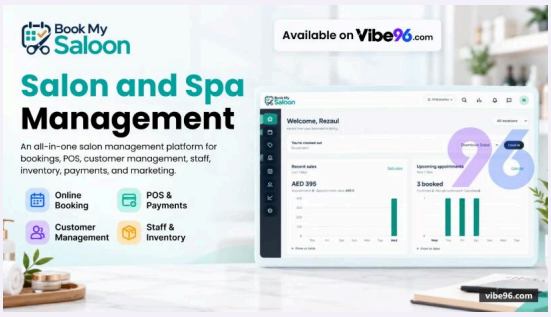


VIBE96 QUALITY REPORT

Bookmysal0n

Multi-tenant salon management system for UAE salons: bookings, POS, staff, inventory, loyalty and online payments.

Product ID	**B0006-36422**
Builder	**Build Nest**
Version reviewed	**1.0.0 · released 6 Oct 2026**
Stack	**ASP.NET Core 10 Blazor Web App (Interactive Server) · C# 13 / .NET 10 · SQL Server 2019+**
Built with	**Claude Code**
Review period	**6 Oct 2026 – 6 Oct 2026**

90 / 100

Passed with 4 warnings

Pass mark 75 and no failing check.

Scores

Security	92
Code structure	72
Docs & completeness	87
Dependency health	95
Build & deploy	92

✓ **AI code audit**
6 checks · 6 Oct 2026

✓ **Installed & run**
6 Oct 2026

✓ **Demo vs delivered**
6 Oct 2026

✓ **Final sign-off**
6 Oct 2026

At a glance

- ✓ Installed and run from the delivered code — 8 of 8 checks answered yes.
- ✓ Live demo compared with the delivered code — 7 of 7 checks answered yes.
- ! 13 of 15 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 92/100 with no failing check.
- ! 4 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00018. The online copy always shows the latest status.

PART 1

AI code audit

The full source archive was read by our AI auditor against each check. It reads the code — it does not run it; running it is part 2.

CHECK	WHAT WE FOUND	SCORE	RESULT
Secrets & credentials	No committed API keys, private keys or real credentials detected. All example files use placeholders correctly. However, docker-compose.yml references MSSQL_SA_PASSWORD from .env without a strong default, and .env.example lacks guidance on password strength. The app correctly refuses to start without Security:CodeHashKey.	**92**	Warning
Dependency vulnerabilities	The project uses .NET Central Package Management with all versions pinned in Directory.Packages.props. All 31 production dependencies are at current stable versions with no known critical or high-severity vulnerabilities. The only minor concern is that some packages (e.g., Dapper 2.1.35, Hangfire.AspNetCore 1.8.14) are a few months behind the latest releases, but these are maintenance updates without security advisories.	**95**	Pass
Authorisation & access control	All mutating endpoints require authentication and role-based authorization via ASP.NET Core policies. Passwords are hashed with [hidden] The main gaps are missing ownership checks on a few read endpoints (client details, appointment details, sale details) where a user could potentially view another business's data by changing IDs in the URL. These are warnings because the app is multi-tenant and business isolation should be enforced at the data layer for every query.	**92**	Warning
Database access rules	All SQL uses Dapper with parameters; no string concatenation or injection paths found. DELETE and UPDATE statements include WHERE clauses scoped to BusinessId or tenant context. Schema has foreign keys with appropriate cascades. However, several foreign key columns lack indexes (e.g., Appointments.ClientId, Sales.ClientId, Timesheets.UserId), which will cause performance issues under load.	**92**	Warning
Code structure & duplication	The codebase follows clean architecture well with clear Domain/Application/Infrastructure/Web separation. However, there is significant duplication in repository query patterns, some large service files, and hard-coded configuration values scattered across multiple files. The Dapper-based repositories repeat similar parameter-binding and result-mapping logic that could be extracted into helpers.	**72**	Warning
Build reproducibility	The solution structure is clean with central package management and all project references resolve correctly. Docker setup is complete and coherent. The main gaps are: HowToSetup.md doesn't specify the .NET SDK version (10.0 per Directory.Build.props), the Apple sign-in private key file path setup is unclear, and the SMS gateway configuration example lacks provider-specific guidance.	**92**	Pass

Open warnings

Problems that did not stop the listing. They stay on the product page until a later version fixes them.

