Small Business Websites and Automation Business Complete Course

Work, Money & Place | Local review edition v2 | November 21, 2026

Begin with the five-step route and keep fictional examples separate from your own evidence.

Build a small website that can receive a real inquiry

Learn three clear pages and one tested owner-review workflow, starting with fictional data on your own computer. Public hosting and email require separate client-owned access and observed delivery evidence.

Begin learning with L01. Check conditions and demand in L02 before buying or accepting paid work.

Five steps to a readiness decision

1. Name one bounded service — L01 / F01. Three pages,one inquiry task and explicit exclusions.

2. Check conditions and demand — L02, L03 / F02, F03. Permission-based demand evidence and a data/ownership map.

3. Rehearse the actual workflow — L04, L05, L06, L07, L08, L09 / F04, F05, F06, F07, F08. Local build and independently observed inquiry/failure evidence.

4. Estimate before selecting a price — L10, L11, L12 / F09, F10, F11. Measured baseline,complete estimate and costed change/correction.

5. Demonstrate readiness and handoff — L13, L14, L15, L16 / F12, F13, F14. Owner operation and separate live evidence where actually authorized.

Stop at missing conditions or evidence. The route is a study order, not permission to sell or perform unverified work.

Three-page informational website and one inquiry-to-owner-review workflow: inspectable local fictional-data lab plus a substantive conditional client-owned static-host/managed-form/email production path.

Study the written lessons in order, then use the blank forms and your own-business planner inputs. Exercises include reasoned worked answers. A paper course is not a credential or practical qualification.

800 minutes (13 hours 20 minutes) of lesson allowances. Separate permitted practice, research, provider review and real business setup take additional time. Fictional full client project: 2,100 owner minutes (35 hours), distinct from learner study time.

How to begin

Start with the local fictional-data lab and its README. Draw browser→receiver→inbox before changing a page; a success screen alone is not delivery evidence.

Study the full service cycle, including demand, content, costing, approval, hosted adaptation, recovery and owner handoff. Actual business readiness,client authority,accounts,publication and mailbox tests are not established by the fictional case.

This complete course is local and unpublished. No purchase,messaging,account creation,public deployment or course-sales decision is authorized by the teaching examples.

First action: Copy the permitted local specimen,draw the three destinations,and run one fictional request that an independent owner role can find.

Estimate before actual accounting

Use the independent Estimate sheet before setting a price or receiving money. Every required driver must be known, including confirmed numeric zero. Its conservative arithmetic floor includes a fee-rounding allowance; it is not a market price or readiness proof. Keep the accepted plan in a dated Baseline, then record Quantity, Stock, Your order and one-order Cash actuals separately. Your order C39 links the independent estimate and never requires actual price, receipts or refunds. Marketing ratios require all three matching funnel counts. Earlier-order refunds need their own dated order record; never charge them to this order.

Stock scans all eight own-business slots, including opening units and rows after a gap. Confirm the number of filled rows in Stock C23. A mismatch or partial row withholds totals J34 and K34; zero is valid only when no rows are filled. Complete every required zero in a used row.

Your order C19 is only the actual fee withheld from service receipts. It cannot exceed those receipts. Record a genuine additional processor bank debit paid in C47 with its reason, date and statement reference in C48; confirm 0 when none. It is included once in management cost and linked Cash49. Unknown or unreconciled fee data blocks a successful result.

L01 Choose a website result you can prove

Define demand and a controlled scope. Study and exercise allowance 35 minutes.

Produce: A three-page prelaunch offer with a named inquiry handoff

Use forms F01.

A small result with an honest status

Define two offers before quoting. The local prototype is three prelaunch pages and one local inquiry-to-owner-review demonstration using fictional data; it does not include public hosting or real email. The complete fictional project CP-026 for $1,800 is a different, explicitly expanded scope: the same three pages plus conditional client-owned hosting, a managed receiving form and one owner email notification, including acceptance and follow-up. Those launch tasks are included in CP-026’s 2,100-minute estimate, rather than an extra charge after that full project is agreed. Domains, hosting and mailbox subscriptions remain client-owned and directly client-paid under its terms. Payments, booking, customer accounts and marketing campaigns remain outside both offers. Identify which offer a customer is considering before discussing a price.

The fictional customer is Juniper Window Care, an owner-operated residential window cleaning business. Fictional owner Morgan Lee wants a clear service area, an explanation of ordinary exterior cleaning, and an inquiry path. We use only invented names and example.invalid email addresses. The finished teaching specimen runs on your own computer, listens only at 127.0.0.1, and saves accepted inquiries in a local owner-review file. It does not send email. The visible outcome says saved for owner review, never email delivered or appointment confirmed. That small promise can be tested completely.

Separate the page from the working system

The browser displays the pages and checks simple input. The server receives the submitted fields, checks them again, and writes an accepted record. The inbox file is the owner's review destination. Drawing these three places stops a common mistake: assuming a green message proves that a request reached a person. The course's original system is a teaching design built with standard HTML and Python library components. Python explicitly warns that http.server is not recommended for production [S01]. Do not place this local server on a public network, forward a router port to it, or put real customer information into it.

A commercial version of this narrow service can still be useful. The customer buys an organized prelaunch asset and evidence about its behavior, with an explicit decision about later deployment. If a prospect expects a publicly available website next week, this prototype scope does not meet that need by itself. Either add a separately estimated production stage after you can demonstrate its controls or decline that part. Do not quietly sell a local demonstration as a live system. A learner who wants to sell launch services must complete the additional production evidence in L13.

Make the boundary observable

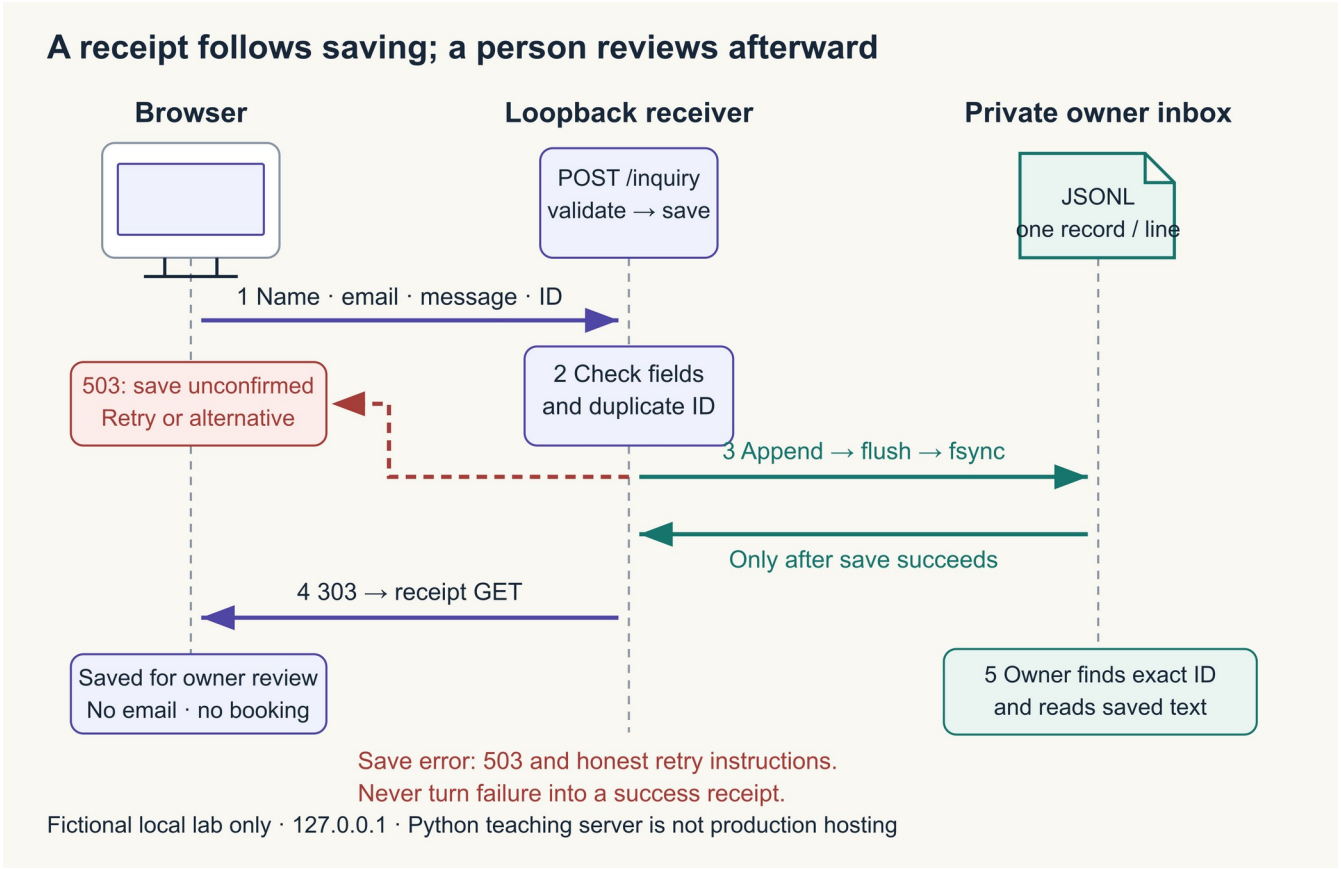
Write one included content revision, the exact pages, the fields, the response wording, and who supplies content. Exclude online card collection, health or financial records, file uploads, customer accounts, large applications, advertising trackers, automated sales replies, and ranking promises. A request for any of these changes the risk and scope. The source-backed technical rehearsal gives you a place to practice; it does not establish real client permission, legal compliance, accessibility conformance, or professional competence.

For F01, name a real type of customer you could reach through permission-based conversations. Then describe a visitor's journey in three verbs: understand, inquire, review. Estimate what the owner must do after a request arrives. If the owner never checks the destination, the automation is unfinished operationally even when the software works. Your first action is to draw the three places on paper and mark where success becomes true. Keep that drawing beside your computer throughout the build.

1. Record three pages, one workflow, one revision, and the explicit prelaunch status.

2. List excluded data and features before selecting tools.

3. Identify the human who will review requests and a separate production evidence gap.



Original local system diagram. HTML/Python components supported by S01–S09 and303 by S16. Local save is not email delivery or public readiness.

Practice

A prospect asks for a contact page that accepts card details and automatically books a cleaning appointment. Does this fit?

Worked answer

No. The agreed starter workflow collects only a name, reply email, and short non-sensitive message for owner review. Card collection and appointment confirmation introduce separate systems and responsibilities. Offer the bounded prototype only if it meets the prospect's need; otherwise refer or prepare a separate supported scope.

Sources: S01. Full links and limits are in the source register.

