.2 Proposed Work

Our system has the following features:

A. New Feed:

The news feed is that the primary system through which users are familiar to content which were posted on the network. Sociama selects a couple of few updates to actually show users whenever they visit their feed, out of a mean of 1500 updates they go to potentially receive.

B. Friends:

The "Friending" someone is that performs of sending another user a "friend request" on Sociama. The two peopleare Sociama friends once the receiving side accepts the friend request. In addition to accepting the request, the userhas the choice of declining the friend request or hiding it using the "Not Now" or "Later" feature. Deleting or removinga request removes the request, but does allow the sender toresend it in the future. The "Not Now" or "Later" feature conceals the request but doesn't delete it, permitting the receiver to reconsider the request later.

C. Timeline:

Intended to update users' profiles in order to showcontent based on year, month, and date as well. "Cover" photos were come into the scenario, taking up a significantportion of the top of pages, and an altered display of personal information such as friends, likes, and photos appeared on the left-hand side, although story posts appeared on the right.

D. Likes:

Enables users to easily socialize with status updates, comments, photos, links shared by their friends, videos, and advertisements as well. Once clicked by a user, the designated content appears in the News Feeds of that user's friends and the button also displays the number of other users who have "liked" the content, including lists of those users.

E. Comments:

The feature allowing the users to add GIFs to commentsfor a better user experience.

F. Notifications:

Notifications tell the user that something has been added or mentioned with respect to their profile. Examples such as a message being shared on the user's profile wall or a discussion of a picture of the user or on a picture that the user has previously commented (speak about) on.

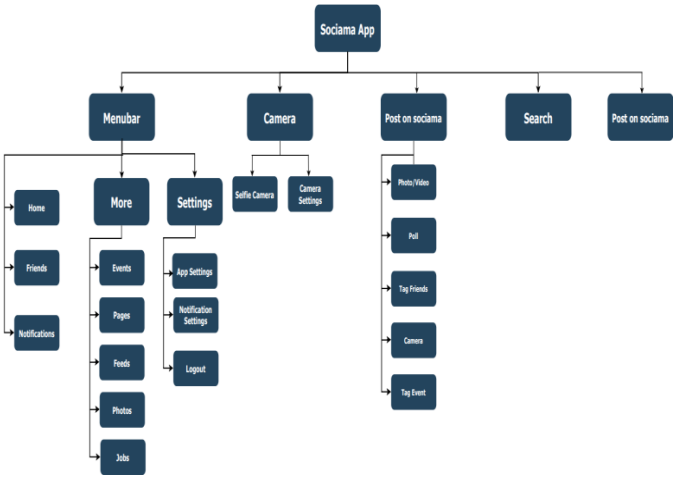

Figure 3.1: Workflow diagram

3.2.1 New feed

A News Feed in a social media app is a core feature that presents a curated stream of content to users, typically consisting of posts, updates, and media shared by their friends, followed accounts, or groups. Robust data model that represents various types of content, such as text posts, images, videos, links, and user-generated comments and reactions. Each piece of content is associated with metadata, including timestamps, user IDs, privacy settings, and engagement metrics (likes, comments, shares).
Database Management. A database is used to store and retrieve content efficiently. Common database technologies like MySQL, PostgreSQL, or NoSQL databases like MongoDB are often employed. Content is indexed for quick retrieval based on user preferences, including sorting by recency and relevance. Users must be authenticated and authorized to access their personalized News Feeds. OAuth 2.0 or JWT-based authentication is commonly used. The authorization ensures that users only see content from accounts they follow or have access to based on privacy settings.

