UDC 004.9

AUTOMATED REAL-TIME EMPLOYEE MANAGEMENT SYSTEM

KOMRAKOVA EVGENIYA VLADIMIROVNA,
senior lecturer
MULUMBA NDAJI JOSUE
student
Sukhoi State Technical University of Gomel

Abstract: Authentication is a critical component of any IT system, providing a frontline defense to safeguard sensitive data while ensuring that only authorized users can access application resources. For this automated employee management system project, implementing a robust and user-friendly authentication process plays a pivotal role in maintaining security and streamlining daily operations. Key words: authentication, backend, frontend, employee management system, user interface.
Key words: authentication, backend, frontend, employee management system, user interface.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ СОТРУДНИКАМИ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Комракова Евгения Владимировна,
старший преподаватель
Мулумба Ндайи Джосуе
студент
ГГТУ им. П.О. Сухого

Аннотация: Аутентификация является основополагающим компонентом любой ИТ-системы, выступая в качестве первой линии обороны для защиты конфиденциальных данных и обеспечения того, чтобы только авторизованные пользователи могли получить доступ к ресурсам приложения. В контексте этого проекта автоматизированной системы управления сотрудниками внедрение надежного и интуитивно понятного процесса аутентификации имеет решающее значение для обеспечения безопасности и бесперебойной работы повседневных задач.
Ключевые слова: аутентификация, Backend, frontend, система управления сотрудниками, пользовательский интерфейс.

Effective communication between the frontend and backend is essential for developing secure, efficient, and intuitive applications. Employing technologies like Fetch API and JWT, alongside adhering to established security best practices, enables seamless data exchange and robust user authentication. This method not only enhances the user experience but also mitigates risks from typical security vulnerabilities.

Figure 1 depicts the global backend architecture, demonstrating interactions among the client, API, and database.

The diagram in Figure 1 outlines the flow within the global backend structure. It illustrates how the client sends an HTTP request to the API, which processes it through its defined layers (Presentation, Business Logic, and Data Access). Subsequently, the API retrieves or modifies the requested data by interacting with the database and responds to the client. This representation effectively highlights the division of responsibilities within the backend, ensuring clarity and scalability in system design.

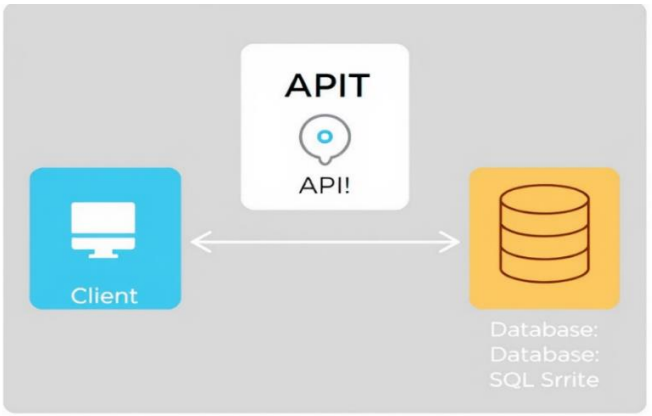

Fig. 1. Overview of global backend architecture

Figure 2 showcases the comprehensive database structure containing tables for employees, activity logs, and JWT tokens. The employees table holds user-specific information such as names and positions. Simultaneously, the activity logs table keeps a record of user actions alongside timestamps for auditing, while the JWT tokens table manages authentication tokens to maintain secure user sessions and control access privileges. This logical and efficient design supports streamlined data storage and retrieval processes.

