

Delivered.

Production Review: Elite Fitness

Prepared for
Elite Fitness

App
Elite Fitness

Date
August 31, 2026

Reviewed by
Jay Lee, Delivered

We reviewed the supplied copy of Elite Fitness on our own computers. We did not touch a live app or any real customer data.

01 SUMMARY

Elite Fitness is **not ready to launch**.

We found **6 Major and 4 Minor problems**.

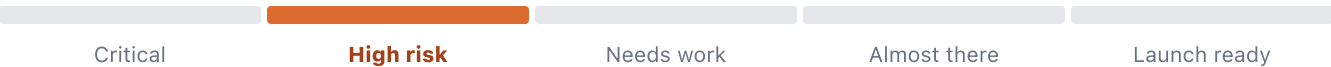
The app's basic build works, but several rules that run the club can disagree or accept impossible records.

The biggest problems are:

1. A new club cannot create its first Owner safely without relying on the documented demo account.
2. Member access, Payments, and Membership records can disagree, so the club can give away visits or reject someone who already paid.
3. The schedule can hide current Classes, move times, or accept bookings that put the same Trainer, Member, room, or Class place in two places at once.

The normal login and main sample workflows work. The app also passed its production build and all seven automated tests. Fix the 6 Major problems before launch, then handle the 4 smaller problems before the club depends on those parts of the app.

Launch readiness: High risk



0

CRITICAL

6

MAJOR

4

MINOR

02 HOW WE REVIEWED THE APP

We read how Elite Fitness was built, ran the app, and used it as gym staff would. We followed 13 important staff journeys and completed the full review checklist. We then tested each possible problem to make sure it was real. Any test that changed data used a throwaway copy of the database.

- ✓ Built the release version and ran the app's code checks, database setup, and automated tests
- ✓ Created test gym records and checked what the app actually saved
- ✓ Used the Owner and Front Desk areas through the same screens staff would use
- ✓ Tried actions at the same time, forced controlled failures, and checked what a person could do by going around the screens

Everyday problems can happen during normal work. Occasional problems need a particular situation, such as two actions happening at once or a service failing. Misuse problems require someone to go outside the normal screens on purpose. Critical, Major, and Minor describe how much harm the problem can cause. Easy, Medium, and Hard describe how difficult deliberate misuse would be. Visual proof means a screenshot appears later in this report.

SEVERITY

	Critical	Someone can gain broad control, expose highly sensitive information, or cause damage that would be very hard to recover from.
	Major	An important part of the business can fail, important records can become wrong, or the wrong person can gain access.
	Minor	The damage is limited, or staff have a reasonable way to work around it for now.
	Proof	This finding has a screenshot in Selected proof.

03 WHAT WE FOUND

Everyday problems

These can show up during ordinary use under common circumstances.

MAJOR

E01

A new club has no safe way to create its first Owner

After a clean setup, nobody can add Staff because that job already requires an Owner. Public signup does not create a Staff account. The demo Owner is fine for local testing, but its email and password are shown in the app and README, so it cannot be the starting account for a real club. A real deployment needs a separate one-time Owner setup before anyone can use it safely.

MAJOR

E02

Membership, Payment, and entry records can disagree

The Foundations Plan promises 12 visits, but we checked the same Member in 14 times and every Check-in worked. In another test, paying an overdue Invoice activated the Membership but still left the Member marked Past Due and blocked at the door. The app can also hold two current Memberships for one person or mark a draft Invoice paid. The club cannot rely on one clear answer about what a Member bought or whether they may enter.

MAJOR

E03

After enough old Classes exist, Staff can no longer see the current schedule

When there are more than 100 older Class Sessions, Schedule shows the oldest 100 and leaves out newer Sessions. Staff can still book the old Sessions if they were never marked finished or cancelled, even years later. The club can lose the schedule it needs to run today's Classes.

MINOR

E04

Some Staff can begin forms they cannot finish

On an iPhone SE-sized screen, Add Member places both Close and Submit outside the area Staff can reach. Front Desk workers and Trainers can also complete Schedule Class, only to learn at the end that their role is not allowed to save it. Staff have workarounds, but the app wastes their time before explaining the limit.