3.2.2. Friends

A data model is created to represent friendships. Typically, this includes a database table with fields like user ID, friend ID, status (e.g., pending, accepted), and timestamps. Friendship statuses include "pending" for friend requests, "accepted" for confirmed friends, and "blocked" for blocked users. Users can send friend requests to other users. These requests are typically stored in the database with a "pending" status. Notifications are sent to users when they receive friend requests. Users can accept or reject friend requests, which updates the status of the friendship in the database. Accepted friend requests result in two-way friendships, while rejected requests are removed from the pending list. Users can unfriend or block other users, which changes the status of the friendship in the database to "unfriended" or "blocked." Blocked users may have limited interaction with the blocker, and their content may be hidden. Users have access to a list of their friends, typically displayed in a user interface. The list shows the user's friends' profiles, including their profile pictures, usernames, and status.

3.2.3 Timeline

A social media app's "Timeline" feature is where users can view a chronological stream of content from themselves and their friends or followed accounts. Below is a technical description of the "Timeline" feature in a social media app:
Technical Description of the "Timeline" Feature:
The "Timeline" feature relies on a data model that represents various types of user-generated content, such as text posts, images, videos, links, and associated metadata.Each piece of content is associated with attributes like timestamps, user IDs, privacy settings, and engagement metrics (likes, comments, shares). A database is used to store and retrieve user-generated content efficiently. Commonly used database technologies include MySQL, PostgreSQL, or NoSQL databases like MongoDB.
Content is indexed for quick retrieval based on user preferences and sorting criteria (e.g., recency, relevance). Users must be authenticated and authorized to access their personalized timelines. OAuth 2.0 or JWT-based authentication is typically employed. The authorization ensures that users only see content from accounts they follow or have access to based on privacy settings. Content for a user's timeline is aggregated from various sources, including their own posts, content from friends, followed accounts, groups, and pages. Algorithms are used to prioritize and organize content based on user engagement, relevance, and recency.

3.2.4 Likes

The "Likes" feature in a social media app allows users to express their appreciation or approval of a post, photo, video, or other types of content created by other users. Below is a technical description of the "Likes" feature in a social media app:
Users can like content by interacting with a specific UI element (e.g., a heart icon). The like action is typically implemented as an HTTP POST request to the server, which records the like in the database. Users can also "unlike" content to remove their like. Like counts and user likes are updated in real-time to reflect new likes and unlikes. Real-time technologies such as WebSockets or server-sent events (SSE) are used for immediate updates. The user interface displays the current number of likes on the content. Privacy settings may affect who can like a user's content. Users can specify whether their content is visible to the public, friends, or a custom audience. Privacy settings are enforced to ensure that only authorized users can like content. Like actions contribute to engagement metrics, and they are often used to determine the popularity of content. The total number of likes on a post can influence its visibility in users' timelines. User activity feeds may include information about the content a user has liked. This can be displayed on the user's profile or in their followers' activity feeds.

3.2.5 Comments

The "Comments" feature in a social media app allows users to provide feedback, engage in discussions, and interact with posts or content created by other users. Below is a technical description of the "Comments" feature in a social media app:

The "Comments" feature relies on a data model to represent user-generated comments associated with specific posts or content. Each comment is typically stored as a record in a database table. The data model includes attributes such as the user who made the comment, the content ID it's associated with, the comment text, a timestamp, and any related metadata. A database is used to manage and store comments efficiently. Commonly used database technologies, both relational and NoSQL, can be employed for this purpose. Indexing and caching mechanisms are often implemented to optimize the retrieval of comment-related data. Users must be authenticated and authorized to post comments. OAuth 2.0 or similar authentication mechanisms are used to ensure that comments are associated with valid user accounts. The authorization ensures that users can only comment on content they have permission to access based on privacy settings and ownership. Users can post comments by interacting with a specific UI element (e.g., a text input field). The comment action is typically implemented as an HTTP POST request to the server, which records the comment in the database.
Users can edit and delete their own comments, and they may report inappropriate comments for moderation.Comment counts and the display of new comments are updated in real time to reflect new comments and edits. Real-time technologies such as WebSockets or server-sent events (SSE) are used for immediate updates. Users are often notified of new comments on their posts or comments they've engaged with. Privacy settings may affect who can comment on a user's content. Users can specify whether their content is open to public comments, limited to friends, or restricted to a custom audience. Privacy settings are enforced to ensure that only authorized users can comment on content.