L02 Find demand before buying subscriptions

Define demand and a controlled scope. Study and exercise allowance 45 minutes.

Produce: A permission-based two-week acquisition experiment with matching costs

Use forms F02.

Use a service specific profile

Use this complete fictional profile as a pattern: "Clear Path Web Studio prepares three-page prelaunch websites for small local service businesses. We organize owner-approved content and demonstrate one

inquiry-to-owner-review workflow using fictional test data. You receive editable files, a test record, and a handoff guide. Public hosting and real email delivery require a separate verified launch stage. We do not promise search rankings, sales, compliance certification, or online payments. Ask for a short fit conversation before requesting an estimate." This profile describes the actual result and its limit without pretending that a prototype is already online.

Appropriate first channels are an existing business-owner network, a permitted introduction through a local business group, and a small demonstration at an event whose organizer agrees. Look for service businesses with a specific information problem: unclear service area, outdated contact instructions, or an owner who cannot explain where inquiries go. Do not scrape a list, invent an audit finding, or approach customers while pretending to represent a directory. SBA's planning resources support researching customer needs and startup costs [S12]; they do not establish demand or prices for your town.

The profile above markets the limited prototype. It is not the scope or price of the full CP-026 case. If a prospect wants the hosted form and one owner notification, describe the expanded offer and estimate the complete work, access and acceptance conditions. The $1,800 case already includes that hosted stage; do not charge again for delivering an included function.

Ask permission without sending anything now

Unsent permission request for a conversation: "Hi Morgan, I'm Casey at Clear Path Web Studio. You mentioned that customers often ask whether you serve their neighborhood. May I show you a short, fictional three-page website demonstration and ask how you currently receive inquiries? It takes about ten minutes. There is no obligation. If this is not useful, I will leave it there." The owner must actually have mentioned that problem before you use that wording; otherwise say honestly that you are researching the need.

Unsent follow-up after permission: "Hi Morgan, thank you for agreeing to look at the demonstration. The package is a prelaunch prototype, not live hosting or guaranteed email delivery. Would Tuesday at 2 or Thursday at 10 suit a ten-minute review? If you would rather stop, reply stop and I will close this follow-up." These are conversation examples, not send-ready legal email templates. If you use commercial email, include accurate sender and subject information, required advertising identification, a real valid postal address, and a functioning opt-out process; the FTC says CAN-SPAM also applies to business-to-business email [S13]. Obtain channel permission and follow its rules. No messages in this course have been sent.

Measure one population over two weeks

In the fictional paid-channel experiment, days 1–2 spend 60 minutes preparing the narrow profile and permitted demonstration. Days 3–5 spend 60 minutes discussing six permitted prospects. Days 6–7 spend 60 minutes making requested demonstrations. In week two, spend 30 minutes on allowed follow-ups and 30 minutes logging outcomes and improving the offer: 240 owner minutes altogether. Spend $60 on one organizer-approved placement. It produces six inquiries, two suitable estimate requests and one full-project booking during fourteen days after the additional live scope and readiness conditions are defined. The optional prototype remains a separate, smaller offer. Verify the placement's actual channel and terms before a real purchase; $60 is an invented planning input.

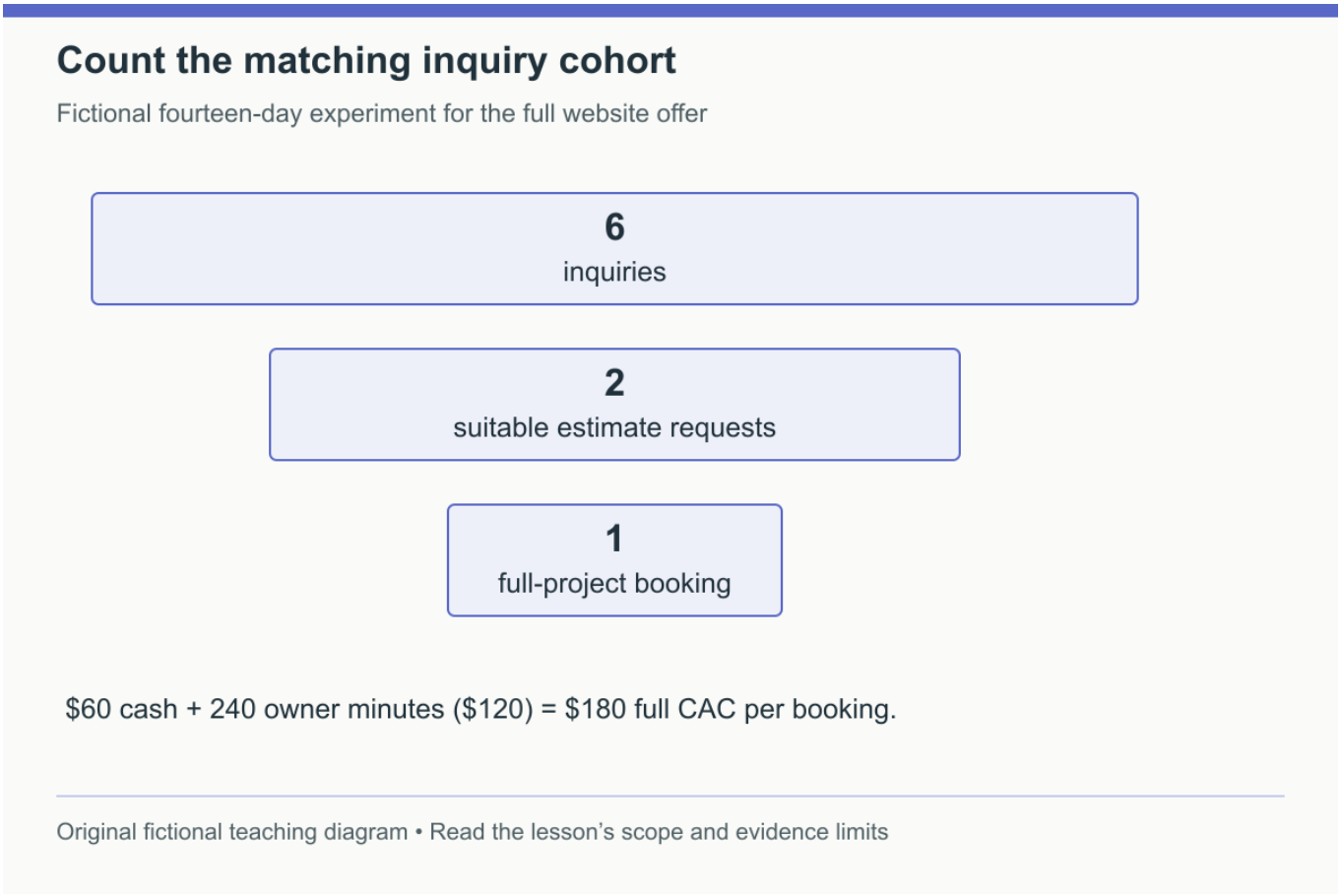
Cash CAC is $60 divided by one booked customer, or $60. At an invented owner value of $30 an hour, four hours cost $120; combined acquisition cost is $180, so full CAC is $180. Cash cost per inquiry is $10 and full cost per inquiry is $30. Estimate-request conversion is two of six, 33.33%; booking conversion is one of six, 16.67%. An unrelated referral is recorded separately and never added to this denominator. With zero bookings, CAC is undefined, not zero. After two weeks, buy nothing further

unless suitable requests, your delivery capacity, and expected contribution after acquisition support continuing. A no-sale outcome can be useful evidence to change or stop the offer.

1. Record consent, channel terms, dates, six inquiries, two estimates and one booking in F02.

2. Compare the $180 full acquisition cost with the project's $548.70 management profit before reserve; $368.70 remains before the reserve.

3. Stop further spending if permission, demand or delivery capacity is missing.



Six inquiries lead to two suitable estimates and one full-project booking in the fictional campaign. $60 cash plus $120 owner value gives $180 full CAC.

Practice

The same two-week period includes one referral booking unrelated to the placement. Can you report $30 cash CAC for the placement?

Worked answer

No. Keep the referral separate. The paid experiment acquired one customer from its own six inquiries, so its cash CAC remains $60 and full CAC $180. Two total business bookings do not change the paid channel's matching population.

Sources: S12, S13. Full links and limits are in the source register.

L03 Discover the work and control access

Define demand and a controlled scope. Study and exercise allowance 45 minutes.

Produce: An approved discovery record and a minimal data/access map

Use forms F03.

Ask questions that change the build

Run discovery as a short decision interview. Ask which service the business wants inquiries about, whom it serves, which locations it can actually serve, how quickly it can normally review a request, and what information it needs to decide the next step. Ask the owner to show the present inquiry process using an invented example. Do not ask for a mailbox password or a real customer's message. A useful discovery answer is “Morgan reviews inquiries at 9 a.m. on business days and responds manually.” “Make it modern” does not define a working result.

For fictional Juniper, the site has Home, Services and Contact pages. Morgan supplies one authorized photograph, a logo or plain text business name, accurate service descriptions and a working public contact alternative for any eventual launch. The prototype uses a clearly labeled fictional phone number, 202-555-0104, as an illustration only. The form asks for name, email and a short message; it expressly asks visitors not to send passwords, card details or sensitive records. It makes no service booking or emergency-response promise. Review these facts with the decision maker, not merely with someone who likes the design.

Map where each field travels

On F03 draw browser → local server → inbox.jsonl → authorized owner review. Name, email and message are personal data even though this rehearsal uses fictional versions. Record purpose, location, access, retention and deletion for each copy. For the invented rehearsal, only the learner and fictional owner role inspect the local test folder, and test records are deleted after the evidence report is complete. That is a rehearsal choice, not a universal retention rule. FTC guidance calls for knowing what information you hold, keeping only needed information, limiting access and planning disposal and incidents [S10].

For an eventual production site, inventory server logs, backups, provider dashboards, notification mailboxes and any support tools as well as the form database. The client decides the business purpose, recipient and suitable retention policy with appropriate legal advice where needed. Your implementation must follow the approved arrangement. A promise such as “we never store data” would be false if the server keeps a record or the email provider stores a message. Avoid adding analytics, third-party embeds or marketing consent checkboxes without a defined purpose and supported implementation. A privacy notice must describe the actual system.