Occasional problems

These need less common data, timing, retries, or account changes, but they can still happen during ordinary use.

MAJOR 

O01

A Front Desk worker can see an Owner's Reports after using the same browser tab

An Owner can open Reports, sign out, and hand the same tab to a Front Desk worker who signs in. The earlier revenue and sales report can stay visible. The app correctly refuses the Front Desk worker's new request, but it leaves the Owner's old numbers on the screen. This exposes business information to a role that should not see it.

MAJOR

O02

A 9:00 a.m. Class can move by five hours depending on where the server runs

The app does not attach the club's timezone when Staff enter a Class or Appointment time. In our test, the same 9:00 a.m. entry was saved five hours apart when the server used UTC instead of Chicago time. Daily and monthly Reports use the server's timezone too. Classes can appear at the wrong time or count toward the wrong day.

MAJOR

O03

The schedule accepts bookings that cannot all happen

Elite Fitness accepted overlapping work for the same Trainer, the same Member, and the same room. It also accepted an inactive Trainer and could accept more Members than a Class holds when final bookings arrived together. In another test, two identical Appointments reached the same Trainer at the same time. Staff must catch these conflicts by hand or Classes and paid coaching can be promised to people when the Trainer, room, Member, or place is already taken.

MINOR 

O04

Screens can show old, blank, or misleading results

Reports and Billing can briefly keep old totals after a successful action. When a real request fails, Dashboard can keep showing loading blocks while Reports looks blank. We also forced one history record to fail after a Lead had already been saved, and the app said the whole action failed. Staff may retry work that already succeeded or trust a screen that never explains what happened.

Ways the app can be misused

These require someone to step outside the normal flow on purpose.

ABUSE EFFORT

Easy	A beginner could do it with ordinary tools and simple instructions.
Medium	The person would need to deliberately change requests or have solid technical knowledge.
Hard	The person would need unusual access, rare conditions, or advanced technical skill.

MINOR ABUSE: EASY

M01

Someone can reserve a future Staff email before the Owner uses it

Public signup creates a login without making a usable Staff account. An outsider can use that to claim a future Staff email, after which the Owner cannot add that person. A similar collision happens if Staff setup fails after creating the login but before saving the Staff profile. The email then needs technical repair or replacement before the Owner can finish setup.

MINOR ABUSE: MEDIUM

M02

A signed-in user can put false information into Class history

Someone who works around the normal screens can record attendance for a cancelled past Class that the Member never booked. They can also book an inactive Member into a Class or Personal Training. These actions do not grant broader access, but they let a signed-in person make Class history and the operating schedule disagree with the Member's real status.

The full checklist in Appendix A shows all 82 checks, including what passed and what still needs a live system to confirm.

Still to confirm

These are checks we could not finish. They are not failed checks.

Live website and domain

This sample did not include a live website or production domain. We therefore could not check whether the live app opens correctly, whether the domain redirects properly, whether its security certificate is valid, or whether the first live page load reports errors.

Production safety and recovery

This sample did not include a production database, backup and restore process, error-monitoring account, or release setup. Those parts still need to be checked before a real launch.

04 THINGS TO CONSIDER

These are not launch-blocking findings. They are choices or future limits worth settling as Elite Fitness moves from a sample into a real club.

Four lists will need a growth plan

Members, Leads, Appointments, and Billing load their full history at once. The largest test list, with 1,210 Leads, was still responsive on our computer, so we did not call this a current failure. These screens will eventually need limits or pages as the club's history grows.

The browser tab still carries a starter icon

Elite Fitness has its own page title and description, but the tab still shows the Vite starter logo and there is no complete app-icon set. This is launch polish rather than a problem with how the club operates.

Production login protection depends on the live setup

The app can start without a private login secret, and its sign-in limit trusts a network address supplied by the host. A real host may replace that address correctly, but no deployment was supplied to confirm it. Before launch, give the app its own secret and check the sign-in limit through the real website.

Decide how Email history should link Members and addresses

Staff can choose one Member while typing a different destination address. That may be intentional for parents, guardians, or other contacts. The review did not find a written rule that forbids it, so the club should decide the intended behavior before connecting a real email service.