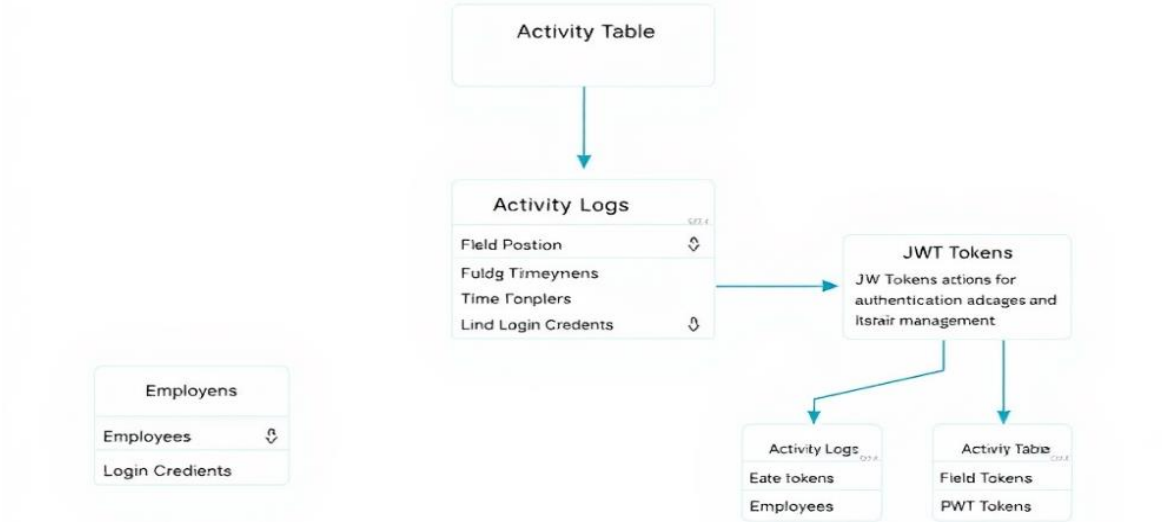

Fig. 2. Representation of database structure using JWT Tokens and activity logs

The sequence diagram in Figure 3 illustrates the step-by-step employee authentication process within the system, visualizing interactions among components such as the Frontend, Backend, Database, and final output (e.g., successful authentication).

Detailed process explanation:

– Frontend: The employee inputs their credentials.
– Frontend → Backend: Credentials are transmitted to the backend.
– Backend → Database: The backend checks the credentials in the database.
– Backend: If validated, a JWT is generated.
– Backend → Frontend: The generated JWT is sent back to the frontend.
– Frontend: The JWT is securely stored locally.
– Frontend → Backend: For subsequent requests, the frontend uses the JWT.
– Backend: The JWT is verified for validity.
– Backend → Frontend: Upon verification, requested data is sent back to the frontend.
– Frontend: The employee gains access to application resources.