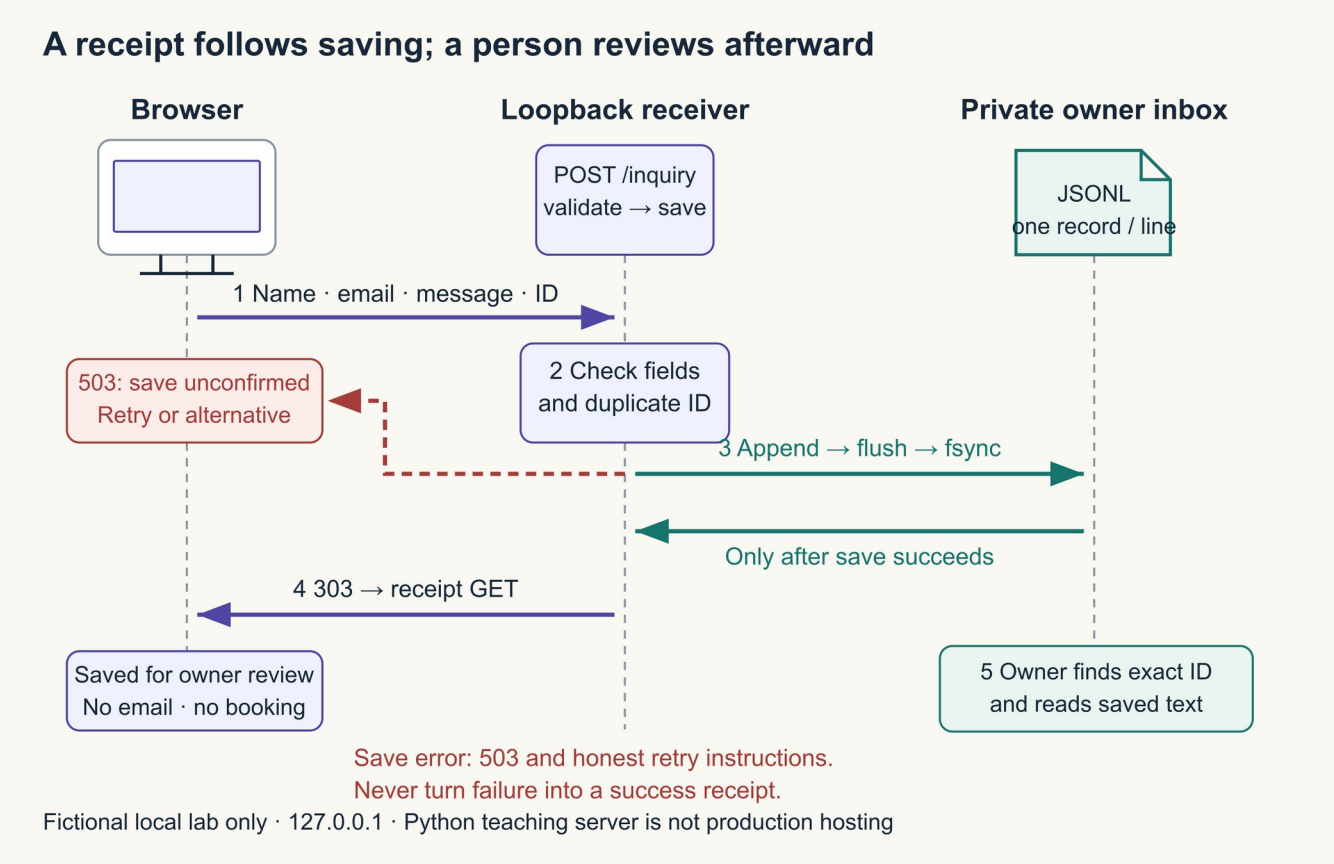
Give the client lasting control

Separate authority to approve the project from access to operate it. The client should own its domain, hosting subscription, business mailbox and recovery contact. If a production platform supports delegated access, use the narrow role needed for the work and remove it at handoff. Do not register a client's important account under your personal recovery email or make yourself the only administrator. Never request shared passwords in chat; use the platform's supported delegation and recovery process. No account creation or delegation occurs in this local course.

Complete F03 with the content approver, account owner, inbox reviewer, backup owner and incident contact. For the prototype those are role assignments, not proof that an external service exists. Record “not selected; launch blocked” for the production host and notification service. Also check your own registration, licensing, insurance and tax obligations for the actual service and location before paid work; SBA identifies these as launch decisions [S14]. Do not describe a generic small-business registration as

permission to provide every technical or regulated service. If the prospect asks for regulated records or functions beyond your capability, narrow the task before continuing.

1. Use invented discovery data until the real client authorizes a specific project.
2. Write who approves content, who pays renewals and who handles missed inquiries.
3. Record missing production ownership evidence explicitly.



Original local system diagram. HTML/Python components supported by S01–S09 and303 by S16. Local save is not email delivery or public readiness.

Practice

Morgan offers to send a shared mailbox password. What is the next useful action?

Worked answer

Do not take the password. Establish whether mailbox access is necessary, identify a supported delegated role and obtain authorization for that role if a later production project requires it. The local prototype needs no real mailbox access at all.

Sources: S10, S14. Full links and limits are in the source register.

L04 Turn approved content into a three page plan

Define demand and a controlled scope. Study and exercise allowance 40 minutes.

Produce: A page map with factual copy, rights evidence and one visitor path

Use forms F04.

Collect facts before arranging boxes

Build a content inventory with one row per fact or asset. For the fictional Home page, the core facts are the business name, residential exterior window cleaning, an owner-approved service area and a request-for-information link. Services explains what an ordinary inquiry concerns and states that access, condition and price are confirmed separately. Contact explains the three fields and what saved for owner review means. Every service claim needs owner approval. Do not invent licenses, insurance, guarantees, customer reviews, years of experience, opening hours or exact response times to make the page look complete.

The inventory includes the source, rights holder, approval date and destination page for each image and text item. A photograph found through a search is not automatically licensed for a client's website. A customer supplying a photo does not settle whether people depicted have appropriate permission. This course uses original illustrations and fictional content. In a real assignment, request usable rights and preserve the record; seek appropriate advice when ownership or releases are uncertain. Use a plain text business name until a logo's rights and readable presentation are resolved.

Sketch the actual visitor route

Draw three page rectangles and connect their navigation in both directions. Home answers what the business does, who it serves and what to do next. Services answers whether the ordinary task is suitable. Contact lets a visitor send an inquiry without reading a long sales page. Put the business name and three links in a consistent header; provide a main heading, readable paragraphs and one clear action. The footer repeats an approved contact alternative and the prelaunch status. Do not put the only contact link inside a large image or rely on a social-media embed.

The original page geometry figure shows a wide two-column home layout becoming one reading column on a narrow screen. The order remains heading, explanation, relevant image and action. Draw the contact form as a single vertical sequence so labels stay close to fields. Before styling, read the draft aloud as a first-time visitor. If you cannot identify the service and next step from the words, a stronger color palette will not solve the problem. A clear page title and useful descriptive headings also support discoverability; Google's guide gives no guarantee of indexing or ranking [S15].

Define completion and handle missing material

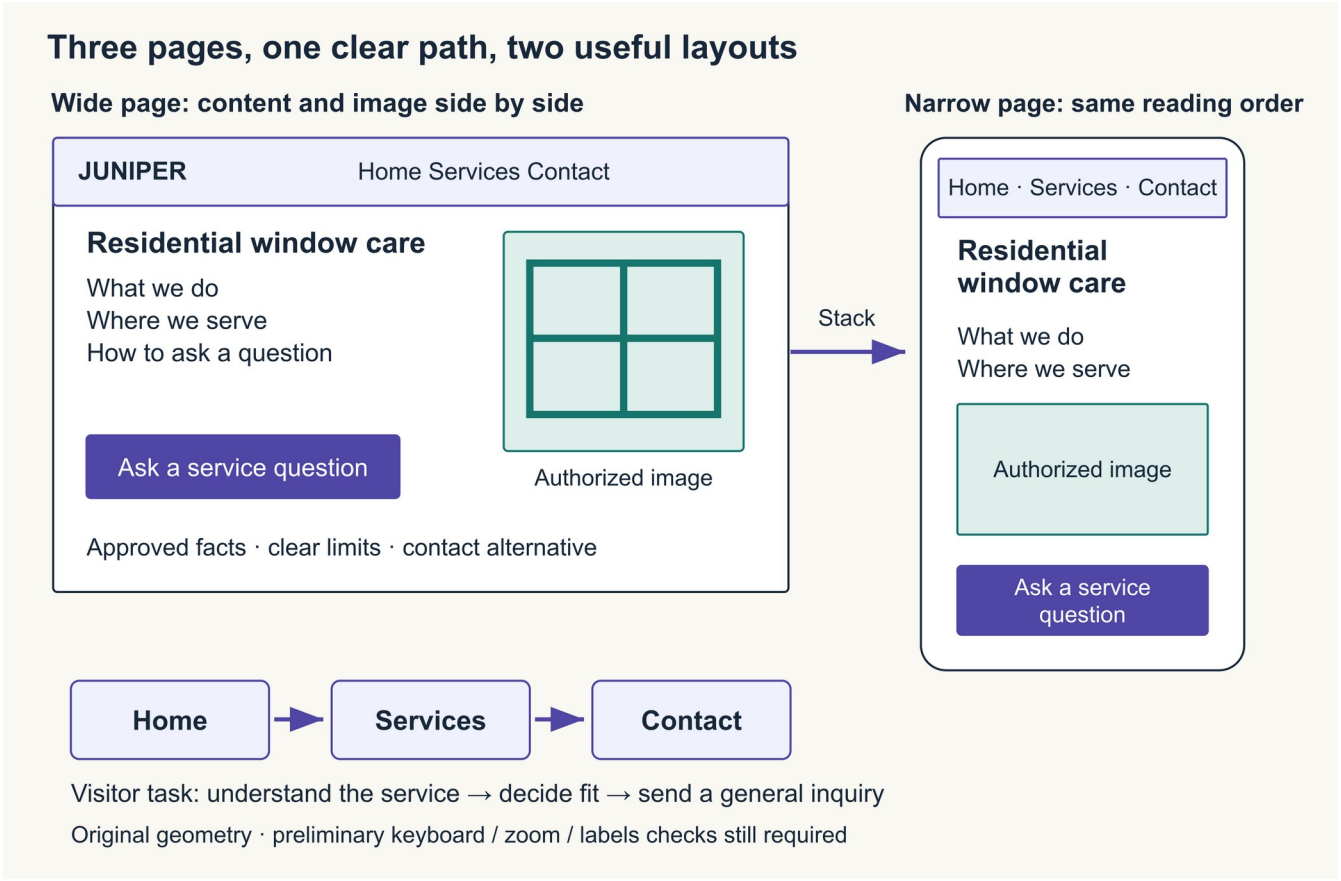
Write acceptance checks beside the map: all three navigation links work; each page has its own meaningful title; facts match the inventory; image alternatives fit the image's purpose; the inquiry action leads to Contact; and the pages work without a sign-in. Identify the one included content revision as one consolidated list received by the agreed deadline. A new page, new language, copywriting from scratch or a new workflow is a change, not an unlimited revision. The plan prevents the word “website” from expanding invisibly.

