Secret & Credential Audit Report

Objective:
To verify that no credentials, API keys, or sensitive environment variables are hardcoded in the repository, and that all secrets are managed securely via the **.env** file or external configuration.
Result Summary:
A full scan was performed across the codebase for potential sensitive keywords (e.g., password, token, auth, secret, credential, apikey). The output file (*scan_grep_results.txt*) was analyzed for violations. **Findings:**
- No hardcoded API keys, authentication tokens, or credentials were found.
- References to sensitive data (passwords, tokens, auth) are part of legitimate code logic (authentication, schema definitions, frontend UI text).
- Environment variables such as JWT_SECRET and NGROK_AUTHTOKEN are correctly loaded from the **.env** file or Docker secrets.
- The **.env** file is excluded from version control, ensuring that no secrets are stored in the repository.
Compliance Status:
■ Fully compliant with the rule: “Any credentials, API keys, or environment variables must be set inside a .env file; no credentials or API keys may be in the git repository.”

Recommendations:
1. Continue to manage all secrets in the **.env** file or Docker secrets.
2. Periodically rerun this scan to ensure compliance before each release.
3. Avoid echoing or logging environment variables to stdout in production.
4. Review new dependencies for potential secret exposure.

Conclusion:
No evidence of credential or API key exposure was detected in this repository. The project adheres to secure secret management practices.