3.2.6 Notification

Notifications are triggered by various events and interactions, including receiving likes, comments, shares, friend requests, mentions, and direct messages, among others. Server-side logic determines when and how to generate notifications based on user actions and privacy settings. Notifications are often delivered in real-time to users through push notifications or in-app notifications. Real-time technologies like WebSockets or server-sent events (SSE) are used for immediate updates.
Users can receive notifications even when the app is not actively open. Notifications come in various types, such as likes, comments, mentions, friend requests, and system alerts. Each type may have a distinct icon, message, and action associated with it. Users can customize their notification settings, specifying which types of events should trigger notifications and how they should be delivered (e.g., push notifications, email notifications).
Notification preferences can be managed in user account settings. Users have access to a notification feed or panel in the app's user interface, displaying a chronological list of notifications. Users can review, interact with, and clear notifications from this feed. Users can mark notifications as read or unread to keep track of which notifications they've already viewed. Unread notifications are typically indicated with a visual cue. The app often displays a badge or counter indicating the number of unread notifications, providing users with a quick way to see their notification status. Users can interact with notifications to perform actions such as viewing the associated content, responding to comments, accepting friend requests, or navigating to the relevant part of the app. Some notifications may have actions like "Like" or "Comment" directly from the notification panel.

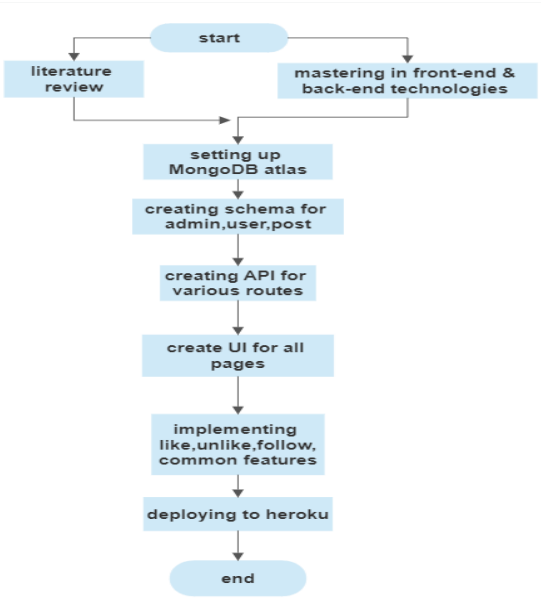

Figure 3.3: Flow diagram app development

4. PROPOSED WORK MODULES

Technologies	Purpose
React js	Front-end framework
Node js	Backend framework
Express js	Server connection
Mongo DB	Database

Table.1

A. ESP32 Microcontroller for Control:
The core intelligence of our IoT-based autonomous vehicle is powered by the ESP32 microcontroller. It serves as the central processing unit, responsible for processing sensor data, executing control algorithms, and making informed decisions. In this project, the ESP32 microcontroller plays a pivotal role in developing functional prototypes capable of obstacle avoidance and autonomous navigation.

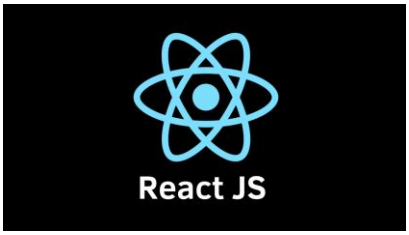

Fig.1 React framework

B. React JS framework for frontend

React.js offers several advantages over other frameworks. It provides a component-based architecture, enabling reusable and modular code. It offers a virtual DOM, allowing efficient rendering and improved performance. React js declarative syntax simplifies UI development. It enjoys a large and active community, with extensive documentation and third-party libraries. React js one-way data flow promotes predictable and efficient state management.

