

Connecting Engineering Student's Talents for Productive Projects Output Using Full Stack Development

**Dr. Nagaraj P¹, Dr. Raja M², Dr. Josephine Selle Jeyanathan³, Mr. T. Venkata Dastagiri Reddy⁴,
Mr. B. Venu Gopal Reddy⁵, Ms. N. Gayathri⁶**

^{1,2,4,5,6} Department of Computer Science and Engineering, School of Computing, *Kalasalingam Academy of Research and Education, Krishnankoil, Virudhunagar, India*

³ Department of Electronics and Communication Engineering, School of Electronics, Electrical and Biomedical Technology, *Kalasalingam Academy of Research and Education, Krishnankoil, Virudhunagar, India*

¹ nagaraj.p@klu.ac.in

² mraja@klu.ac.in

³ drjosephine@gmail.com

⁴ dastagirireddy8765@gmail.com

⁵ venugopalreddybanka666@gmail.com

⁶ gayathrinetaipalli@gmail.com

Abstract— Technology has become an integral aspect of our lives in today's fast-paced society. However, students frequently confront difficulties in solving problems or obtaining money for their inventive ideas. To solve these issues, we created a set of applications that can assist students with their projects, inquiries, and financial requirements. Users can utilize our posting application to submit their projects and solicit comments from an entire group of like-minded people. They can also cooperate with different people and receive feedback on their work. Our chatbot employs cutting-edge AI technology to give speedy and accurate responses to customer inquiries. Users may simply create high-quality photographs for presentations or projects with our AI photographic creation tool. Our chat program enables users to speak in real time with one another, allowing for effortless cooperation and the exchange of data, photographs, and links. Finally, our startup fundraising tool enables students to generate funds for their ventures by publishing their thoughts and seeking contributions from those who are interested. Overall, our application suite provides a holistic solution for students wishing to improve their educational and entrepreneurial endeavors. Our software programs are intended to encourage creativity, cooperation, and success through intuitive user interfaces and technological advances.

Keywords— Engineering Students, Connecting Talents, Productive Projects, Full stack Developments, Chatbot.

I. INTRODUCTION

Our project is a complete platform aimed at providing solutions to numerous challenges that pupils and young developers encounter as they embark on their projects. The platform is made up of five separate applications that appeal to various demands.

The sharing application is an environment aimed at assisting users in improving the outcomes of non-graphical user interface initiatives. It allows people to share their ideas or prior tasks that we want to improve as well as upload images and materials relevant to the project. Other users can browse the posts and make recommendations, as well as collaborate with them via direct messages. In addition, the platform includes a pagination mechanism and a search bar to help visitors navigate through postings and discover the data they need.

The chatbot application uses the Open AI API to give users rapid and detailed responses to their questions. Users can just type their questions into the chat box, and the program will retrieve the response from the Open Artificial Intelligence API and display it to them. This is especially handy for learners who need to locate information quickly and don't want to sift through many sources. Users who have signed in to the uploading application can also access the chatbot. Fig.1 depicts the formal Logo for the project Connecting Talents.



Fig. 1. Project Logo

This paper was submitted for review on August 21, 2023. It was accepted on November 15, 2023.

Corresponding author: Nagaraj P, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Address: Anand Nagar, Krishnankoil, Srivilliputtur, Tamil Nadu, India. – 626 126 (e-mail: nagaraj.p@klu.ac.in).

Copyright © 2024 JEET.

The AI Image-generating tool is an excellent resource for students who require high-quality photographs for presentations or projects. Users can type a prompt into the tool, and the application will provide a picture based on that prompt. This can save pupils an enormous quantity of energy and time that they would ordinarily devote to browsing the internet for appropriate photographs. The app is coupled with the posting app, so users may share the photographs they create with the community.

The chat interface is intended to give users an easy way to connect. Users may establish chat groups, communicate privately, and send photographs, links, and files. A green dot displays on a user's profile when they are online, suggesting that they are open for a chat. This app has been combined with the uploading app and is accessible by logged-in users.

The startup funding software is intended to assist students in raising funds for their initiatives. Users can write a post about their idea and how much funds they need. Other people can then support the initiative by contributing funds using an Eth digital currency wallet called MetaMask. This can be a useful tool for learners who have brilliant concepts but lack the necessary funds to see them through to completion.