In the fictional case, Morgan supplies approved copy on November 24 and the prototype review is scheduled for December 1. If the copy is late, stop the affected build stage, record the dependency and offer a revised date. You can prepare the layout with obvious placeholders, but do not replace missing facts with plausible claims. Mark all demonstration pages as fictional and prelaunch. F04's finished inventory should let another person trace any visible business fact back to an approval. It should also show what is missing, who must supply it and which date depends on it.

1. Inventory each visible fact and asset with its approval and rights status.

2. Draw Home → Services → Contact and a return route.

3. Write five observable checks before building the first page.



Original page geometry for the fictional business. W3C preliminary review guidance S11; design choices do not establish accessibility conformance.

Practice

The business has no customer review to provide. Should you create a realistic placeholder testimonial?

Worked answer

No. Use no testimonial, or an unmistakable design placeholder that is removed before any client-facing approval. A plausible invented review misrepresents evidence and can survive accidentally into a public page. The fictional teaching case is labeled throughout.

Sources: S15, S11. Full links and limits are in the source register.

L05 Build the pages in an inspectable local project

Build and test the local receiving workflow. Study and exercise allowance 55 minutes.

Produce: A local three-page site whose content and links you can change and restore

Use forms F05.

Open the supplied lab and its short guide

The course menu and Excel and tools page contain a prominent Working inquiry lab link. Open that guide before starting. In a downloaded review archive, the same short instructions are in practice-lab/QUICK_START.txt; the full README sits beside them. Extract the complete archive first. The course viewer is read-only teaching material; the separate lab is the running three-page exercise.

Use Python 3.10 or newer already available on your computer. From the extracted practice-lab folder, run python3 START_PRACTICE_LAB.py and keep that terminal open. Open the exact printed http://127.0.0.1 address in an ordinary browser. Use the existing safeguards unchanged; if an embedded preview rejects the request’s origin, use the ordinary browser rather than weakening the check. No installation, real account, notification or public service is needed for this local exercise.

Understand the project before changing it

Work in a separate copy of the supplied local specimen. The project has a server file, a public folder for Home, Services and Contact, a stylesheet, a private data folder for inbox.jsonl, and a test report. Read the specimen README before starting it. Use its documented start command from the project folder and open the printed 127.0.0.1 address in your browser. Do not double-click the contact HTML file and expect its server submission to work. The address must point to the running local server. If the documented Python prerequisite is missing, stop and arrange an authorized installation rather than downloading unrelated software.

Find a plain heading in the Home page, change one sentence, save, and reload. That small action teaches the file-to-browser relationship. Next, deliberately break a copy of the Services navigation link, observe the failure, restore the original path and test it again. Record the file and corrected path in F05. Keep a clean original copy or a versioned backup before making wider changes. A rollback is useful only if you have demonstrated that it restores the page and behavior you intended.

Build semantic structure before decoration

Each page needs a document language of English, a meaningful page title, a header, navigation, main content and footer. Use a single clear main heading and sensible lower headings for real subsections. Make navigation real links and form submission a real button. Do not use clickable decorative boxes as substitutes for controls. HTML's form model separates the interface from server processing [S02]. This project keeps that boundary visible: page files explain and collect; the server decides whether a submitted request is accepted.

Use the original geometry figure as a layout specification rather than tracing a competitor's site. Keep text in ordinary HTML. Give images a maximum width that fits their container and avoid a fixed page width larger than the viewport. Use a readable line length, generous spacing and visible focus. On narrow screens, stack content in the same reading order. These are original design choices for the specimen, not a claim that a short stylesheet guarantees accessibility. Avoid adding animations, external fonts or plugins just to make the page appear more advanced; every added dependency creates another thing to test and maintain.

Verify routes and keep private material private

The local server should serve only the named page and asset routes. The data folder, test fixtures and server source must not become downloadable pages. In the specimen, an attempt to request the inbox through a browser should return not found. Do not place inbox.jsonl inside the publicly served assets folder. Server logs should avoid printing complete inquiry bodies or private credentials. The fictional contact values remain obvious demonstration data, so the exercise can be repeated without accumulating actual client records.

Make a small route inventory: Home, Services, Contact, receipt, stylesheet and any original image. For each page, follow navigation from a different page, reload directly, and use the back button. A receipt page should not contain the submitted name or email in its URL. Save a screenshot at a narrow width and one at a wider width. Document what you changed, what you checked and how to restore the clean copy. If a page looks right only because your browser cached an old file, the direct reload check will

expose it. Finish by stopping the local server using the README's method and confirming the address no longer serves the project.

1. Start only the local specimen and change one approved sentence.
2. Test every declared page route and a deliberately nonexistent route.
3. Restore a clean copy and repeat the navigation checks.

Working inquiry lab quick guide

Run the working inquiry lab

Use this separate local exercise to follow one fictional inquiry from an ordinary browser to the private owner inbox. The course viewer and lab use separate local servers.

Use an extracted writable copy of the review archive and Python 3.10 or newer already available. No extra packages or installation are performed by the specimen.

Use only invented names, example.invalid email addresses and a non-sensitive unique message. Keep the receiver's Host/Origin safeguards unchanged.

Find the complete folder

From the review index, obtain the Website course review ZIP, or use an already extracted combined package. Find practice-lab with START_PRACTICE_LAB.py, REVIEW_OWNER_INBOX.py, QUICK_START.txt and the full README. Do not open contact.html directly.

Start the receiver

Open a terminal in practice-lab and keep it open. The starter chooses an unused loopback port and prints the exact address. On Windows an existing py -3 command may be used instead of python3.

```
python3 START_PRACTICE_LAB.py
```

Submit once in an ordinary browser

Open the printed 127.0.0.1 address in an ordinary browser. Choose Contact. Enter Avery Practice, avery@example.invalid and a unique fictional message such as "Practice follow-up WMP-EXAMPLE: please explain ordinary window cleaning." Submit once and observe Saved for owner review. This confirms no email or booking. The form generates its own fresh identifier; DEMO-001 is reserved for the automated fixture.

Find the same private record

From a second terminal in the same folder, run the reviewer. Match the exact name, email and unique message, and record the generated ID and saved status. Compare the count before and after; a confirmation page alone is insufficient.

```
python3 REVIEW_OWNER_INBOX.py
```

Stop and record the limits

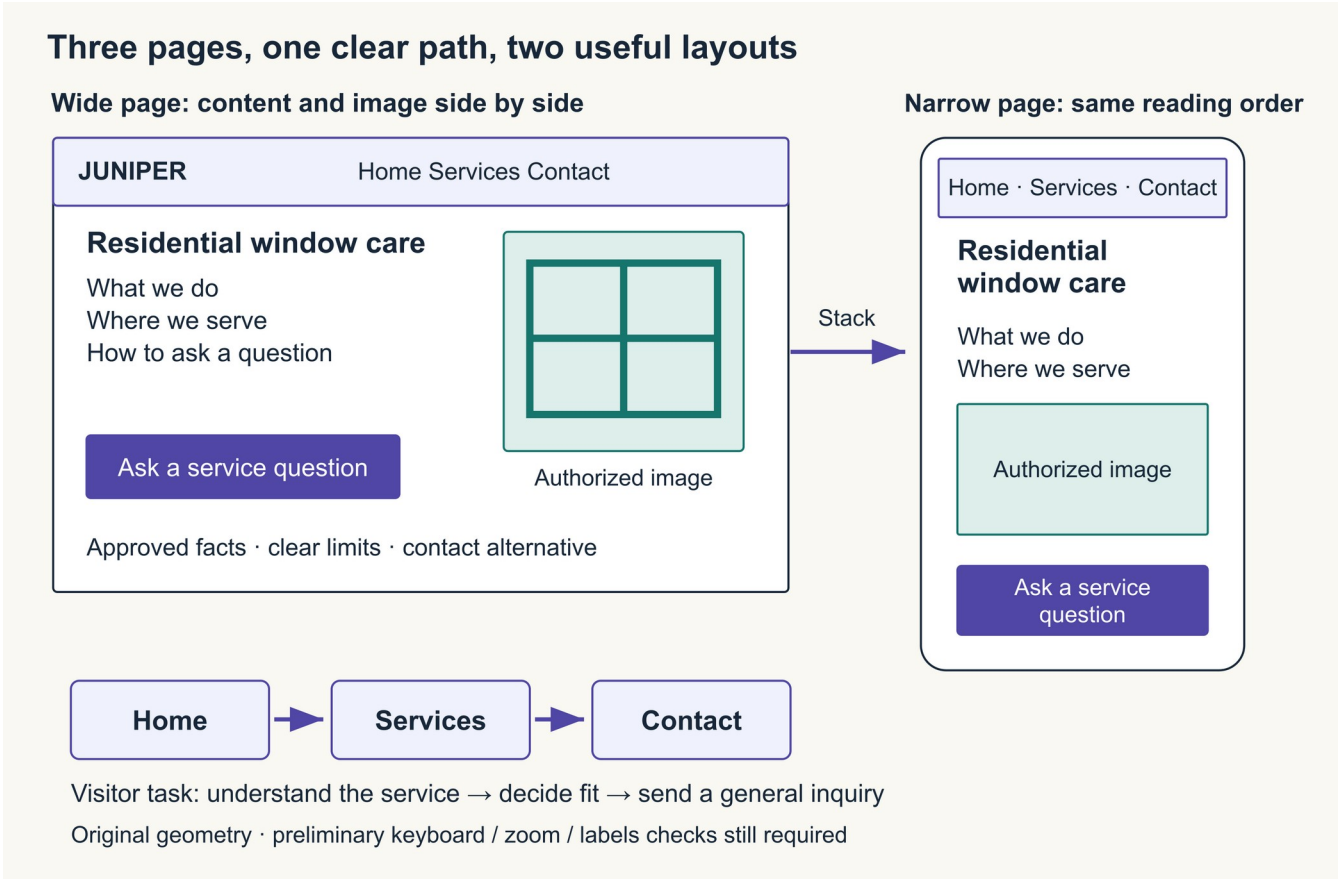
Return to the starter terminal and press Control+C. Record the observed URL, generated ID, message, confirmation and owner-review result in F07/F08. Keep fictional fixture evidence private. The local lab sends zero emails; it cannot establish the hosted notification or real client-owned access path.

One ordinary manual submission corresponds to one exact private record.

The displayed receipt and the independently found inbox record are different pieces of evidence.

A missing notification on the full hosted offer requires the agreed dashboard fallback and repair/retest; the local lab cannot prove email delivery.

