Design Choices

- **Database Structure:** The database was expanded to support a finer, more detailed report structure. This is serviced by getall handlers that return comprehensive item details.
- **Model Centralization:** All data models are consolidated within internal/data/models.go. This improves organization, simplifies passing the database context to the models, and avoids a single, large, unmanageable file.
- **User Soft Delete**: A soft delete policy was implemented for users. This preserves historical user records, which are valuable for auditing and analysis.
- **Data Integrity:** Delete handlers were intentionally omitted (despite an existing model) to prevent the accidental deletion of critical records, prioritizing data retention.
- **Type Definitions & API Documentation:** A dedicated types.go file is used for shared type definitions. Swagger AI is integrated to generate clean, professional documentation for all API handlers and routes.
- **Role-Based Access Control (RBAC):** The system uses roles to manage user access. Permissions are assigned to roles, and roles are assigned to users. This simplifies permission management, prevents overwhelming users with individual permissions, and reduces the risk of accidental privilege escalation.

Limitations

- **Time Constraints:** Significant time constraints limited the project's scope, preventing the implementation of more granular features.
- **Feature Scope:** The desired level of detail, particularly for reporting functions, was not fully achieved due to time limitations.

Future Improvements

- **Enhanced Reporting:** Implement more detailed and granular reporting options with advanced filtering.
- **Data Lifecycle Management:** Re-evaluate and design safe-guarded handlers for deleting non-critical records, while maintaining protection for essential data.