Overall, this work is a unified platform that provides a variety of helpful apps and tools to assist students with their educational pursuits. The combo of the uploading app, chatbot, AI image production tool, chat app, and funding for startups program offers students a complete academic solution.

II. REVIEW OF LITERATURE REVIEW

Henriyan et al., (2019) implemented a Web and Mobile Chat Interface Server using Node.js, Socket.io, MongoDB, and Java, targeting Bandung City and Indonesian users. Clients communicate with the server using JSON as middleware. The application begins with a login page on the main interface. Users need to establish friendships to initiate chats, leading to automatic chatroom creation. The application ensures data transmission in under one second.

Dipina Damodaran et al., (2016) compared MySQL and MongoDB performance on a hypermarket app using execution time as the metric. Initially, both showed similar times for small record counts. However, as records increased, MongoDB outperformed MySQL significantly, making it preferable for higher volumes. RDBMS lacked horizontal scaling for heavy transactions, while NoSQL offered scalability but with compromises in data integrity and joins.

Sahu et al., (2023) created a chat service using React, emphasizing user engagement. The app offers text interactions, smooth engagement, and upcoming features like file sharing, multimedia messaging, and voice calls. They compare it favorably to a PHP-MySQL alternative, showcasing quicker real-time performance, notable process time gains, and improved RAM usage, all while highlighting React and MongoDB's superiority for the application. The Limitation of the application was allowing a maximum of only 100 users to access and utilize the app concurrently.

Liang et al., (2017) created Express Supervision System, emphasizing AngularJS for frontend, NodeJS for backend, and

MongoDB for efficient data storage. It highlights MongoDB's role in storing large data and MapReduce-based statistical analysis. The study focuses on building web services meeting big data visualization needs using NodeJs. The application outlines the system's architecture, user query, and statistical analysis modules, while future work aims to enhance analysis accuracy with machine learning.

Wang et al., (2017) explores Node.js's server-side popularity and its event-driven architecture leading to concurrency problems. They introduce NodeCB, which scrutinizes 57 actual concurrency bugs in open-source Node.js apps. The study dissects bug patterns, causes, and solutions, highlighting atomicity violations as a key factor. NodeCB seeks to improve bug detection and resolution in Node.js by using Pattern-based, Resource-oriented, and API-usage-guided approaches.

Saundariya et al., (2021) created a web application using MongoDB, Express.js, React.js, and Node.js for booking handyman services. It addresses the growing demand for household repairs by offering an online platform where users can easily find and hire skilled workers. Professionals can showcase their skills and availability, while users can access various services based on location and cost. This streamlined system enhances the informal job sector, benefiting both job seekers and clients.

Abdullah and Zeki (2014) have focused on the technological foundations of social networking sites, using Facebook as a prime example. It explores the frontend and backend technologies employed by Facebook, including tools like Thrift, Scribe, Cassandra, and Hadoop. The project also highlights Facebook's innovations such as Hiphop for PHP optimization, Haystack for photo storage, and Varnish for caching and load balancing. The study underscores Facebook's impact on global social networking to address unprecedented traffic and user challenges.

Laksono (2018) compared SQL (PostGIS) and NoSQL (MongoDB) databases for handling geospatial big data. A NodeJS-based web app tests their performance using simulated data. MongoDB shows better loading performance, suggesting its potential for web GIS apps with large spatial data. The research highlights NoSQL's advantage over traditional SQL databases like PostGIS in handling geospatial data, demonstrated through a NodeJS full-stack app.

Nagarhalli et al., (2020) examine current trends in chatbot development across domains, highlighting gaps in language processing techniques, knowledge types, and data methods. It proposes leveraging diverse machine learning and exploring unstructured data for improved chatbots, revealing research prospects in natural language processing.

Patil et al., (2017) compared MongoDB and MySQL databases for insertion and retrieval operations in a web/android application, focusing on Load Balancing through MongoDB's auto-sharding. MongoDB's document-oriented nature and built-in load balancing are highlighted as effective solutions for managing large data loads. The study demonstrates that MongoDB outperforms MySQL in terms of load and retrieval times.

