**SIT774 | TASK 10.4 HD | S224750485**

Introduction

This project aims to develop a secure and modular web application for a sneaker store, using Node.js, Express.js, SQLite, and EJS templating. The implementation includes modern authentication methods to safeguard user data, reusable UI components for a uniform shopping experience, dynamic content rendering for product listings and inventory updates, and enhanced error handling mechanisms to ensure seamless functionality. The goal is to improve security and user experience while keeping the application maintainable and scalable for future development.

Key Features Implemented

1. Secure User Authentication Using JWT and Bcrypt

* User authentication is implemented using JSON Web Tokens (JWT) for session management.
* Passwords are hashed using bcrypt before storage in the database to ensure security.
* JWT tokens are stored in localStorage upon successful login and are used for authentication on protected routes such as the user profile page.
* A middleware function (authenticateToken) ensures that only authenticated users can access certain resources.

2. Profile Picture Upload with Formidable

* File uploads are handled using Formidable.js, enabling users to upload profile pictures during registration.
* Uploaded images are stored in the /uploads/ directory, and the file path is saved in the SQLite database.
* Profile pictures are dynamically retrieved and displayed on the user profile page.

3. Modular UI with EJS Templating

* All static HTML pages have been converted to EJS, allowing for dynamic content injection from the server.
* Reusable UI components such as:
  + Navbar (components/navbar.ejs)
  + Footer (components/footer.ejs)
  + News Modal (components/newsModal.ejs)
  + Error Handling Page (errors/error.ejs)
* This improves code maintainability and ensures UI consistency across all pages.

4. Personalized User Greeting

* After login, the application displays a personalized greeting message, dynamically retrieving the user's name from the JWT token.
* The implementation decodes the JWT token on the client side, extracts the user's name, and updates the UI accordingly.

5. Improved Error Handling and Routing

* Implemented custom error handling for:
  + 404 Not Found pages for undefined routes.
  + 500 Internal Server Error pages for handling unexpected server issues.
* Middleware is used to handle invalid requests and return structured JSON error responses for API endpoints.

6. Background Image Scrolling for Enhanced User Experience

* Implemented background image scrolling on the login and registration pages for improved UI aesthetics.
* This is achieved through CSS background properties and is optimized for responsive design.

7. Interactive News Section with Modal Popups

* Developed a news section that uses Bootstrap modals for article previews.
* Users can click "Read More" to open a modal popup containing detailed news content and images.
* The modal system is modularized into components/newsModal.ejs for reusability.

8. User Profile Page with Secure Data Retrieval

* The profile page dynamically retrieves user details from the database, ensuring that only authenticated users can access their information.
* User information such as name, email, phone number, address, date of birth, and profile picture is displayed.

9. Logout Functionality

* Users can securely log out, which clears their JWT token from localStorage and redirects them to the login page.
* This ensures that user sessions are properly managed and access is restricted after logout.

10. Optimized Form Design and Validation

* Bootstrap-based form designs for login and registration pages ensure a modern and responsive UI.
* Implemented client-side and server-side validation for email, phone numbers, and passwords, improving data integrity and user experience.

Conclusion

This project demonstrates a comprehensive approach to full-stack web development, integrating secure authentication, modular UI components, interactive UI elements, and robust error handling mechanisms.

By implementing JWT-based authentication, file uploads, dynamic UI interactions, and EJS templating, the system ensures:

* Enhanced security through password hashing and token-based authentication.
* Improved maintainability through modular and reusable components.
* Scalability with a structured and extendable architecture.

This solution follows modern web development best practices, providing a secure, interactive, and user-friendly application that can be expanded with additional features in the future.