05 WHAT WE VERIFIED WORKS

Several important parts of Elite Fitness already work:

- ✓ The release build, code-quality checks, database setup, and all seven automated tests passed
- ✓ Correct passwords work, wrong passwords are rejected, people stay signed in as intended, and signing out blocks the old session
- ✓ Adding a Member saves the Member, Membership, and first Invoice together
- ✓ The database protects the main links between records, stores money without rounding errors, and handles submitted values safely
- ✓ The main desktop pages and all files needed by the released app loaded correctly

So the app is not starting from scratch. Its basic build, sign-in, and data storage are working. The launch risk comes from rules that must stay consistent across accounts, payments, access, schedules, and failures.

06 OUR ESTIMATE

This sample report does not include a repair quote. We were not given an agreed repair scope, price, discount, delivery timeline, guarantee, or work involving a live service, so none has been added.

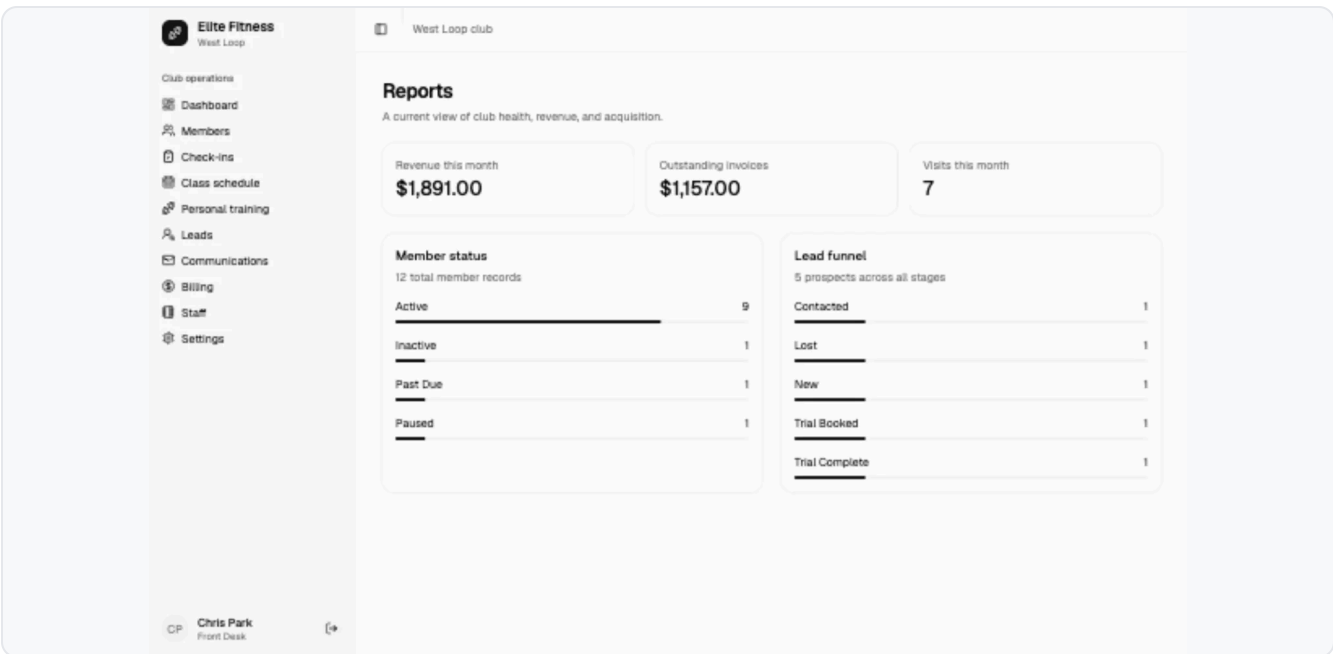
REPAIR ESTIMATE

Not included

No commercial terms were supplied for this sample report.