III. MATERIALS DETAILS

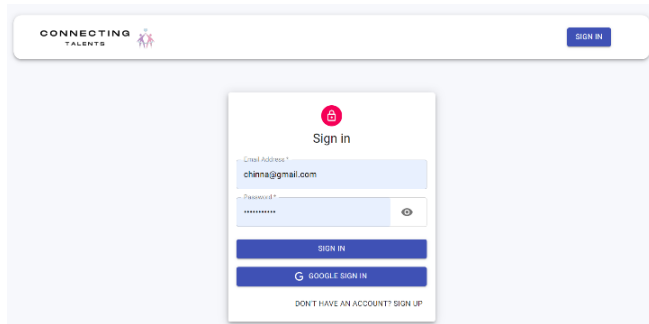


Fig. 2. Sign-in to the Application.

A. Technologies

1) Posting Application:

- A laptop or smartphone with access to the internet
- The database of MongoDB
- Image captioning API key from Open AI
- Google Authentication API
- A pagination library (for example, React-Paginate).
- A search library (for example, React-Search-Box).

2) Chatbot

- An internet-connected computer or mobile device.
- Open AI API key for NLP
- Chatbot interface (such as React-Simple-Chatbot)
- Dialog flow integration (optional)

3) AI Image Generation Tool:

- An internet-connected computer or mobile device.
- Open AI API key for the generation of images.
- Processing image libraries (for example, Pillow in Python or Sharp in Node.js)
- A file storage system (for example, Amazon S3)

4) Chat Application:

- An internet-connected computer or mobile device.
- The database of MongoDB
- Interaction in the real-time library (for example, Socket.io)
- A user identification library (for example, Passport.js)
- Online status signal (for example, Online/Offline User Status in React)

5) Startup Funding:

- An internet-connected computer or mobile device.
- Eth wallet for cryptocurrency MetaMask
- Integration of payment gateways (for example, Stripe or PayPal)
- A back-end website (which could use Node.js or Python Flask) for transaction processing as well as information storage.

B. Methodology

1) Posting Application:

- Determine the app's customer needs and capabilities.
- Create and build a user interface wireframe
- Create the front end with HTML, CSS, and JavaScript

frameworks such as React or AngularJS.

- Integrate with the backend via Node.js and Express.js, as well as create a MongoDB database for data storage.
- Use technologies such as OAuth or JWT to implement user authentication and authorization.
- Properly test the application before deploying it to a cloud-based platform such as AWS or Google Cloud.

Fig.2 to 6 describes the Login Page of the application.