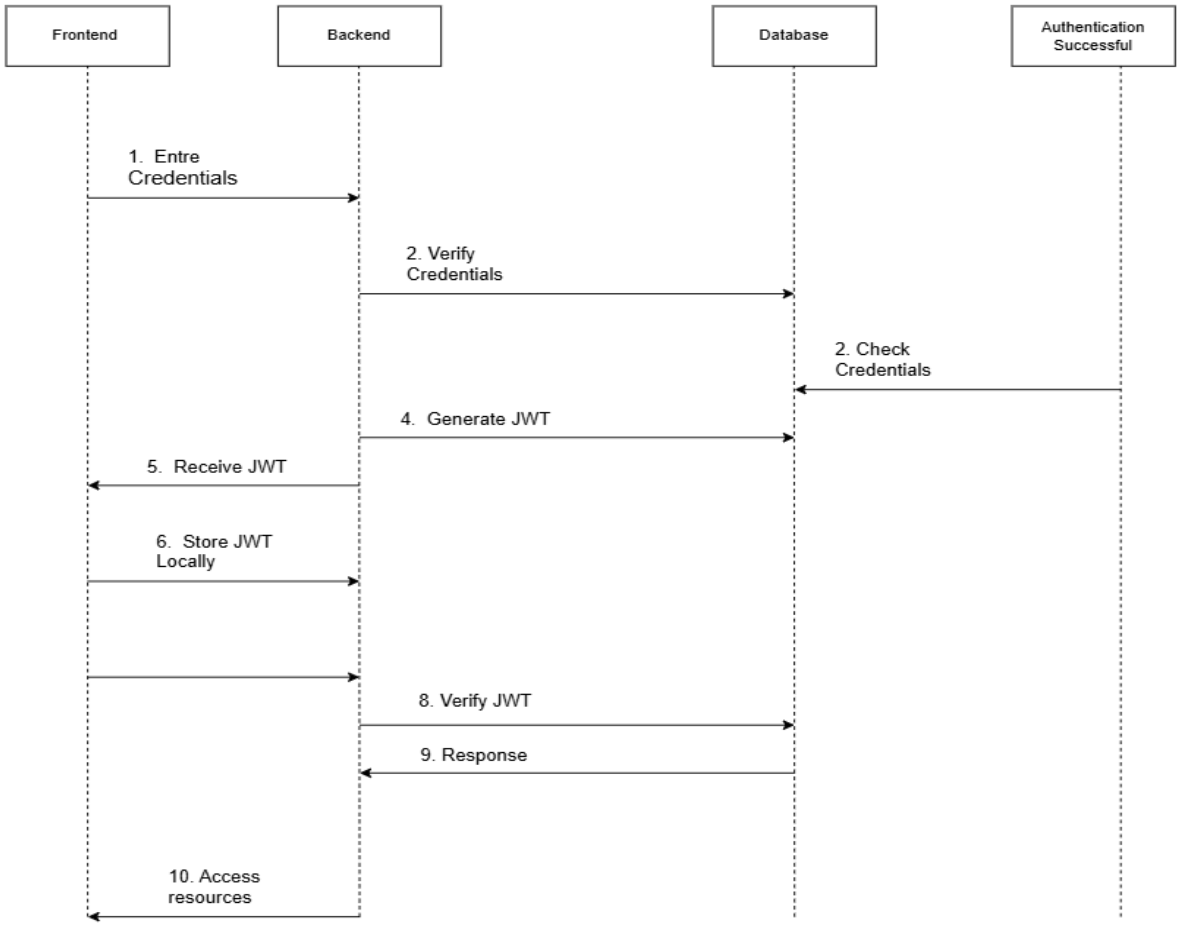

Fig. 3. Sequence diagram

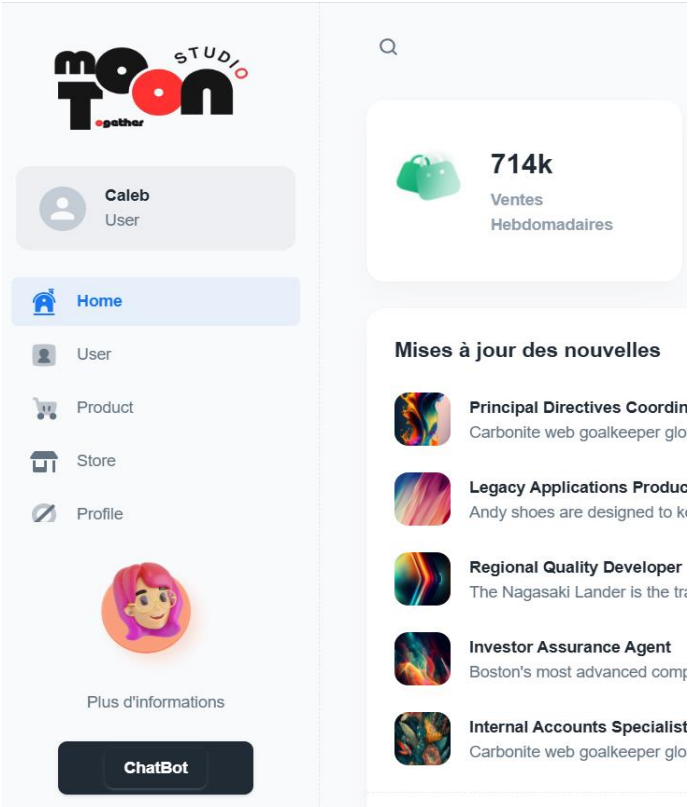

Fig. 4. User interface displayed

All data exchanges between frontend and backend modules are secured using HTTPS protocols to prevent interception or tampering during transmission. Additional safeguards implemented within this system include:

1. **HTTPS Encryption**: Ensures encrypted communication between frontend and backend to block unauthorized access or data interception.
2. **Password Hashing**: User passwords are encrypted using secure algorithms like bcrypt before being stored in the database, rendering them unreadable even in the event of a breach.
3. **Secure JWT Generation**: Each generated JWT is signed with a robust secret key to guarantee integrity. Tokens have limited validity periods to minimize potential damage from unauthorized access.
4. **Safe JWT Storage**: Tokens are securely stored on the frontend (e.g., HTTP-only cookies) to prevent exploitation through cross-site scripting (XSS).
5. **CSRF Protection**: Cross-Site Request Forgery attacks are mitigated through CSRF tokens embedded in interactions between client and server.
6. **JWT Verification**: The backend checks every token's signature for integrity before validating requests or granting access permissions.

The figure 4 shows the user interface displayed after a successful login.

Once the backend verifies the user's credentials and confirms their existence in the database, they are directed to this dashboard. The interface includes a table with user-related options such as Home, Product, Profile, and ChatBot, providing easy navigation. The layout is clean and organized, ensuring a seamless experience for the user.

References

1. Stellman, A C#/ A learner's guide to real-world programming with C# and .NET / A. Stellman, J. Greene – O'Reilly, 2024 – 795 p.
2. Price, M. C# 13 and .NET 9 – Modern Cross-Platform Development Fundamentals Ninth Edition / M. Price – Packt Publishing, 2024 – 806 p.
3. Khalfallah, H.B. Crafting Clean Code with JavaScript and React / H. B Khalfallah – Apress, 2024 – 441 p.
4. Rippon, C. ASP.NET Core 3 and React / C.Rippon – Packt Publishing, 2019 – 768 p.
5. Dyer, R. Learning MySQL and MariaDB / R.J.T. Dyer – O'Reilly, 2023 – 443 p.
6. Banks, A. React: Modern Patterns for Application Development / A.Banks, E. Porcello – O'Reilly, 2023 – 320 p.