International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Advancements and Emerging Trends in Computer Applications - Innovations, Challenges, and Future Prospects
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

A Web-Based Insurance Quotation and User Management System using MERN Stack

Divya Ghodmare

PG Student, Department of Computer Application,
G. H. Raisoni Amravati University, Nagpur, Maharashtra, India

ABSTRACT

The digital transformation of the insurance industry has significantly altered how customers interact with insurers and obtain coverage. Traditional methods, which often require manual intervention and face-to-face meetings, are being replaced by more efficient, self-service online platforms. This paper presents a web-based insurance quotation and management system developed using the MERN stack (MongoDB, Express.js, React, Node.js). The system provides users with an easy-to-use platform to generate dynamic insurance premium quotes based on user-specific inputs, such as age and insurance type, and securely stores user data and quote history. The backend leverages Node.js and Express.js for efficient handling of user requests, while MongoDB is used to store user profiles and quotes. Security is ensured through the use of JSON Web Tokens (JWT) for user authentication and authorization. Additionally, the system uses bcrypt for secure password hashing to protect user credentials.

The system simplifies the insurance process by enabling users to interact with an automated platform, reducing human errors and time spent on manual tasks. This system also offers scalability, making it an ideal solution for insurance providers to integrate into their services. The paper highlights the architecture, features, and technologies behind the system, showcasing how modern web technologies can be used to enhance the user experience and streamline insurance quoting and management.

KEYWORDS: *Insurance system, MERN stack, Node.js, MongoDB, Express.js, React, JWT, user management, premium calculation, web application, API, password security, quotation system.*

INTRODUCTION

The insurance industry plays a crucial role in providing financial security, but traditional methods of obtaining insurance quotes often involve lengthy processes, paperwork, and face-to-face interaction. With the rise of digital platforms, consumers now expect quicker, more efficient ways to interact with insurance companies. The ability to generate quotes online in real-time based on user data can make the process much faster and more user-friendly.

This paper presents a web-based insurance quotation system designed to streamline the insurance process by allowing users to generate quotes online. Using modern web technologies like the MERN stack (MongoDB, Express.js, React, Node.js), the system provides users with a simple and secure interface where they can input their personal information, select their desired insurance type, and receive an instant premium quote.

The platform also includes secure user registration and login systems, ensuring that users can save their quote history and access it later. By implementing a backend using Node.js and Express.js, the system ensures scalability and efficiency in handling user data and quote generation. Additionally, the application uses MongoDB as a NoSQL database for storing user data and quotes, while employing JWT (JSON Web Tokens) for user authentication.

This project aims to provide an easy-to-use, reliable, and scalable solution for both insurance providers and consumers, enhancing the insurance experience through automation and accessibility.

RELATED WORK

In recent years, several digital platforms have emerged to simplify the insurance process by offering online policy comparisons, quote generation, and digital customer support. Websites like **PolicyBazaar**, **Coverfox**, and **Insurancedekho** provide users with the ability to compare insurance plans from various providers and receive quotes instantly. These platforms have proven that users prefer quick and transparent insurance services online.

Most of these platforms are built using modern web technologies and integrate APIs to fetch data from multiple insurance providers. However, they are large-scale systems with complex architectures and often require third-party integration. In contrast, many academic projects and small businesses focus on building basic insurance systems that handle core functions such as premium calculation, user data storage, and simple login systems.

Research papers and student projects found on platforms like GitHub and IEEE have implemented similar ideas using languages like Java, PHP, or Python. However, very few of them make use of the full **MERN stack**, which is known for building fast, scalable, and secure web applications.

What sets this project apart is its focus on creating a fully working **insurance quote generator with login and data saving features** using only open-source tools, and a focus on simplicity, usability, and data security. By using JWT for authentication and MongoDB for storing dynamic quote data, this system combines best practices from both industry and academic implementations.

KEY FEATURES

1. User Registration and Login

Users can create an account with their name, email, and password. Secure login system using hashed passwords.

2. JWT Authentication
JSON Web Tokens are used to keep users logged in securely and protect private routes.

3. Insurance Quote Generator
Users can select insurance type (e.g., health, life, auto) and enter their age to get an instant premium quote.

4. Quote Calculation Logic
A backend formula calculates the premium amount based on the user's age and selected insurance type.

5. Quote History Storage
Each user's generated quotes are saved in the database, allowing them to view past quotes any time.

6. MongoDB Database
Stores user details and quote data efficiently using collections.

7. RESTful API
The backend follows REST architecture, making it easy to connect with any frontend or mobile app.

8. Scalable and Modular Backend
The Node.js and Express.js backend is organized in routes and controllers for easy updates and scaling.

9. Error Handling and Validation
Input data is validated, and meaningful error messages are provided for better user experience.

10. Frontend Connectivity (optional)
The backend can be connected to any HTML, React, or other frontend to display user interface.