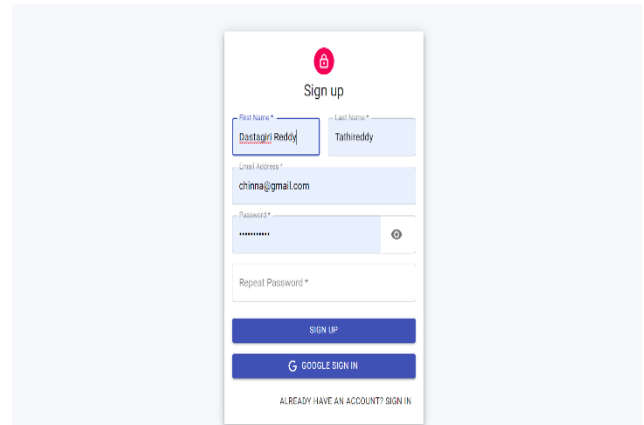


Fig. 3. Signup to Application

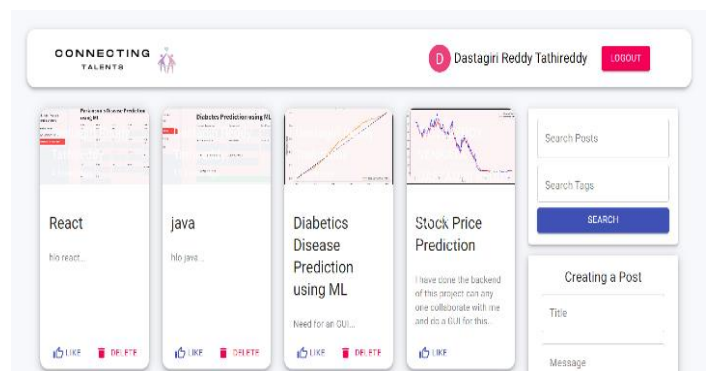


Fig. 4. Posts Likes and delete a post

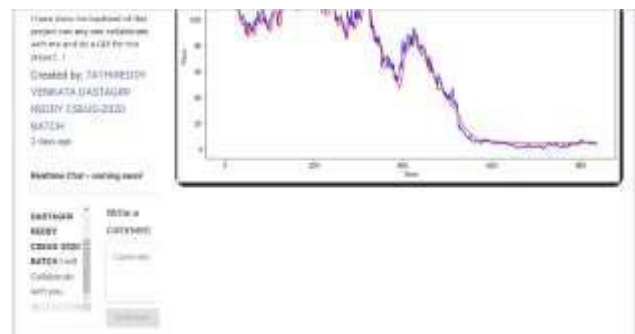


Fig. 5. Detail the single post and add a comment

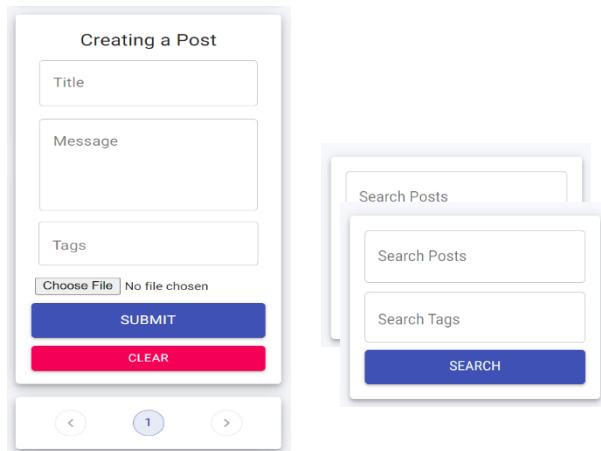


Fig. 6. Creating a Post, Pagination, and Search Post

Chatbot:

- Get an API key for OpenAI's language analysis platform.
- Compile a collection of questions raised by users and their replies.
- Create an overview of customers-posted issues and their solutions.
- Connect the chatbot to the publishing app using a RESTful API.
- Analyze and develop the chatbot thoroughly for increased accuracy and effectiveness.

Fig. 7 to 11 shows the user prompting and chatbot application layout.

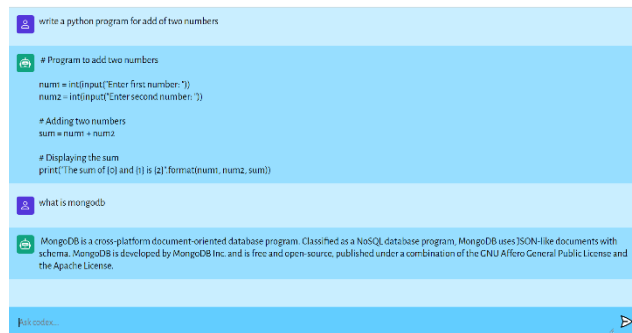


Fig. 7. Chatbot Query and answer

AI Image Generation Tool:

- Compile a dataset of photos relevant to the user's expectations.
- On the dataset, train a deep learning model such as Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs)
- Use HTML, CSS, and JavaScript frameworks such as React or AngularJS to create an internet-based user interface for the picture-generating tool.
- Build up a MongoDB database for storing data and interconnecting the backend using Node.js and Express.js.
- Completely test the image-generating tool before deploying

it to a cloud-based platform such as AWS or Google Cloud.

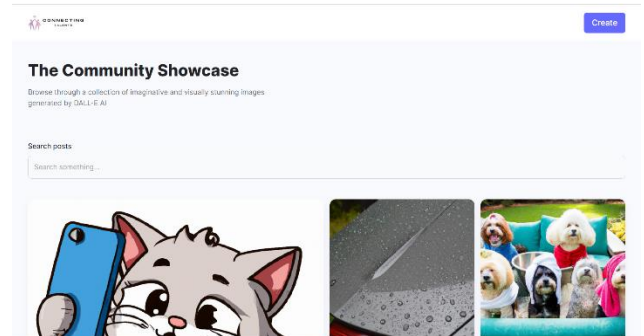


Fig. 8. Home page containing all the shared posts with the community.