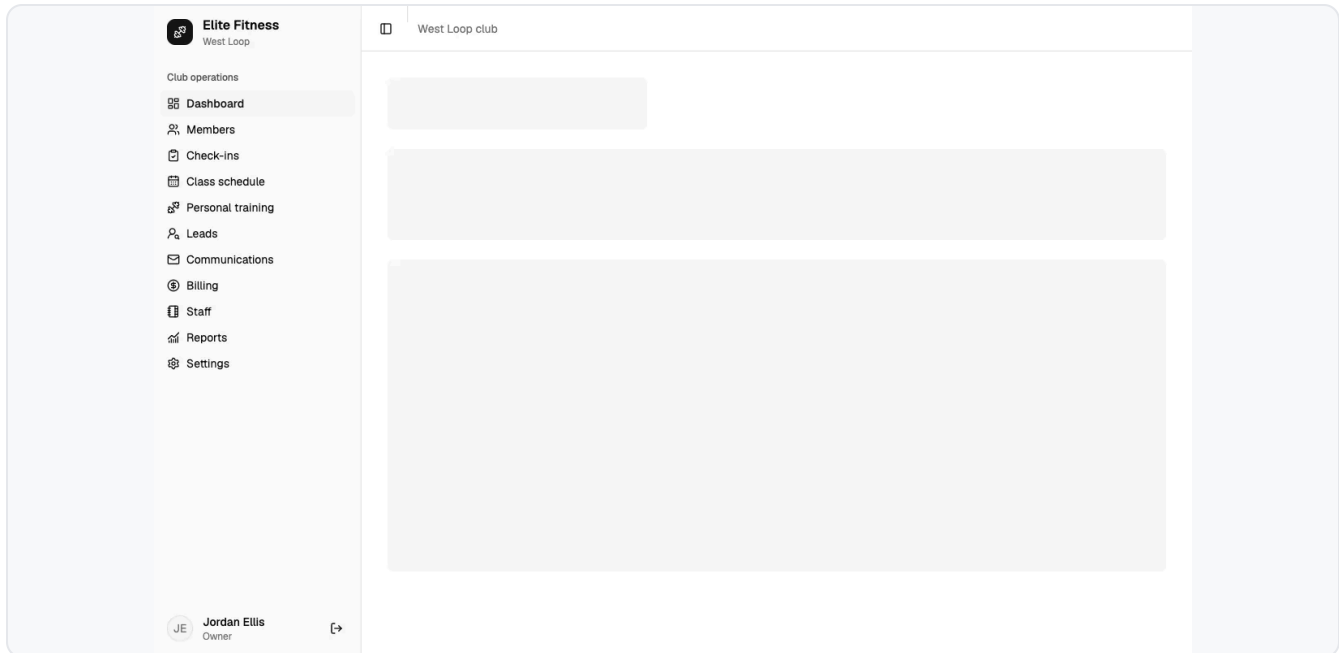
07 SELECTED PROOF

001 A Front Desk account can still see an Owner's report



The bottom of the screen says Chris Park is signed in as Front Desk. The page still shows the revenue, Invoices, visits, Member status, and Lead results that the previous Owner opened.

004 The Dashboard never explains that loading failed



The Dashboard request had already failed and stopped retrying. The screen kept showing loading blocks instead of telling the staff member what happened or offering a way to try again.

E04 Add Member cannot be completed on a small phone

A screenshot of the 'Add Member' form on a small screen, likely a smartphone. The form is centered and contains several input fields: First name, Last name, Email, Phone, Date of birth (with a date picker icon), Membership plan (a dropdown menu), Emergency contact, Emergency phone, and Health notes. The form is presented in a light gray box against a darker background. The layout is simple and functional, but the form's width is fixed, causing it to extend beyond the screen's boundaries on smaller devices.

On an iPhone SE-sized screen, the form extends past both ends of the screen. Scrolling does not bring the close or submit button into reach.

APPENDIX A: FULL MECHANICS CHECKLIST

This checklist shows the rest of what we reviewed. Pass means the check worked. Fail means we confirmed a problem. Warning means we found a limitation worth knowing, but not a separate reportable problem. N/A means the app does not offer or claim that feature. Not verified means the check required a live production system that was not supplied.

