

AutoDBA Web – Error Handling Strategy

1. Authentication Errors

- Missing credentials:

Return 400 Bad Request with message: 'Email and password are required.'

- Invalid email format:

Return 400 Bad Request using validator/regex. Message: 'Invalid email format.'

- Duplicate email on signup:

Catch unique constraint violation. Return 409 Conflict with 'Email already registered.'

- Incorrect login credentials:

Return 401 Unauthorized with 'Invalid email or password.'

- Expired or malformed JWT:

Return 401 Unauthorized. Frontend should logout and redirect to login.

2. Database Connection Errors

- Missing DB credentials:

Return 400 with 'All DB connection fields are required.'

- MySQL connection refused:

Catch connection error, return 500 with 'Unable to connect to database. Check credentials or server availability.'

- Authentication failure:

Return 403 Forbidden with 'Invalid database username or password.'

- Database does not exist:

Return 404 Not Found with 'Specified database not found.'

- Timeout connecting to DB:

Return 504 Gateway Timeout with 'Connection timed out. Try again later.'

3. Audit Engine Errors

- Script file missing:

Return 500 with 'Audit module not found: audit_schema.py'. Log full path error.

- Python execution fails (stderr):

Return 500 with 'Audit process failed'. Log stderr output.

- Malformed JSON output:

Return 500 with 'Unexpected output from audit script'. Mark audit as failed.

- Partial success (1 or 2 modules fail):

Return 207 Multi-Status with available results and list failed modules.

4. Frontend Handling Strategy

- 401 Unauthorized:

Auto-logout the user, clear token, redirect to login. Show toast: 'Session expired. Please log in again.'

- 500 Internal Server Error:

Show generic error toast: 'Something went wrong. Please try again later.'

- Validation errors (400/409):

Show inline errors in forms using toasts/snackbars (e.g. notistack).

- Partial audit results (207):

Display available tabs. Show banner: 'Some modules failed to run.'

- DB connection failures:

Keep user on ConnectDBPage. Display reason using detailed toast.

5. Summary

This strategy ensures robust error communication across the frontend, backend, and audit engine. Clear feedback is delivered to the user through toast messages and inline form validation, while the backend logs all technical errors for debugging. All API responses follow consistent status code conventions.