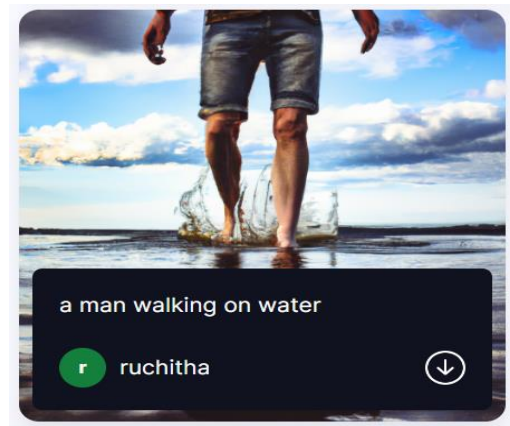


Fig. 9. A hover effect with a prompt and a download button

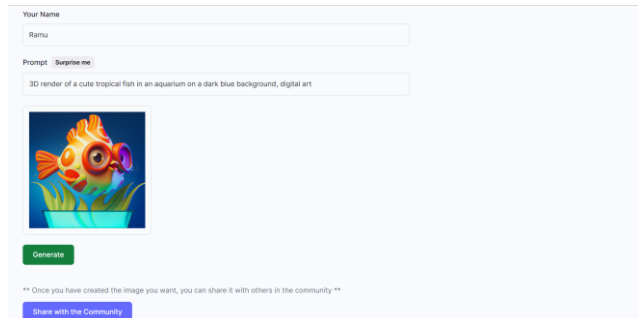


Fig. 10. Generate a post with a user prompt

Chat Application:

- Describe the app's customer necessities and capabilities.
- Make a wireframe of the customer's interface.
- Create the front end with HTML, CSS, and JavaScript frameworks such as React or AngularJS.
- Build up a database called MongoDB for data storage and integrate the backend using Node.js and Express.js.
- Use mechanisms such as OAuth or JWT to implement user identification and permission.
- Use a WebSocket-based solution such as Socket.io or Pusher to establish real-time chat features.
- Completely test the chat app before deploying it to a cloud-based environment such as AWS or Google Cloud.

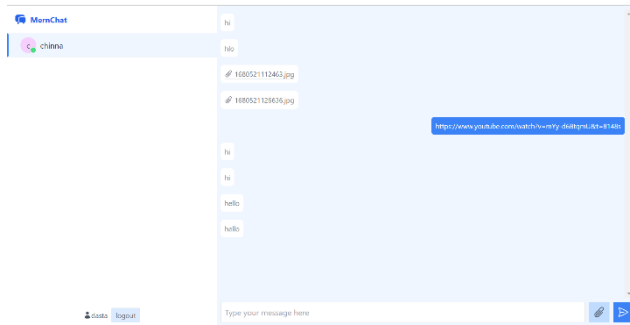


Fig. 11. Chat application Startup Funding

- Define the app's user needs and functionality.
- Create and implement a wireframe of the app's user interface.
- Build the front end using HTML, CSS, and JavaScript frameworks such as React or AngularJS.
- Merge the backend using Node.js and Express.js, and create a blockchain on the Ethereum network for transactions involving cryptocurrency.
- Establish authorization and identification of users using technologies such as OAuth or JWT.
- Create an intelligent agreement using the programming language known as Solidity to handle the funding for startups process.
- Thoroughly test the application before deploying it to a cloud-based platform such as AWS or Google Cloud.

Fig.12 to 14 represent the startup and the funding representation.