DATA SOURCES
This insurance web project uses data that is either entered by the user or generated by the system itself. Below are the main data sources used:

1. User Input
- Name, email, and password (during registration)
- Insurance type and age (for quote generation)

2. System-Generated Data
- Quote amount calculated based on input
- Quote creation date and time
- Unique user ID associated with each quote

3. Database (MongoDB)
- Stores all user accounts and login details (securely hashed passwords)
- Stores all generated quotes with references to the user who created them

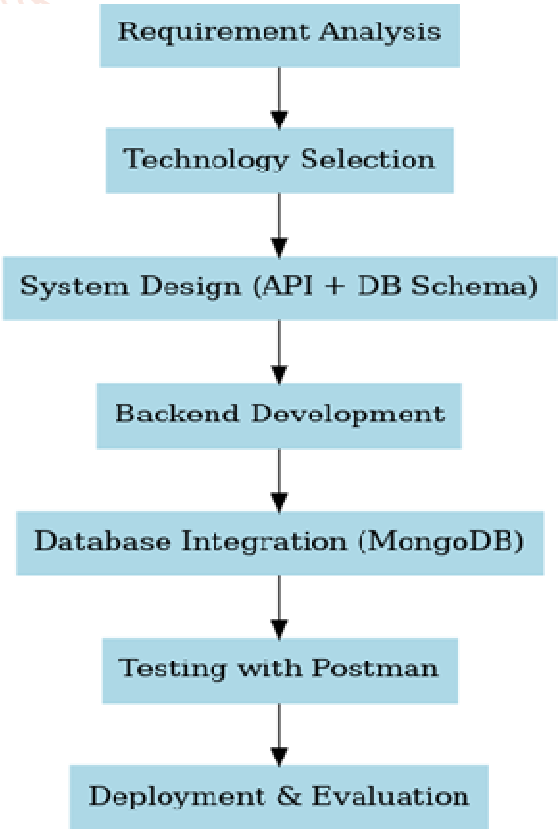
4. Internal Logic (Backend)
- The premium calculation is done using fixed formulas based on insurance type and age.
- No third-party APIs or external datasets are used in this version.

RESEARCH METHODOLOGY
The development of the insurance web application followed a structured and step-by-step approach. The process began with **requirement analysis**, where the core needs of the system were identified — including user registration, login functionality, secure authentication, quote generation, and data storage. After understanding the requirements, the **MERN stack** (MongoDB, Express.js, React (optional), and Node.js) was chosen due to its efficiency in creating full-stack web applications using JavaScript across both frontend and backend.

Once the technologies were selected, the **system architecture** was designed. This included separating the backend into modules for authentication and quote management, using **RESTful API principles**. The database schema was modeled using **Mongoose** to define collections for users and quotes. The **backend development** phase involved implementing secure user registration and login with hashed passwords (using bcrypt), and token-based authentication (using JWT) to protect user sessions.

The **quote generation module** was developed to calculate premiums dynamically based on user input (age and insurance type). The calculated quote, along with user data, was stored in **MongoDB**. All backend functions were tested using **Postman**, ensuring reliable API responses, input validation, and error handling.

The final stage involved running the backend server using **Visual Studio Code and Node.js**, evaluating the project based on performance, accuracy, and security. This structured methodology ensured the project was built logically, securely, and efficiently.



RESULT
The insurance web application was successfully developed and tested using the MERN stack. The system allows users to register and log in securely, enter their age and insurance type, and receive an instant insurance premium quote. The backend processes user inputs using a predefined calculation logic and stores both user data and quote history in the MongoDB database.

During testing, the application was able to:
- Register and authenticate users with secure password encryption using bcrypt.

- Generate accurate insurance quotes based on different age ranges and insurance types.
- Store and retrieve user quote history linked to each individual account.
- Maintain session security using JSON Web Tokens (JWT), preventing unauthorized access to user data.

The application performed well in terms of speed and reliability during API testing in Postman. Quotes were generated instantly, and all data was saved correctly in the database. The modular structure of the backend makes it easy to scale and integrate with a frontend interface like React or HTML.

The project successfully meets the goal of simplifying the insurance quote process through automation and secure web technology. It provides a strong foundation for further development, including adding more insurance categories, integrating payment gateways, or using real-time data from insurance providers.

CONCLUSION

This project demonstrates how modern web technologies can be effectively used to simplify and automate the insurance quotation process. By using the MERN stack, a secure and scalable backend system was built that allows users to register, log in, and generate insurance quotes based on personal inputs such as age and insurance type. The use of technologies like Node.js, Express.js, MongoDB, bcrypt, and JWT ensured that user data was handled safely and efficiently.

The system reduces the need for manual paperwork and provides users with a fast, user-friendly experience. It also stores quote history, allowing users to access their previous calculations at any time. The modular structure and clean API design make the system easy to maintain and expand in the future.

This project lays the groundwork for a more advanced insurance management system. Future enhancements can include a complete frontend interface, admin panel, integration with real-time insurance provider APIs, and support for more complex insurance rules and premium structures.

In conclusion, the project fulfills its goal of creating a simple, secure, and efficient platform for online insurance quote generation and serves as a practical example of applying full-stack web development in real-world applications.

REFERENCES

[1] MDN Web Docs. (n.d.). *Express.js Guide*. https://developer.mozilla.org/

[2] MongoDB, Inc. (n.d.). *MongoDB Documentation*. https://www.mongodb.com/docs/

[3] Node.js Foundation. (n.d.). *Node.js Documentation*. https://nodejs.org/en/docs

[4] bcrypt. (n.d.). *bcrypt for password hashing*. https://www.npmjs.com/package/bcrypt

[5] JWT.io. (n.d.). *JSON Web Tokens Introduction*. https://jwt.io/introduction

[6] Postman. (n.d.). *API Testing Platform*. https://www.postman.com/

[7] W3Schools. (n.d.). *RESTful API Tutorial*. https://www.w3schools.com

[8] Stack Overflow. (n.d.). *Developer Discussions and Solutions*. https://stackoverflow.com/

[9] GitHub. (n.d.). *Open-source insurance system projects*. https://github.com/

[10] Tutorials Point. (n.d.). *MERN Stack Tutorials*. https://www.tutorialspoint.com/mern/index.htm