1. Auth and accounts

CHECKPOINT	RESULT	NOTE
Signup creates a usable account	FAIL	Accounts created by an Owner work. Public signup does not. It creates a person who cannot use the gym, and it can also stop an Owner from adding that email later. See M01.
Correct login works and wrong login fails	PASS	A correct email and password opened the app. A wrong password was rejected.
Session persists as intended	PASS	After signing in, the person stayed signed in for later requests. The session is set to last 12 hours.
Logout clears the session and protects routes afterward	PASS	Signing out ended the session. Trying to use the old session afterward did not open protected information.
Password reset works end to end	N/A	This version does not offer or claim a password-reset feature.
Email verification is enforced where claimed	N/A	This version does not offer or require email verification.
OAuth providers complete and link correctly	N/A	This version does not offer Google, Facebook, or another outside login service.
Logged-out visitors cannot open protected routes	PASS	People who are signed out are sent to the login screen. Gym information is available only after a Staff Member signs in.
Role and permission gating works	FAIL	The server blocks restricted actions. However, a Front Desk worker can still see a report left behind by an Owner, and some workers are shown a Class form they are not allowed to submit. See O01 and E04.
Signup avoids duplicates and email-collision holes	FAIL	The app blocks a second account with the same email. But public signup or an interrupted Staff setup can reserve an email without creating a usable Staff account. See M01.

2. Payments

CHECKPOINT	RESULT	NOTE
Checkout with a real test card	N/A	Payments are intentionally kept inside this sample. The app never contacts a real card company, so a card checkout does not apply.
Webhook receipt and signature verification	N/A	The sample payment system does not receive messages from a payment company, so this check does not apply.
Live production keys and test development keys	N/A	The sample payment system does not use payment-company keys.
Entitlement state matches the provider after purchase	WARNING	There is no payment company to compare against. Even so, the app can record a successful sample Payment while still blocking the Member because two status records disagree. See E02.
Cancellation, upgrade, and downgrade remain synchronized	N/A	This version does not offer payment-company cancellations, upgrades, or downgrades.
Failed or declined payment does not grant access	N/A	The sample payment system only represents a successful Payment. It does not offer failed or declined payments.
Prices cannot be changed by the client	PASS	The app gets the amount from the Invoice. It does not trust an amount supplied by the person's browser.
Unpaid users cannot access paid features	N/A	This sample intentionally allows Members with a pending Membership to enter. We did not treat that chosen rule as a payment bypass.
Refunds and chargebacks remove paid access	N/A	This version does not offer refunds or chargeback handling.

3. Security and row-level access

CHECKPOINT	RESULT	NOTE
Row-level security on user-data tables	N/A	The browser cannot connect directly to the database. All database access goes through the app's server, so browser-level row rules do not apply here.
Policies stop users reading or changing another user's rows	N/A	Elite Fitness is one shared club workspace. Signed-in Staff are meant to share the club's Member and day-to-day records.
Authorization is enforced on the server	WARNING	The server checks whether a person may perform protected actions. However, an Owner's report can remain on screen when a Front Desk worker signs in in the same tab. See O01.
Changing an ID does not expose another user's rows	N/A	Staff are meant to work with the same club records. The separate problem caused by submitting a particular Class or Member ID is covered by M02.
API routes authenticate before work	PASS	Every route that reads or changes gym information first requires an active Staff session.
Server validates and sanitizes user input	WARNING	The server checks the shape and format of submitted information. Some combinations that pass those checks are still unsafe, such as paying a draft Invoice or scheduling an inactive Member. See E02 and M02.
No obvious SQL/command injection or unsafe evaluation	PASS	We did not find a way for submitted values to become a database command, operating-system command, or code that the app runs.
Auth and abusable endpoints have appropriate rate limits	WARNING	The local server trusts a caller-supplied network address. A live host may overwrite it, but no deployment was supplied to confirm that boundary. See Things to consider.
File uploads restrict type, size, and enumeration	N/A	The app does not offer file uploads.
CORS is not dangerously open	PASS	A browser can connect only from the two local app addresses the server allows.

4. Secrets and environment variables

CHECKPOINT	RESULT	NOTE
Private secrets and API keys stay out of the client bundle	WARNING	The login page includes the working demo Owner credentials created by sample setup. They are labeled as demo data, but a real club needs a different first-Owner process. See E01.
Keys, tokens, and <code>.env` files are absent from Git history</code>	NOT VERIFIED	The supplied copy did not include the project's earlier change history, so we could not check old versions. The current environment file is set to stay out of Git.
Required production environment variables are enforced	FAIL	The released app can start without its own private login secret. When that setting is missing, it uses the same public fallback value every time. See Things to consider.
Service-role and admin keys stay server-side	N/A	This app does not use a service-role or admin key.
Public and server-only variables are separated	PASS	Database, login, and port settings remain on the server. They are not placed in the browser files.
Third-party keys are appropriately restricted	N/A	The intentionally local payment and email systems do not use keys from outside companies.