Fig. 12. All Startup's



Fig. 13. Detail about a single startup

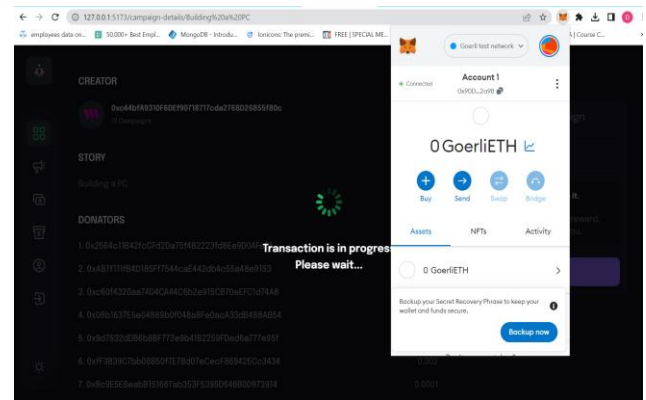


Fig. 14. Funding (Transaction of ETH to a startup)

IV. PROPOSED SYSTEM

The suggested system comprises of five applications: a posting application, a chatbot, an AI picture production tool, a chat application, and a startup finance application.

Users may utilize the posting application to make and publish posts on a variety of topics. The chatbot program offers customers an automated conversational interface that may assist them with a variety of questions. Users may utilize the AI picture production tool to produce images depending on the prompts provided by the program. The chat program allows users to converse in real-time using text, photos, and files. Finally, the startup fundraising tool allows users to fundraise for their ideas by writing a post and collecting donations via a Bitcoin wallet.

To guarantee seamless operation, the suggested system makes use of a variety of technologies and techniques. The posting app is developed using Node.js, Express.js, and MongoDB, meanwhile, the chatbot app is driven by OpenAI's API. GANs are used in the AI picture production tool, while Socket.io is used in the chat application for real-time communication. Finally, for safe and decentralized transactions, the startup funding tool makes use of the Ethereum blockchain and the MetaMask wallet.

The suggested system is developed with the needs of users in mind, with an emphasis on usability and accessibility. The posting application has a simple and easy-to-use interface that enables users to effortlessly make and manage postings. The chatbot program provides users with an automated conversational interaction that may assist them with a variety of inquiries without requiring human participation. The AI picture production tool enables users to make photos by merely supplying instructions, making it a simple process. The chat program is user-friendly, allowing users to connect in real-time using text, photos, and files. Finally, the startup finance tool allows users to obtain cash for their initiatives easily and securely, allowing them to focus on their projects rather than bothering about financial restraints.

Overall, the proposed system offers a full platform for online communication, cooperation, and financing. Its user-centric approach, along with cutting-edge technology and tools, makes it a great alternative for modern users who want smooth and hassle-free online solutions for a variety of purposes.

V. RESULT

A group of students tested the posting application and were able to effectively develop and submit material to the site. They could browse and engage with various articles, and the search option was found to be successful in locating relevant topics.

The chatbot had been evaluated by a group of users, who considered it to be user-friendly and useful in addressing their questions. The replies were likewise accurate, demonstrating that the NLP methods utilized were competent at comprehending user input.

The chat program was tested with a set of users who were able to share messages, photographs, links, and files effectively. The chat clubs and private messaging features were judged to be useful in facilitating user communication. Fig. 15 and 16 represent the Users and Posts in MongoDB for set of Learners

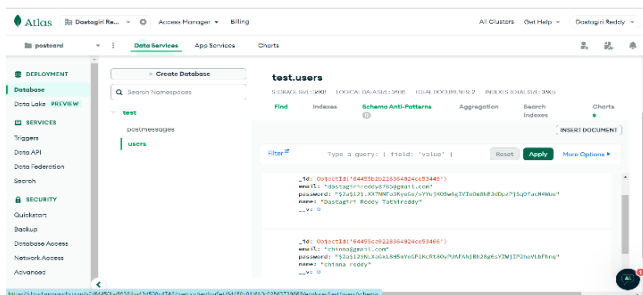


Fig. 15. Users in MongoDB

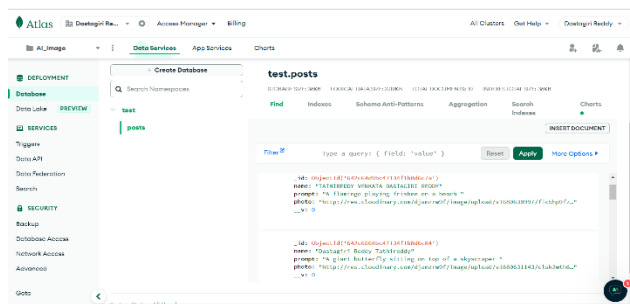


Fig. 16. Posts in MongoDB

A set of learners who were able to effectively construct and publish financing requests assessed the startup funding tool. Users who wanted to contribute to the projects may do so using the MetaMask wallet, and the sum that was raised for each project was logged and shown on the site.