If the embedded browser returns an origin rejection, use an ordinary browser at the exact printed address. Never disable or widen the receiver's protection. Actual native-browser UI observations and protocol-only tests are reported separately.



Original page geometry for the fictional business. W3C preliminary review guidance S11; design choices do not establish accessibility conformance.

Practice

The Home page looks right when opened as a file, but Contact fails to submit. What do you check first?

Worked answer

Check whether you opened the running specimen's local server address rather than a file URL. The browser can display a static page without a server, but the POST endpoint requires the documented local process. Then inspect the form action and server route; do not add an invented email setting.

Sources: S01, S02, S11. Full links and limits are in the source register.

L06 Create a form people can understand

Build and test the local receiving workflow. Study and exercise allowance 50 minutes.

Produce: A labeled three-field form with honest instructions and outcomes

Use forms F06.

Connect labels fields and instructions

Open Contact and identify the three visible fields: name, email and message. Each has a visible label associated with the matching control identifier. A label is not the same thing as placeholder text that disappears when someone types. Mark required fields in words. The email control uses the email input type; the message uses a multiline text area. The original specimen asks for one to eighty characters for

name, an email of no more than 254 characters, and a ten-to-one-thousand-character message. These limits are teaching design choices, not legal or technical maximums for every website.

Show this instruction before the form: “Use this form for a general service inquiry. Do not send passwords, card details or sensitive personal records. This fictional local demonstration saves accepted requests for owner review; it sends no email and confirms no appointment.” For an eventual real site, the wording must describe the real workflow and approved contact alternative. W3C's forms guidance explains explicit labels and clear instructions [S03]. Keep help beside the relevant field so a visitor does not have to remember a paragraph from the top of the page.

Specify what the browser sends

The form method is POST and its action is /inquiry. The browser sends ordinary application/x-www-form-urlencoded fields. The name attributes are name, email, message and request_id. Without a name, a visible control's value may not be submitted as expected. The server supplies a hidden request identifier when it serves Contact; the user does not type it. It helps the specimen recognize a repeated request, but it is not authentication, a privacy control or an anti-abuse service. No JavaScript is needed for the basic submission route [S02].

Required and length attributes provide helpful first checks in the browser. They do not establish that the server can trust the request. Someone can bypass the browser interface or send different data. W3C explicitly distinguishes client-side assistance from server-side validation [S04]. The next lesson repeats the agreed checks at the receiving end. Do not add a checkbox that says a user accepts every possible privacy term merely to make the page look official. Collect only a purposeful decision that you can record and honor; a generic checkbox does not repair an inaccurate privacy practice.

Give useful success and error messages

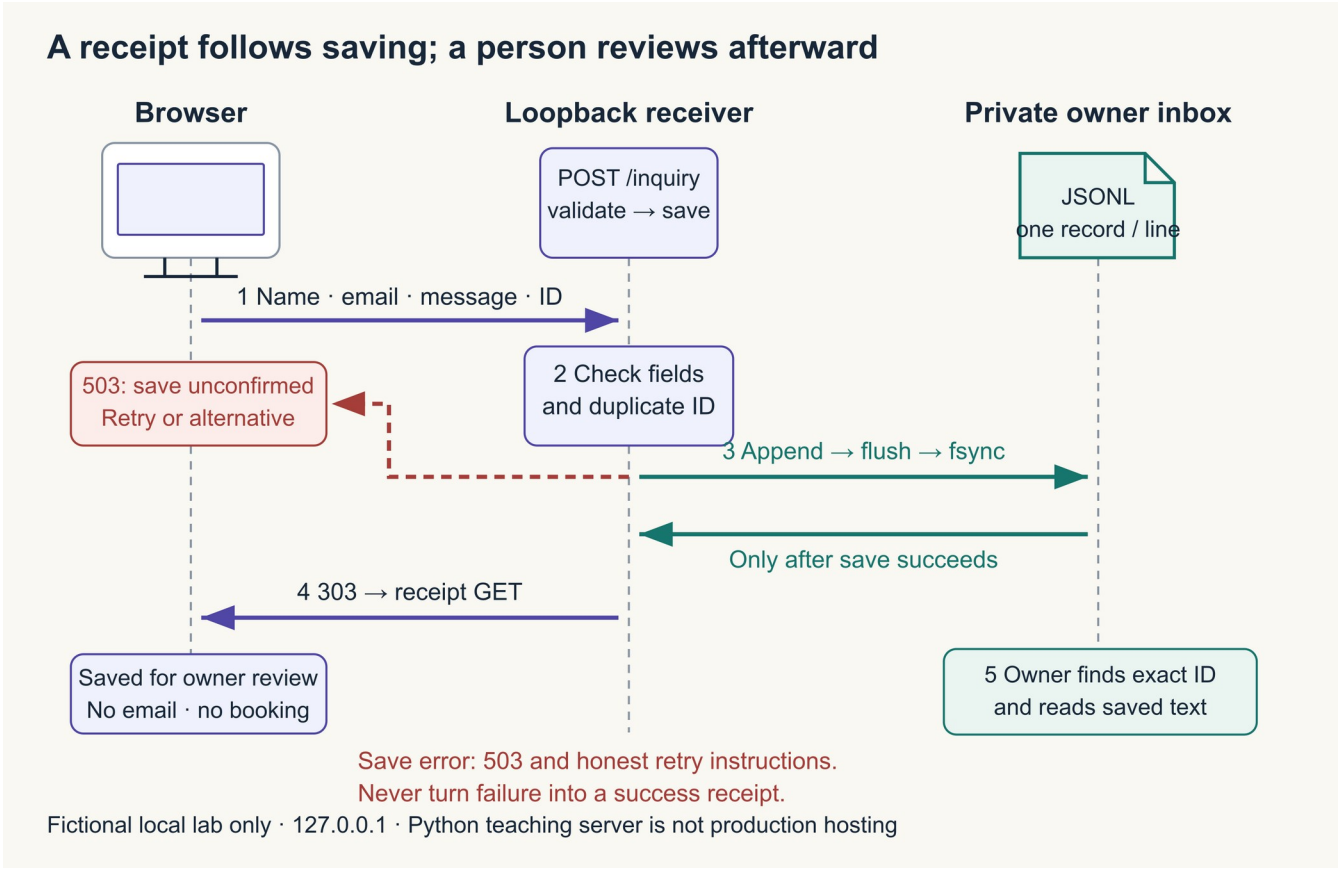
After a valid new submission is saved, the receipt says: “Saved for owner review. This local demonstration sends no email and confirms no booking.” An invalid submission explains the exact field problem and how to correct it. A storage failure says that the request was not confirmed saved and offers retry or the approved alternative. The fictional phone number is clearly an illustration. Do not show a celebratory success page before the server has completed the operation that success describes. W3C recommends clear feedback for both successful and unsuccessful submissions [S05].

Try the form using only the keyboard. Move through links and fields in a predictable order, see the focus indicator, activate Submit, and locate the resulting message. Then leave email empty and inspect the browser's response; submit malformed data through the provided local test runner to check the server response separately. Confirm that user-entered text is escaped when shown in an error page, so angle brackets remain text. The Python html.escape function supports that output step [S08]. F06 records the exact visible labels, instructions, payload names and response wording. This record becomes the contract for the server and the test ledger, not merely a design screenshot.

1. Match each visible label to its field identifier and submitted name.

2. Test required fields through the browser and separately at the server.

3. Read the success and failure copy aloud and remove every unsupported delivery promise.



Original local system diagram. HTML/Python components supported by S01–S09 and303 by S16. Local save is not email delivery or public readiness.

Practice

Can a receipt say “Your email has been delivered” when the local server has saved a JSON record?

Worked answer

No. Saving a record proves only the agreed local save. The accurate receipt is saved for owner review. Real email acceptance and arrival require separate evidence, and a person reading the message is a further operational event.

Sources: S02, S03, S04, S05, S08. Full links and limits are in the source register.

L07 Follow a request through validation and saving

Build and test the local receiving workflow. Study and exercise allowance 60 minutes.

Produce: A field-by-field server contract and a verified owner-review record

Use forms F07.

Read the receiving sequence

Inspect the specimen's receiving handler in order. It accepts only POST /inquiry, checks the content type and body length, decodes the form, validates fields, checks the request identifier, writes one record, and only then issues the receipt redirect. You do not need to memorize a programming language to trace these decisions. Write each decision beside a matching request in F07. A server response is a

description of a result, not a mechanism that makes the result true. Moving the success response above the save would break the service promise.

The local teaching contract limits the request body to 8,192 bytes. Missing or invalid length is rejected; an excessive body receives 413, a wrong content type 415, and invalid fields 422. The form decoder preserves blank values and limits the number of decoded fields. Python's parse_qs supports those controls [S06]. Reject repeated keys and unexpected field names so one person's browser and your server do not interpret different values. These particular choices are original specimen rules. They do not make the demonstration a production security framework.

DEMO-001 is a fixed identifier supplied by the automated test fixture. A normal visit to Contact generates a fresh identifier inside the form; the learner does not type DEMO-001 into the ordinary browser form. For a manual rehearsal, use one unique fictional message, submit once, then match its exact name, email and message in the private inbox and record the generated identifier shown there. A receipt page alone does not prove that the owner found that record.

Validate meaning before writing

Trim outer whitespace from name, email and message. Require exactly one of each permitted field. Check the agreed lengths and a modest email syntax rule that rejects whitespace and requires a local part and a plausible domain shape. An example.invalid address is allowed in this fictional rehearsal. Syntax does not prove ownership, consent or deliverability. Keep messages as text; do not execute them, place them into commands or interpret them as HTML. Escape text if it is shown in a page [S08]. Preserve meaningful interior spaces and punctuation rather than deleting characters until every message looks valid.

The request_id is a safe identifier supplied for the form instance. The specimen compares the normalized payload for any existing identifier. A repeated identifier with identical normalized fields returns the existing success without adding another record. The same identifier with different fields returns 409 and asks for a fresh form. This rule prevents a retry from becoming two owner tasks. It is a local, single-process demonstration of duplicate handling, not a guarantee across distributed servers or simultaneous production workers. The test runner supplies fixed identifiers so the behavior is repeatable.