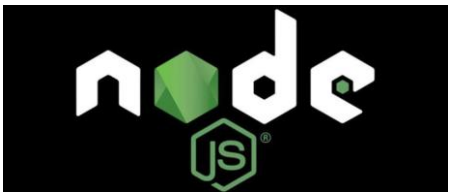

Fig.2 Node js

C. Node Js framework for backend:

Node.js brings several advantages compared to other frameworks. It uses JavaScript on both the client and server sides, promoting code reuse and a consistent development experience. Node.js offers an event-driven, non-blocking I/O model, resulting in high scalability and performance. Its extensive package ecosystem (npm) provides a wide range of modules and libraries. Node.js enables real-time applications and is well-suited for microservices architectures.

Fig.3

D. Express JS

Express.js, a popular Node.js framework, offers several advantages over other frameworks. It provides a minimalist and unopinionated approach, allowing developers to have more flexibility and control over their applications. Express.js has a simple and intuitive API, making it easy to learn and use. It offers a vast middleware ecosystem, allowing developers to extend and customize their applications easily. Express.js is lightweight, efficient, and well-suited for building RESTful APIs and web Future Prospects:

As the field of IoT-based autonomous vehicles evolves, our project points toward exciting possibilities. This may involve the integration of advanced Artificial Intelligence (AI) algorithms for smarter decision-making, exploration of advanced vehicular networking protocols for enhanced communication, and further enhancements in energy efficiency. The combination of ESP32, ToF sensors, motor driver, UI, and ESP8266 sets the stage for shaping the future of transportation, ushering in safer, efficient, and user-friendly autonomous vehicles capable of navigating intricate environments with ease

5. OVERALL IMPLICATIONS AND DISCUSSION

The technical features described, such as "Timeline," "Likes," "Comments," and "Notifications," play a pivotal role in enhancing user engagement and providing a seamless user experience. User engagement metrics, including likes, comments, and notifications, are essential for assessing the app's success and identifying areas for improvement. Privacy controls are integral to social media apps, allowing users to manage who can see their content and engage with it. Implementing robust data security measures is crucial to protect user data from breaches and unauthorized access. The use of real-time technologies like Web Sockets and server-sent events (SSE) ensures that users receive timely updates, such as new posts, comments, and notifications, providing engaging and responsive user experience features like customizable notifications and privacy settings empower users to tailor their experience to their preferences. This personalization contributes to user satisfaction and engagement. Implementing content moderation algorithms and user reporting mechanisms is vital to maintaining a safe and respectful online environment. Ensuring that users can report inappropriate content helps build trust and safety. As user bases grow, ensuring scalability and optimal performance of the app becomes paramount. Efficient database management, caching, and load balancing are key considerations.

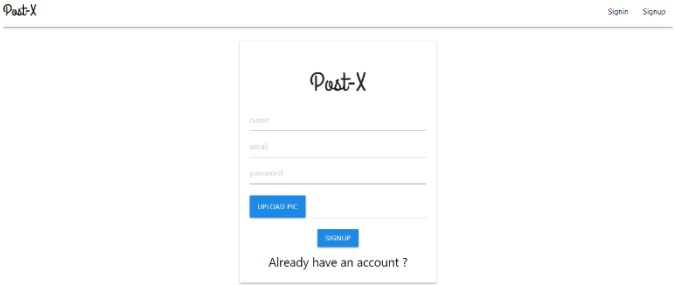

Figure 5.1 Home Page

This is the actual web application of this project. The UI shows the home tab, The web application is named Post-X.

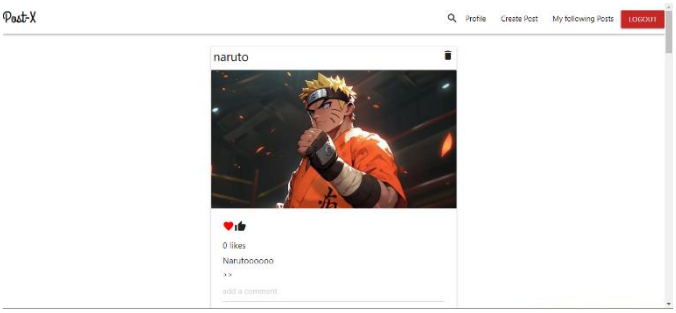

Figure 5.2 Home page

As shown in the figure 5.2, the posts of all users are fetched from database and displayed on the home page, and all this actions are managed under the state of the application