Overall, the findings show that the produced apps were successful in meeting their objectives and offering a smooth user experience.

VI. CONCLUSION AND FUTURE WORK

Finally, the offered initiatives have shown the ability to increase communication, cooperation, and productivity in a variety of disciplines. The blogging app allows users to convey information and thoughts, whilst the chatbot delivers personalized help and customer care. Users may generate bespoke photos using the AI image-generating tool, and the chat program allows for real-time discussion and file sharing.

Students may use the startup financing scheme to support their assignments and bring their unique ideas to reality.

About future work, these projects have the potential for additional development and refinement. The posting application, for example, may benefit from new features like tagging and search capabilities, whilst the chatbot could be taught with more sophisticated and diverse data to increase its accuracy and reactivity. More advanced techniques and a broader variety of customization choices might be added to the AI picture-generating tool. The chat software could be connected with fresh platforms and gadgets, and the startup fundraising initiative could be expanded to more consumers with more diversified funding possibilities.

These initiatives' continued study and development might lead to even bigger gains in interaction, cooperation, and creativity.

ACKNOWLEDGEMENT

We acknowledge the support of Kalasalingam Academy of Research and Education for providing us with the necessary facilities and resources for conducting this research. I also thank the Computer Science and Engineering Department for their academic support and guidance.

REFERENCES

- Henriyan, D., Subiyanti, D. P., Fauzian, R., Anggraini, D., Aziz, M. V. G., & Prihatmanto, A. S. (2016, October). Design and implementation of web-based real time chat interfacing server. *In 2016 6th International Conference on System Engineering and Technology (ICSET)* (pp. 83-87). IEEE.
- Dipina Damodaran, B., Salim, S., & Vargese, S. M. (2016). Performance evaluation of MySQL and MongoDB databases. *Int. J. Cybern. Inform. (IJCI)*, 5.
- Sahu, A., Vishwakarma, V., Yadav, S., & Anand, A. (2023). Web-Based Chat Application Using REACT. *Available at SSRN 4366804*.
- Liang, L., Zhu, L., Shang, W., Feng, D., & Xiao, Z. (2017, May). Express supervision system based on NodeJS and MongoDB. *In 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)* (pp. 607-612). IEEE.
- Wang, J., Dou, W., Gao, Y., Gao, C., Qin, F., Yin, K., & Wei, J. (2017, October). A comprehensive study on real world concurrency bugs in Node.js. *In 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 520-531). IEEE.
- Saundariya, K., Abirami, M., Senthil, K. R., Prabakaran, D., Srimathi, B., & Nagarajan, G. (2021, May). Webapp service for booking handyman using mongodb, express JS, react JS, node JS. *In 2021 3rd International Conference on Signal Processing and Communication (ICSPSC)* (pp. 180-183). IEEE.
- Abdullah, H. M., & Zeki, A. M. (2014, December). Frontend and backend web technologies in social networking sites: Facebook as an example. *In 2014 3rd international conference on advanced computer science applications and technologies* (pp. 85-89).

IEEE.

- Laksono, D. (2018, August). Testing spatial data deliverance in SQL and NoSQL database using NodeJS fullstack web app. *In 2018 4th International Conference on Science and Technology (ICST)* (pp. 1-5). IEEE.
- Nagarhalli, T. P., Vaze, V., & Rana, N. K. (2020, March). A review of current trends in the development of chatbot systems. *In 2020 6th International conference on advanced computing and communication systems (ICACCS)* (pp. 706-710). IEEE.
- Patil, M. M., Hanni, A., Tejeshwar, C. H., & Patil, P. (2017, February). A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retrieval operations using a web/android application to explore load balancing—Sharding in MongoDB and its advantages. *In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (pp. 325-330). IEEE.