Save before announcing success

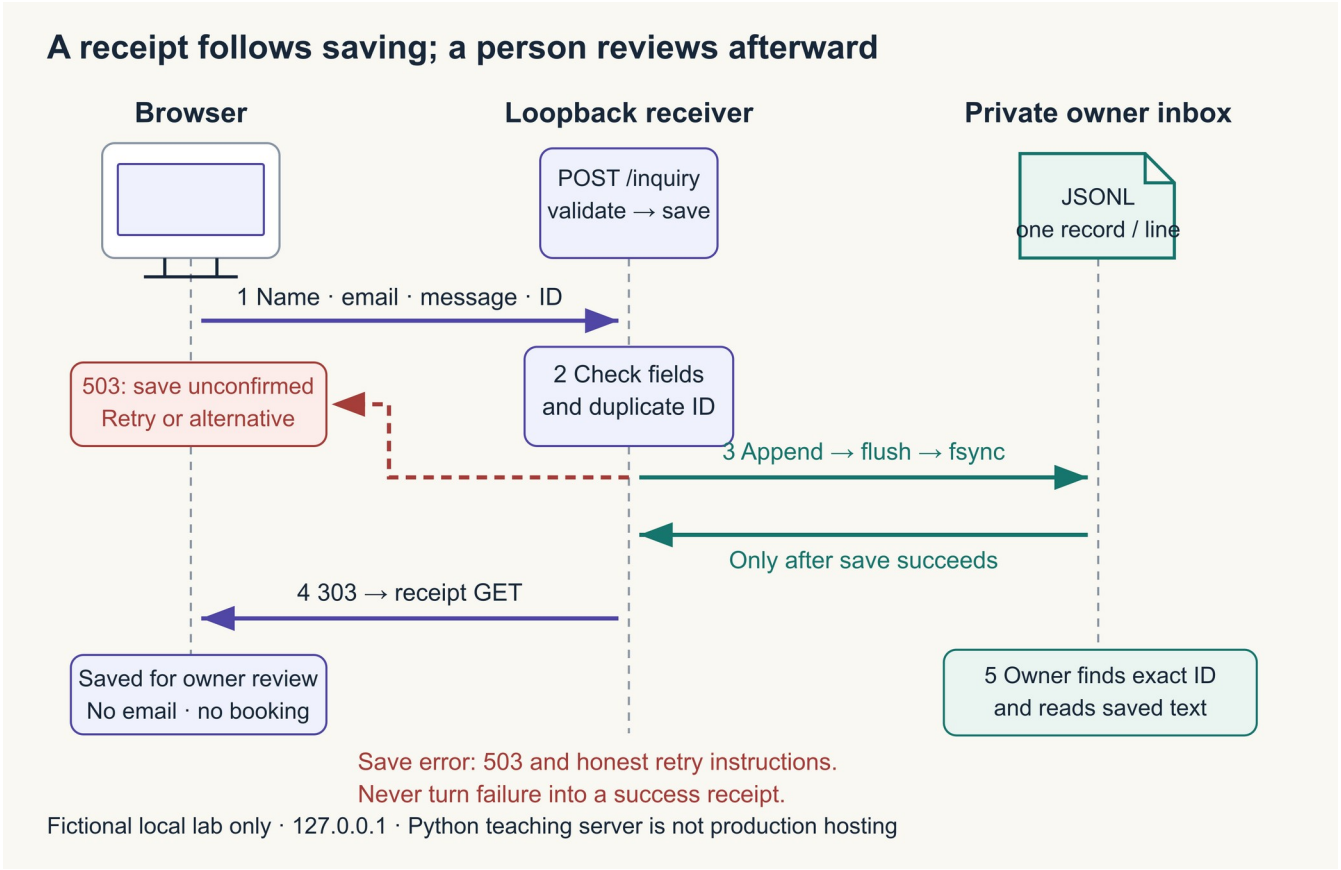
Each accepted record contains an identifier, timestamp, name, email, message and status awaiting owner review. JSON encoding keeps the structured fields together; one encoded object followed by one newline forms each JSONL record. Python's JSON documentation warns that repeatedly dumping objects does not by itself create one valid JSON document [S07]. The specimen intentionally uses a line-based record format and reads each line separately. It writes to a private local path, flushes the Python buffer and calls os.fsync before returning success [S09]. This improves the evidence for a completed write; it does not guarantee survival of every hardware or filesystem failure.

If writing or flushing fails, return 503 and do not claim success. A real failure can be ambiguous if some bytes were written before an error, so preserve the request identifier when investigating and use duplicate-aware retry. Do not delete an inbox just to make an error disappear. After a successful save, HTTP 303 directs the browser to a separate receipt retrieval [S16]. The identifier and personal fields do not need to appear in that receipt URL. Open the local inbox through the documented owner-review method, find the exact identifier and compare every field. That is the first end-to-end handoff evidence. It is still not an email or a human response to a customer.

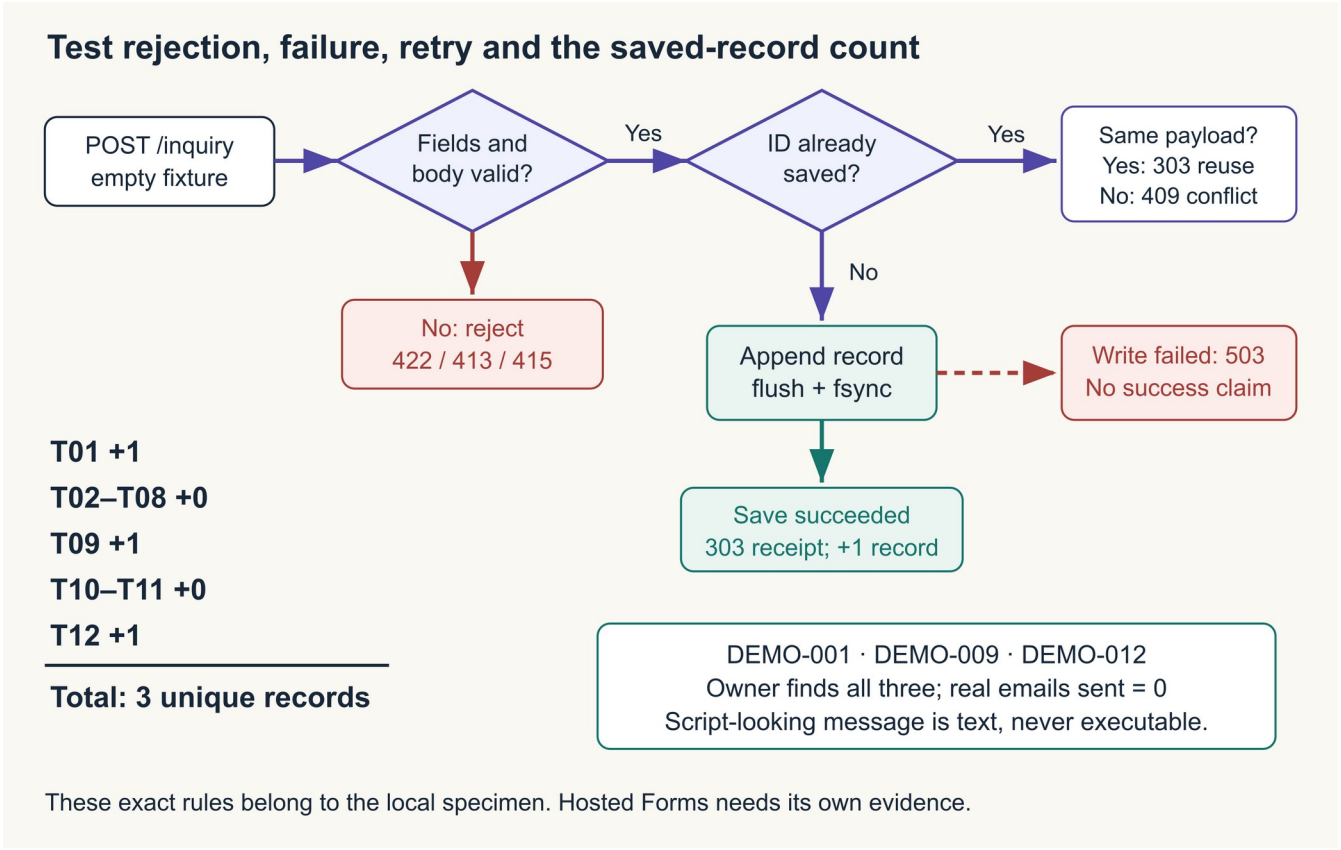
1. In the automated fixture, trace DEMO-001; in a manual rehearsal, trace the newly generated identifier and exact fictional message to one saved record.

2. Check that a failed write cannot lead to the success redirect.

3. Read the saved record and reconcile its identifier and text with F07.



Original local system diagram. HTML/Python components supported by S01–S09 and303 by S16. Local save is not email delivery or public readiness.



Original worked local request logic. Exact statuses and limits are specimen rules; managed hosted Forms does not inherit them. Sources S01, S04, S06–S09, S16.

Practice

The browser shows success, but the owner-review file has no matching record. Is the test passed?

Worked answer

No. The promised handoff is unproved or false. Stop delivery, preserve the request and response evidence, inspect route and save behavior, correct the cause, reset only the fictional fixture as documented, and repeat the complete test. A screenshot of success alone cannot establish saving.

Sources: S01, S06, S07, S08, S09, S16. Full links and limits are in the source register.

L08 Test delivery rejection and recovery

Build and test the local receiving workflow. Study and exercise allowance 60 minutes.

Produce: A twelve-case inquiry ledger with exact expected counts

Use forms F08.

Create evidence beyond the happy path

Start from an empty fictional inbox using the specimen's documented reset procedure. Confirm you are in the rehearsal folder before resetting anything. Record the date, specimen version and starting count zero in F08. The provided test runner exercises the receiving endpoint and checks saved records; browser checks still matter because a protocol test cannot show whether a person can use the form.

Keep expected results separate from observed results. The fictional ledger below is a worked answer, not evidence that your own copy has passed.

T01 submits DEMO-001 with Avery Test, avery@example.invalid and “Please explain your ordinary exterior cleaning service.” Expected: 303 followed by a receipt, one new record, owner review finds it. T02 leaves email empty: 422, no new record. T03 uses email without an at-sign: 422, none. T04 exceeds 8,192 body bytes: 413, none. T05 sends the wrong content type: 415, none. The total remains one record. The browser's own required-field warning does not substitute for T02's direct server test.

Prove failure and retry behavior

T06 activates the test-only failing sink in a separate fixture before any save. It submits DEMO-006 and expects 503, an honest failure page and zero new records. Restore the normal sink after the test. This switch is a local test arrangement, never a public visitor field. T07 repeats DEMO-001 with exactly the same normalized content: 303, zero new records and still one copy of that identifier. T08 reuses DEMO-001 with a different message: 409, no overwrite. These tests ask whether a visitor's retry can create duplicate work or silently replace a request.