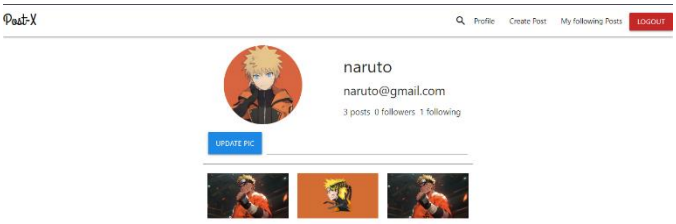

Figure 5.3 Profile page

Social media apps operate in a highly competitive landscape. To succeed, apps must offer unique value propositions, innovation, and features that differentiate them from competitors. Building a strong and engaged user community is critical for the long-term success of a social media app. User trust, authenticity, and a positive user experience are central to achieving this goal. Keeping an eye on emerging technologies and trends, such as augmented reality (AR), virtual reality (VR), and blockchain, can provide opportunities for differentiation and innovation in social media apps. As social media apps employ algorithms for content curation and recommendation, addressing algorithmic bias and ensuring ethical AI practices are essential to maintaining user trust and fairness. Social media apps often have a global user base. Adapting to diverse cultural norms, languages, and regulatory requirements is a complex but necessary aspect of app development.

5. CONCLUSIONS

The endeavor to create an IoT-based autonomous vehicle using the ESP32 microcontroller, Time-of-Flight (ToF) sensor, and motor driver has resulted in noteworthy findings and conclusions. Throughout the project's development and experimentation, several key insights have emerged.

Firstly, the integration of the ESP32 microcontroller with the ToF sensor and motor driver showcases the potential of combining these components to establish a functional and adaptable autonomous vehicle system. The ToF sensor's capability to accurately measure distances has played a pivotal role in enabling the vehicle to navigate its surroundings and avoid obstacles proficiently. The motor driver's role in controlling the vehicle's movement is fundamental, enabling precise control over speed and direction. This aspect has greatly contributed to the vehicle's capacity for autonomous navigation and effective maneuvering. The ESP32's versatility has been evident, acting as the central hub for processing sensor data, executing commands, and making real-time decisions. As a result, this project serves as a stepping stone for further advancements in autonomous vehicle systems, particularly those focused on minimalistic yet effective control mechanisms. It underlines the significance of efficient hardware integration and emphasizes the potential of the ESP32 microcontroller in autonomous vehicle applications. The project's outcome resonates with the ongoing innovation in the field and sets the stage for future explorations.

ACKNOWLEDGEMENT

I want to express my gratitude to the Mentor as well as educational institution foe their assistance and leadership

REFERENCES

[1] Carducci, C. G. C., Monti, A., Schraven, M. H., Schumacher, M., & Mueller, D. (2019). Enabling ESP32-based IoT Applications in Building Automation Systems. 2019 II Workshop on Metrology for Industry 4.0.

[2] M.H. Schraven, C. Guarnieri Calo' Carducci, M.A. Baranski, D. Muller, ¨ A. Monti, "Designing a Development Board for Research on IoT Applications in Building Automation Systems," 36th International Symposium on Automation and Robotics in Construction (ISARC 2019), Banff, Canada, 2019, in press.

[3] ESP32 Overview by Espressif Systems (Shanghai) Co., Ltd. [Online].Available:https://www.espressif.com/en/products/hardware/esp32/ overview

[4] Guide to Open Protocols in Building Automation, Schneider Electric, 2015. [Online]. Available: https://download.schneider-electric.com/files? p enDocType=Brochure&p File Name=SE+Protocols+Guide.pdf

[5] A. Maier, A. Sharp and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," 2017 Internet Technologies and Applications (ITA), Wrexham, 2017, pp. 143-148

[6] D. Ghosh, A. Agrawal, N. Prakash and P. Goyal, "Smart Saline Level Monitoring System Using ESP32 And MQTT-S," 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), Ostrava, 2018, pp. 1-5

[7] Rai, P., & Rehman, M. (2019). ESP32 Based Smart Surveillance System. 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET).