5. Deployment and infrastructure

CHECKPOINT	RESULT	NOTE
The app is deployed and reachable at its live URL	NOT VERIFIED	No live website or production system was supplied, so we could not check this. The release version did start correctly on our computer.
The production build succeeds	PASS	The release build completed. It created the pages, scripts, styles, fonts, and icon file the app needs.
Production does not depend on localhost-only configuration	PASS	The released app serves both its screens and data from the same place. It can also be given a real public address and a list of allowed website addresses.
Error monitoring exists and is capturing failures	WARNING	The app writes basic request and error logs. No monitoring service or live error history was supplied, so we could not confirm that someone would be alerted to production failures.
The first page load has no unexplained console errors or failed network calls	NOT VERIFIED	There was no live website where we could check the browser's error and request logs.
Production has no obviously broken or missing assets	WARNING	Every file the released app asks for was built and loaded. The tab icon is still the Vite starter logo. See Things to consider.
Required background jobs, cron jobs, and queues run in production	N/A	This version has no scheduled or background work to run. Its sample payment and email actions finish during the request.

6. Domain, email, and basic site identity

CHECKPOINT	RESULT	NOTE
The custom domain serves the app instead of a placeholder or parked page	NOT VERIFIED	No production domain or live website was supplied.
HTTPS is valid and the certificate has not expired	NOT VERIFIED	No live website certificate was supplied to inspect.
The `www` and apex domains resolve, or one redirects cleanly to the other	NOT VERIFIED	No production domain was supplied.
HTTPS pages do not load insecure HTTP assets	NOT VERIFIED	The local app loads its own files and data from the same place. Without a live secure website, we could not check whether any live page tries to load an insecure file.
Apps that send email have SPF, DKIM, and DMARC configured	N/A	Email is intentionally kept inside this sample. There is no real sending domain to check.
The favicon, app icons, and web manifest are intentional and load correctly	WARNING	The browser tab still shows the Vite starter logo. Elite Fitness also has no complete app-icon set or file that describes an installable web app. See Things to consider.
Public pages have a product-specific title and meta description	PASS	The page title and search description both identify Elite Fitness as a gym-management app.
Public pages can be indexed where intended, while staging and private pages are not exposed to search	WARNING	Elite Fitness is a private Staff app, but it does not tell search engines to keep production or test copies out of search results.
`robots.txt`, the sitemap, and canonical URLs exist where needed and point to the production domain	WARNING	The app does not include search-engine instructions or a preferred production address. A sitemap and preferred page links may not be needed for a private Staff app.
Shared links have intentional Open Graph titles, descriptions, images, and canonical URLs where relevant	N/A	The app does not offer public pages or content meant to be shared.

7. Core user flows

CHECKPOINT	RESULT	NOTE
The app's primary happy path completes end to end	WARNING	Automated checks covered login and Check-in, and we confirmed that the main day-to-day actions saved data. We did not run every journey through a live production website because none was supplied.
Important create, read, update, and delete actions work	PASS	The normal paths for Members, Leads, Check-ins, Classes, Reservations, Appointments, Billing, Email, and Staff worked. The known exceptions are explained in the findings.
Data persists across reloads and sessions instead of living only in memory	PASS	The local app saves business records and login sessions in its database, so refreshing the browser does not erase them. We could not check the durability or backups of a production database.
The main flows have no dead buttons or links	FAIL	A person who uses public signup can land in a screen they cannot use or leave. Front Desk workers and Trainers can also fill out a Class form that their role can never submit. See M01 and E04.
Multi-step flows keep their state without silently losing data	N/A	The day-to-day journeys in this version use one-page forms. They do not use multi-page setup flows with unfinished information to preserve.
The app works for a brand-new user with an empty account, not only a seeded account	FAIL	A fresh installation has no safe way to create its first Owner. See E01.