T09 submits DEMO-009 with the message “Please show <script>alert(1)</script> as text only.” The allowed message is stored as plain text and any displayed version is escaped; expected 303, one new record, no executed script. T10 requests an unknown route: 404, no new record. T11 attempts to retrieve /inbox.jsonl: 404, no inbox content. T12 submits a second ordinary request, DEMO-012, with Riley Sample and riley@example.invalid: 303 and one new record. End with exactly three unique accepted records: DEMO-001, DEMO-009 and DEMO-012. The owner-review exercise locates all three. Real emails sent: zero.

Interpret the result and recover safely

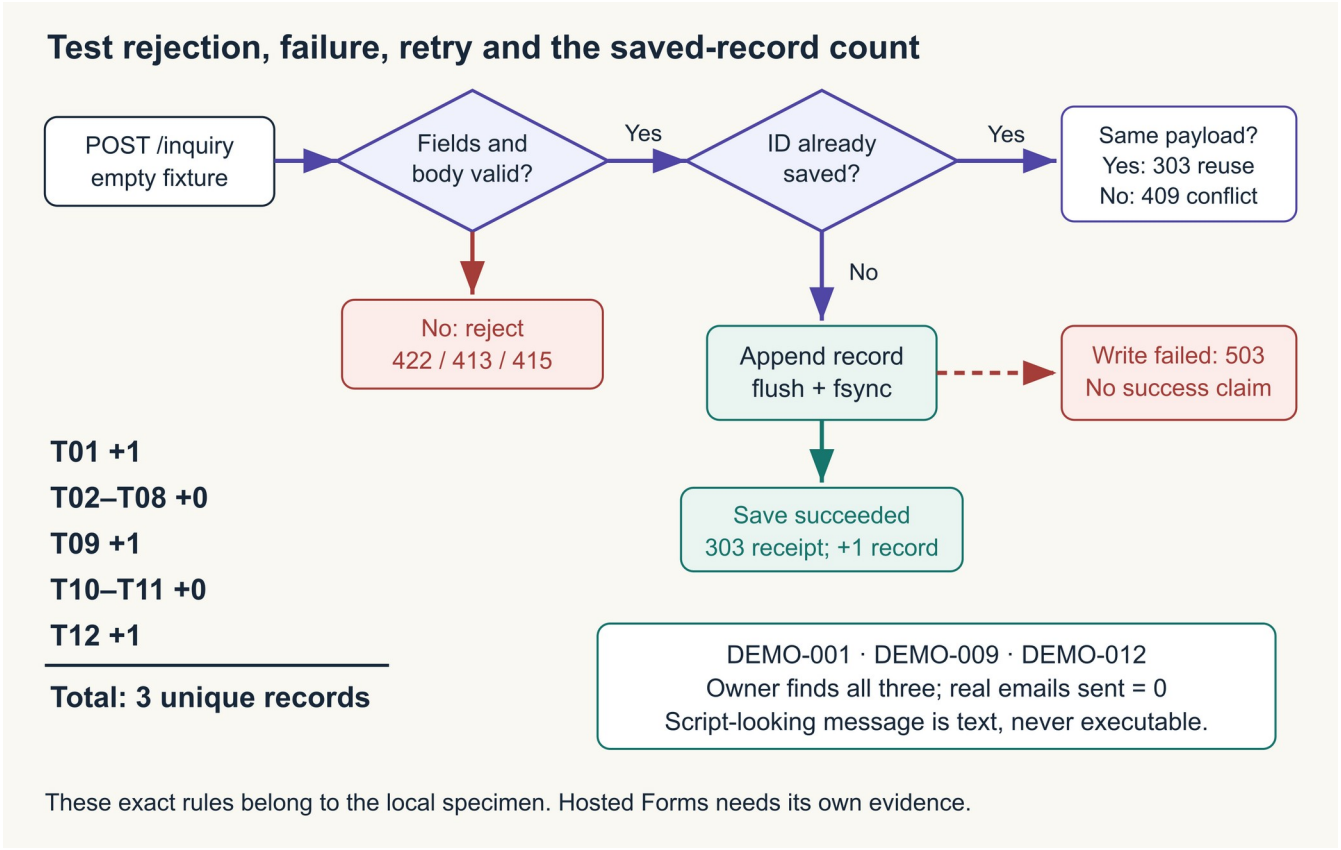
For every row record request identifier, expected status, observed status, added-record count, owner-review evidence and pass/fail. The expected new-record vector is 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1. Its sum is three. A redirected browser may display the receipt's final 200; record the initial POST response as well as that final page. Otherwise you might mistake a successful page load for successful submission. The workbook's ledger calculates counts and flags disagreement, but you must inspect the actual saved content and user messages.

If a test fails, preserve the evidence, identify the smallest affected component, change one cause, and rerun the failed test plus dependent success/retry checks. If escaping changes, repeat T09 and the error-page cases. If persistence changes, repeat the entire inquiry ledger. After any reset, distinguish the old evidence from the new run. Before handoff, repeat the browser success path with a fresh fictional identifier and have the owner role find it without your coaching. W3C's notifications guidance supports clear outcome communication [S05]; Python's local-server warning remains in force [S01]. A twelve-row pass proves this bounded rehearsal, not production security or comprehensive accessibility.

1. Run protocol tests on an empty fictional fixture and reconcile exactly three accepted records.

2. Repeat one success and one failure in the browser.

3. Save actual observations separately from the fictional expected ledger.



Original worked local request logic. Exact statuses and limits are specimen rules; managed hosted Forms does not inherit them. Sources S01, S04, S06–S09, S16.

Practice

A browser reload repeats the receipt GET and the count stays three. Does that prove duplicate POST handling?

Worked answer

No. A receipt reload may send no POST at all. T07 must repeat the same POST identifier and normalized payload and prove that the record count does not increase. Both redirect behavior and duplicate-aware receiving behavior need their own evidence.

Sources: S01, S05, S16. Full links and limits are in the source register.

L09 Review usability content and accessibility

Build and test the local receiving workflow. Study and exercise allowance 45 minutes.

Produce: An issue log with reproduced defects, corrections and remaining limits

Use forms F08.

Check a real task at different sizes

Ask a permitted reviewer to act as a first-time visitor using only fictional information. Give a task: “Find out whether this business handles ordinary residential exterior window cleaning and send a question.” Watch where they hesitate. Do not begin by explaining which button to press. Record their words and the step where confusion occurs. A task observation identifies a concrete defect: “The service area is

missing from Home” is actionable; “make it pop” is not. Keep the reviewer task separate from the protocol tests that count saved requests.

Inspect a narrow viewport around 360 CSS pixels and a wider one around 1280, then try an actual phone if available and permitted. These widths are specimen checks, not a complete device list. Look for horizontal scrolling, clipped labels, images covering text and buttons outside the visible page. Increase browser zoom to 200% and check that content and controls remain available. Read long messages and long business names. A layout that works only with the shortest sample strings is not ready for ordinary content.

Use preliminary accessibility checks honestly

Work through page titles, heading order, useful image alternatives, visible keyboard focus, keyboard operation, labels, errors and readable contrast. W3C's Easy Checks are explicitly preliminary; passing them does not establish full accessibility [S11]. Use the source to learn each check and record what you actually examined. An automated tool can help locate issues, but a green score cannot replace keyboard use or explain whether the words make sense. Do not issue an accessibility certificate or legal compliance promise from this course's short review.

On the contact form, click the label and confirm it selects the intended field. Move through the page with Tab and Shift+Tab; activate links and buttons without a mouse. Submit an invalid request and locate both the overall error and the field guidance. Make sure the error is not communicated by color alone. Then correct it and verify the saved-record path. If you use a screen reader, record the tool and task; if you do not, say that check was not performed. Honest limits are more useful to the client than a vague statement that everything is accessible.

Record defects as reproducible work

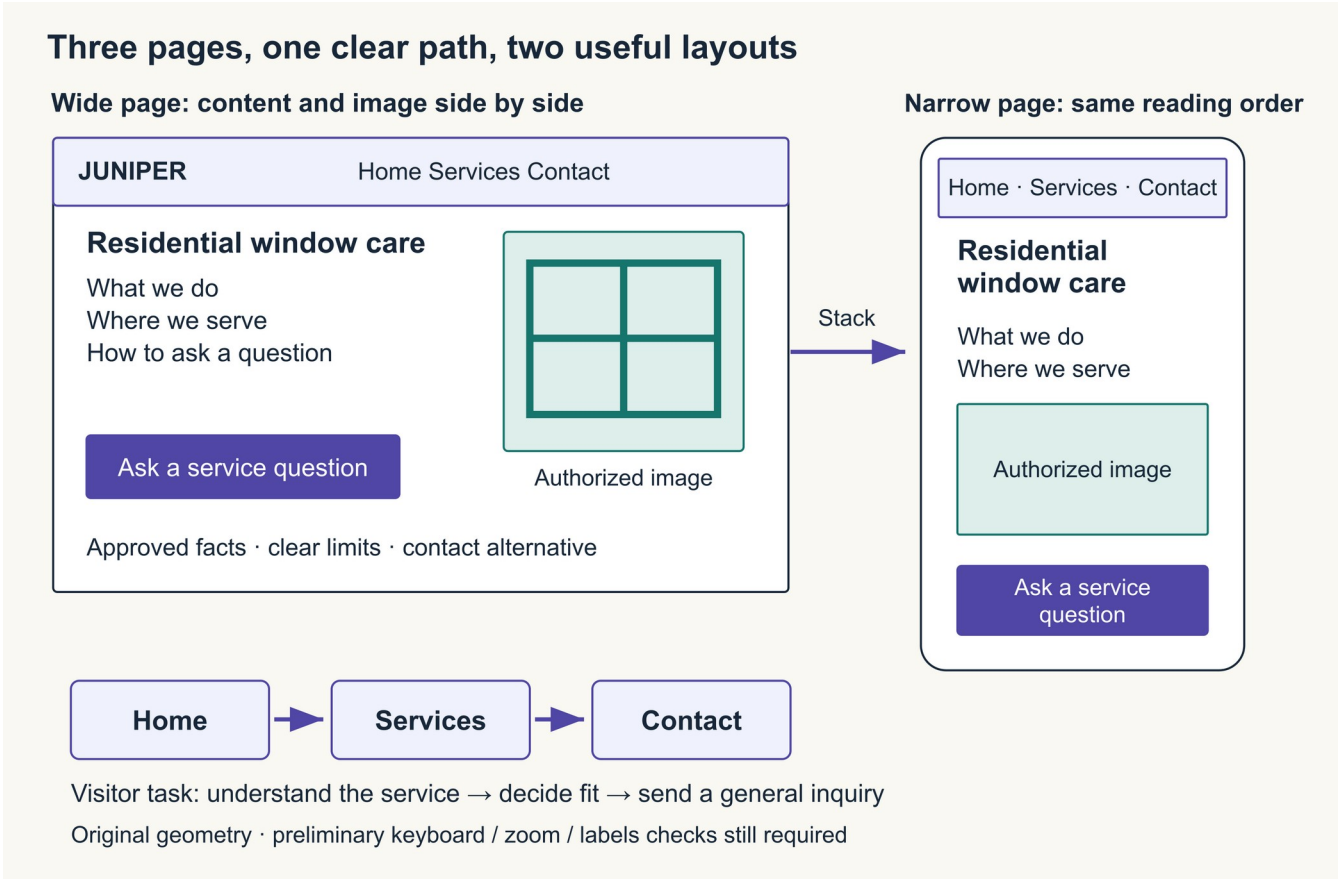
Each issue needs a page, steps, expected result, observed result, severity, owner and retest evidence. “Contact, narrow view, enter 1,000 characters; message box covers Submit; expected control remains reachable” gives you something to fix. Correct content, routes and functional blockers before polishing subtle styling. After a change to navigation, retest every page route. After a change to form markup, repeat keyboard checks and the browser submission route. Do not assume a copy correction cannot break a link or a quoted attribute.

