Rails Application Audit Checklist

1. Understand the Business Context

☐ What is the app's core purpose?

☐ Which workflows are business-critical?

☐ Are there known pain points or outages?

☐ Who uses the app and how frequently?

2. Review Overall Architecture

☐ Are models, services, and views well-separated?

☐ Is domain logic centralized or scattered?

☐ Any use of Rails Engines or service boundaries?

☐ Custom patterns like Interactors, FormObjects, or use cases?

☐ Is there a clear application boundary (e.g., APIs, WebSocket layers, background workers)?

☐ Check if the app embraces Rails conventions, or fights against them.

3. Audit for Security Best Practices

☐ Mass-assignment protection (strong_parameters)

☐ Authorization: Pundit, CanCanCan, or something custom?

☐ Authentication: Devise, Authlogic, or manual handling?

☐ CSRF protection enabled?

☐ Proper escaping in views (html_safe abuse?)

☐ Secrets stored via encrypted credentials or leaked in git?

☐ Use of eval, YAML.load, or untrusted deserialization?

4. Check Test Coverage & Strategy

☐ What's the test stack? (RSpec, Minitest, Capybara, etc.)

☐ Is test coverage broad or superficial? (Use SimpleCov)

☐ Are factories clean and fast, or bloated and brittle?

☐ Any CI pipeline in place? (GitHub Actions, CircleCI, etc.)

☐ Are tests consistently green?

☐ Is CI integrated with GitHub PRs?

☐ Do test failures block merges?

5. Dependencies and Gemfile Hygiene

☐ Are core dependencies outdated? (bundle outdated)

☐ Are there unused or redundant gems?

☐ Forked gems or private gems—why?

☐ Any unmaintained dependencies?

☐ Overuse of gems for trivial logic?

☐ Licenses of 3rd party gems—any risks?

6. Database & Query Patterns

☐ Are N+1 queries present? Use bullet gem to detect.

☐ Are expensive queries being cached?

☐ Proper indexing in place?

☐ Do models have a sane amount of callbacks or complex scopes?

☐ Any misuse of default_scope?

☐ Query times via logs or APM tools (NewRelic, Scout, Skylight)

☐ Long-running migrations or locking issues

7. Background Jobs & Asynchronous Workflows

☐ Are tools like Sidekiq, Resque, DelayedJob in use?

☐ Jobs idempotent and retry-safe?

☐ Any monitoring for failed jobs?

☐ Long-running tasks handled asynchronously?

☐ Any scheduled jobs via whenever, sidekiq-scheduler, or cron?

8. Deployment, CI/CD, and Secrets Management

☐ How is deployment handled? (Capistrano, Heroku, Docker, GitHub Actions?)

☐ Is deployment automated or manual?

☐ Are environment variables managed via .env, Rails credentials, or a secret manager?

☐ Is rollback strategy documented and tested?

☐ Are logs centralized?

☐ Are there structured logs or tagging?

☐ Use of APM, error tracking (Sentry, Rollbar)?

9. Developer Experience (DX)

☐ Does bin/setup work?

☐ Are dev instructions in the README up-to-date?

☐ Is onboarding time short?

☐ Are rake tasks or custom scripts available for common operations?

☐ Can a new dev run rails s, rails c, and get started in minutes?