[8] Pahuja, R. and Kumar, N., 2014. Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller. International Journal of Scientific Engineering and Research, 2(7), pp.14-17.

[9] Singh, P., Sharma, D. and Agrawal, S., 2011. A Modern Study of Bluetooth Wireless Technology. Dept. of Computer sci. & Eng. Raipur, (Chhattisgarh).

[10] Chinmayi, R., Jayam, Y. K., Tunuguntla, V., Dammuru, J. V., Nadella, H., Anudeep Dulla, S. S. K., … Nair, J. G. (2018). Obstacle Detection and Avoidance Robot. 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).

[11] Arvind Kumar Saini1, Garima Sharma2, Kamal Kishor Choure3, "BluBO: Bluetooth Controlled Robot," International Journal of Science and Research (IJSR) National Conference on Knowledge, Innovation in Technology and Engineering (NCKITE), 10-11 April 2015, pp. 325-328.

[12] Singh, A., Gupta, T., & Korde, M. (2017). Bluetooth controlled spy robot. 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC).

[13] Hossai, M. R. T., Shahjalal, M. A., & Nuri, N. F. (2017). Design of an IoT based autonomous vehicle with the aid of computer vision. 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE).

[14] Bairy A. (2022). Modeling Explanations in Autonomous Vehicles. In: ter Beek, M.H., Monahan, R. (eds) Integrated Formal Methods. IFM 2022. Lecture Notes in Computer Science, vol 13274. Springer, Cham.

[15] Mazri T, Tibari S. The Proposed Self-defense Mechanism Against Security Attacks for Autonomous Vehicles. In: Ben Ahmed M, Boudhir AA, Karaş İR, Jain V, Mellouli S, editors. Innovations in Smart Cities Applications Volume 5 SCA 2021 Lecture Notes in Networks and Systems. Cham: Springer; 2022.

[16] Madhav, A.V.S., Tyagi, A.K. (2023). Explainable Artificial Intelligence (XAI): Connecting Artificial Decision-Making and Human Trust in Autonomous Vehicles. In: Singh, P.K., Wierzchoń, S.T., Tanwar, S., Rodrigues, J.J.P., Ganzha, M. (eds). Proceedings of Third International Conference on Computing, Communications, and Cyber-Security Lecture Notes in Networks and Systems. Springer, Singapore.

[17] Li Q, Wang Z, Wang W, Yuan Q. Understanding Driver Preferences for Secondary Tasks in Highly Autonomous Vehicles. In: Long S, Dhillon BS, editors. Man-Machine-Environment System Engineering. MMESE 2022. Lecture Notes in Electrical Engineering, vol. 941. Singapore: Springer; 2023.Kolb,

[18] A., Barth, E., & Koch, R. (2008). ToF-sensors: New dimensions for realism and interactivity. 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. doi:10.1109/cvprw.2008.4563159

[19] Monika Jain, Aditi, AshwaniLohiya, Mohammad Fahad Khan: Wireless gesture control robot; International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 10, December 2012 Viraj Mali, AnkitGorasia, MeghanaPatil, Prof. P.S.WawageDepartment of Information Technology, Vishwakarma Institute of Information Technology, Pune

[20] Prof. Horacio Espinosa: Robotic Control with Bluetooth Wireless Communication; Northwestern University

[21] Aiman Ansari, Yakub Ansari, SaquibGadkari, AartiGoku: Android App Based Robot; Aiman Ansari et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2) , 2015, 1598-1600

[22] Singh, P., Sharma, D. and Agrawal, S., 2011. A Modern Study of Bluetooth Wireless Technology. Dept. of Computer sci. & Eng. Raipur, (Chhattisgarh).