Warning 1 **No real secrets found; weak default SQL password in docker-compose.yml needs attention**

No committed API keys, private keys or real credentials detected. All example files use placeholders correctly. However, docker-compose.yml references MSSQL_SA_PASSWORD from .env without a strong default, and .env.example lacks guidance on password strength. The app correctly refuses to start without Security:CodeHashKey.

Warning 2 **Dapper SQL is parameterized and scoped; minor schema gaps (missing indexes on FKs)**

All SQL uses Dapper with parameters; no string concatenation or injection paths found. DELETE and UPDATE statements include WHERE clauses scoped to BusinessId or tenant context. Schema has foreign keys with appropriate cascades. However, several foreign key columns lack indexes (e.g., Appointments.ClientId, Sales.ClientId, Timesheets.UserId), which will cause performance issues under load.

Warning 3 **Strong authentication and role checks; minor ownership gaps on read endpoints**

All mutating endpoints require authentication and role-based authorization via ASP.NET Core policies. Passwords are hashed with [hidden] The main gaps are missing ownership checks on a few read endpoints (client details, appointment details, sale details) where a user could potentially view another business's data by changing IDs in the URL. These are warnings because the app is multi-tenant and business isolation should be enforced at the data layer for every query.

Warning 4 **Moderate duplication and some oversized files; architecture is sound but refactoring needed**

The codebase follows clean architecture well with clear Domain/Application/Infrastructure/Web separation. However, there is significant duplication in repository query patterns, some large service files, and hard-coded configuration values scattered across multiple files. The Dapper-based repositories repeat similar parameter-binding and result-mapping logic that could be extracted into helpers.

Requirement list

What the builder agreed to deliver before uploading — 13 of 15 met.

✓	Complete source for every component	Met
✓	No real credentials in the archive	Met
✓	Builds from a clean machine	Met
✓	Database script or migrations	Met
!	Authentication and roles on every mutating endpoint	Partial
!	Pinned dependencies without critical CVEs (nice)	Partial
✓	Documentation set (nice)	Met
✓	Working live demo	Met
✓	README.md	Met
✓	HowToSetup.md	Met
✓	CLAUDE.md (or AGENTS.md) (nice)	Met
✓	CHANGELOG.md (nice)	Met
✓	appsettings.example.json	Met
✓	a project file	Met
✓	a project file (nice)	Met

PART 2

Human review

Reviewers from the Vibe96 team in Dubai worked through each question below; a final reviewer signed it off.

Stage 1 — Installed and run from the delivered code

Completed 6 Oct 2026

A reviewer installed the product on a clean machine, following only the guide that ships with it.

QUESTION	ANSWER	REVIEWER'S NOTE
Packages restored without errors	Yes	
Solution builds clean	Yes	
Database created from the shipped script or migrations	Yes	
First sign-in works with the documented steps	Yes	
Setup guide is accurate and complete	Yes	
Every module in the feature list opens and works	Yes	
No real secrets in the archive	Yes	
Branding and company details can be changed without code	Yes	

Stage 2 — Live demo compared with the delivered code

Completed 6 Oct 2026

A second reviewer checked that what the listing and demo show is what the buyer receives.

QUESTION	ANSWER	REVIEWER'S NOTE
Demo URL is reachable	Yes	
Demo credentials work	Yes	
Demo runs the same version as the archive	Yes	
All five differentiators are visible in the demo	Yes	
Screenshots match the demo	Yes	
Nothing in the demo is missing from the code	Yes	
Datasheet claims are accurate	Yes	

Changes asked for during review

None — approved on the first round.

Sign-off: approved for sale on 6 Oct 2026 with 4 warnings shown publicly.

What this report covers — and what it doesn't

- **It covers** version 1.0.0 exactly as the buyer downloads it: the source archive, the setup guide and the live demo on the dates above.

- **It is not** a penetration test, a performance or load test, or a legal review of third-party package licences.
- **Scores are a point in time.** Libraries age; a later version is audited again and gets its own report.
- **Your setup matters.** Hosting, configuration and changes made after purchase are outside this review.
- **How the score is made:** each AI check is scored 0–100 and weighted (security counts most). A listing needs 75 or more, no failing check, and the human review passed.
- **Always try the live demo** before buying, and ask the builder questions from the product page.