8. Error handling and edge cases

CHECKPOINT	RESULT	NOTE
Failures do not show a raw stack trace or blank screen	FAIL	The app does not show private server details. But when loading fails, a person may see a blank page, loading blocks that never finish, or a list that looks empty even though it failed. See O04.
Async actions show a loading state instead of looking frozen	PASS	Forms and Payment actions normally disable the submit button while they work. They then show whether the action succeeded or failed.
New users and other empty states look intentional	WARNING	Several screens show only blank space or an empty table when there are no records. They do not explain that the empty result is normal.
Forms give clear feedback when input is invalid	PASS	Forms check required information and common formats. If something is wrong, the app shows a readable message on the page or in a notification.
Double-clicks and rapid submissions do not duplicate actions or charges	FAIL	A button usually turns off after one click. But requests sent at the same time can still overbook a Class or create two conflicting Appointments. See O03.
The app has intentional 404 and 500 pages	WARNING	An unknown address sends the person back to the app's home screen. Data errors use a consistent response, but the app has no clear full-page message when a whole screen cannot load. See O04.
Network failures do not leave the app stuck on an infinite spinner	FAIL	If the Dashboard request fails, the loading blocks can remain forever. Other screens can turn blank or look as though there are simply no records. See O04.

9. Performance and cost

CHECKPOINT	RESULT	NOTE
Key pages have no obvious N+1 or runaway queries	PASS	Summary pages use a predictable number of database requests. List pages load related information together instead of making another request for every row.
Loops and repeated calls cannot hammer a paid API	N/A	Payment and email are intentionally local in this sample. There is no paid outside service for repeated calls to consume.
LLM and other paid API usage has a ceiling or guard	N/A	Elite Fitness does not use AI or another service that charges based on usage.
Images and other assets are not shipped at unreasonable sizes	PASS	The release build contains no large images. Its scripts, styles, and fonts are a reasonable size for this app.
The app has no accidental render loops or effects that fire on every render	PASS	We did not find a screen that accidentally reloads the same information over and over.
Large list endpoints use reasonable limits or pagination	WARNING	Members, Leads, Appointments, and Billing load their entire history at once. The tested sample remained responsive, so this is listed under Things to consider rather than as a current failure.

10. Data and database integrity

CHECKPOINT	RESULT	NOTE
The schema uses sensible data types, keys, and constraints	PASS	The database uses sensible rules for money, dates, IDs, positive amounts, Class size, and start and end times. The separate problems where related records disagree are covered by E02 and O03.
Foreign keys and relationships are enforced instead of merely implied	PASS	The database protects the main links among Staff, Members, Memberships, Classes, Reservations, Appointments, Invoices, and Payments.
A fresh database can be built from the migrations	PASS	The supplied setup command created a complete new database successfully.
Core relationships do not create orphaned or duplicate records	FAIL	An interrupted action or two actions at the same time can leave an unfinished Staff account, missing history, two current Memberships, an overbooked Class, or conflicting Appointments. See M01, O04, E02, and O03.
Unique constraints exist where the app assumes uniqueness	FAIL	The database prevents several kinds of duplicates. It does not prevent one Member from having more than one current Membership. See E02.
Backups exist, or the provider's backup and recovery setup is understood	NOT VERIFIED	No production host, backup schedule, restore instructions, or provider account was supplied.
Destructive operations have guards against mass deletion or unexpected cascades	N/A	The reviewed app does not offer a delete button or a bulk action that removes records.

11. Code quality and maintainability

CHECKPOINT	RESULT	NOTE
The app is organized clearly enough to make future production changes safely	PASS	The screens, shared interface parts, business rules, database definitions, login, payment, and email code are separated clearly. The main server file will become harder to change safely if it keeps growing.
Automated checks or tests run before deployment	PASS	The code-quality checks, type checks, seven automated tests, and release build all passed on our computer. No release setup was supplied, so we could not confirm that every future release must pass them.
Dependencies are in acceptable health	WARNING	One supporting package used by the development server has a moderate published security warning. The reviewed release server does not run that development server.
Basic documentation exists, such as a README	PASS	The project explains how to set it up, use the sample roles, run its commands, understand its structure, find its database, and work with the intentionally local payment and email systems.