Compare all visible claims with the approved content inventory. Remove placeholders, unsupported testimonials and any accidental live-looking email address. Review the receipt, failure page and printed handoff as well as the attractive Home page. Search the project for words such as guaranteed, delivered, booked and compliant; keep them only where they are supported and accurate. Finally, ask the reviewer to find the alternative contact instruction and explain what happens after submission. F08 records remaining issues and their consequences. A client may accept a cosmetic difference, but a missing save, exposed inbox or false success message is a blocker for the promised prototype.

1. Run one visitor task without coaching and record the evidence.

2. Check keyboard, zoom, labels and errors against W3C’s preliminary guidance.

3. Retest affected routes and inquiry cases after each correction.



Original page geometry for the fictional business. W3C preliminary review guidance S11; design choices do not establish accessibility conformance.

Practice

An automated score is high but a keyboard user cannot reach Submit. What takes priority?

Worked answer

Treat the blocked submission as an unresolved functional and accessibility issue. Reproduce and correct keyboard access, then retest the complete task. The score does not cancel observed evidence, and the preliminary review does not establish comprehensive conformance.

Sources: S03, S05, S11. Full links and limits are in the source register.

L10 Estimate the whole project before choosing a price

Estimate,agree and manage the project. Study and exercise allowance 55 minutes.

Produce: An independent 2,100-minute full-project baseline and separate cash/economic model

Use forms F09.

Count time that is easy to hide

The fictional Oregon live-site baseline contains seventeen stages: discovery 90 minutes; scope/access planning 60; content preparation 150; information architecture 90; page building 300; local form/server integration 180; quality/accessibility review 120; local delivery/failure tests 150; one content revision 90; local handoff 120; administration/invoice 60; used follow-up 60; backup/reset/closure 30; hosted form

adaptation 120; DNS/TLS verification 120; hosted acceptance/rollback tests 180; and live owner handoff 180. Total 2,100 minutes, or 35 hours. The first thirteen stages total 1,500 minutes; the hosted extension adds 600. No travel occurs. An unused three-business-day calendar buffer is separate, with no labor cost until used. All approvals, accounts and live results are fictional assumptions; no actual hosted project has been established.

Estimate these stages from your own permitted rehearsal before selecting a price. A novice may need much longer. The case's times are invented teaching values, not a promise about beginner speed. Include reading provider documentation and arranging any production stage in a later quote; neither disappears because software is inexpensive. Keep an accepted dated baseline unchanged. Record actual minutes in separate cells so a late revision does not rewrite your original promise or conceal an estimating error.

Separate money paid from management costs

The invented full-project price is $1,800 before tax. In this teaching case both provider and customer operate in Oregon and the work is supplied there; assume no other-state nexus or special tax applies. Oregon DOR states that Oregon has no general sales or use/transaction tax [S18], so the modeled general sales tax is $0 and invoice total $1,800. This does not resolve income tax, local business taxes or out-of-state obligations. Verify actual facts before invoicing. A fictional processing assumption of 3% of gross plus $0.30 for one payment gives $54.30. No processor is selected or payment processed. The separate 8% stress exercise is not an Oregon rate: $1,800 + $144 = $1,944 gross, fee $58.62.

Job-specific cash costs are $30 for an authorized independent fictional test review and $15 for project-specific transcript preparation, totaling $45. They exclude inventory, equipment purchases and subscriptions allocated elsewhere. There is no physical stock: stock_items is empty and stock purchase/consumption are zero. Allocate $30 of existing computer/tool use and $72 general overhead: $48 existing software subscription allocation and $24 administration. These $102 allocations are economic costs, not another $102 paid for this project. Client-owned domain, hosting and mailbox subscriptions are paid directly by the client and excluded from provider revenue and costs. Vehicle use is zero. Owner value at the invented $30 per hour is $1,050.

Compute the result and the cash position

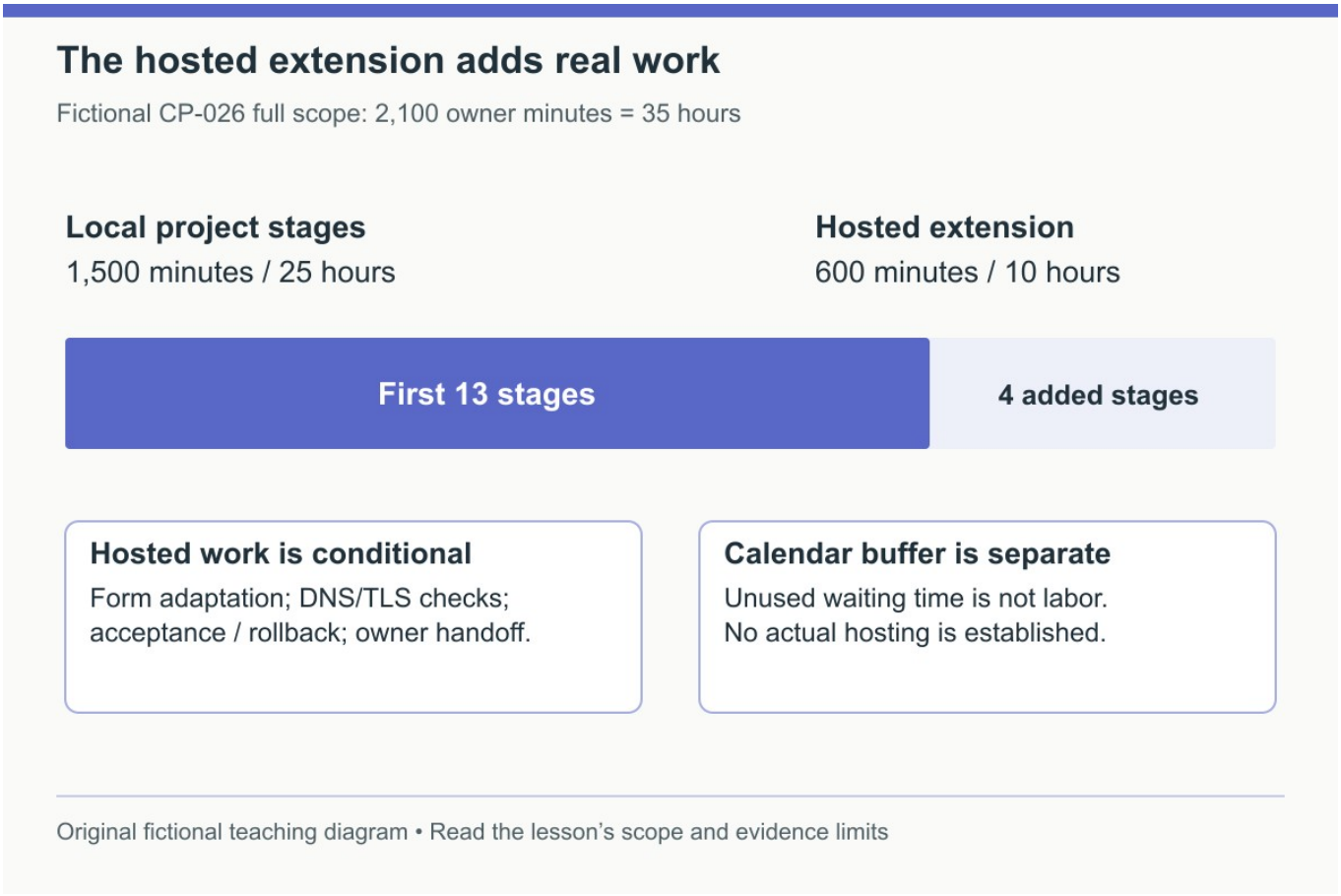
Total management cost is $45 + $30 + $72 + $1,050 + $54.30 = $1,251.30. Pretax revenue $1,800 leaves $548.70 before a designated $90 business reserve, or $458.70 afterward. This remainder is not take-home pay, tax net income or a market-rate claim. Reserve designates bank cash and is not an expense or automatic deduction. With the separate paid campaign’s $180 cash-plus-owner acquisition cost, $368.70 remains before reserve and $278.70 after reserve. Do not count that campaign again in overhead. At a price selected after estimating, the example has room for acquisition and correction; a learner’s actual time may produce a different result.

With opening cash $900, receipt $1,800, retained processing fee $54.30 and job cash $45, closing cash is $2,600.70 before an owner draw. There is no general sales tax collected or remitted in this Oregon model. Bank cash exceeds management profit because owner labor and allocated equipment/subscriptions were not paid again by this job. IRS guidance supports records of income, expenses and evidence [S17]; this worksheet is not a tax return. In the optional 8% arithmetic stress case, remitting $144 gives $2,596.38 closing cash and management profit $544.38 before reserve. Record any actual processor fee from evidence, separately from an estimate or a later bank debit.

1. Add all seventeen stages independently; preserve the accepted baseline date.

2. Calculate gross invoice and fee before profit.

3. Reconcile cash separately from owner value, allocations and reserve.



The conditional hosted extension adds 600 minutes to 1,500 local-project minutes. These invented 35 owner hours do not establish a real deployment or beginner completion time.

Practice

An extra 120 owner minutes are needed and nothing else changes. What is the revised management profit?

Worked answer

At $30 per hour, 120 minutes add $60 owner value. Cost rises from $1,251.30 to $1,311.30; profit falls from $548.70 to $488.70 and after the unchanged $90 reserve becomes $398.70. Extra unpaid owner time does not itself change bank cash.

Sources: S17, S18. Full links and limits are in the source register.

L11 Confirm the exact deliverable and approval

Estimate,agree and manage the project. Study and exercise allowance 45 minutes.

Produce: A complete fictional estimate and a clear real-project approval boundary

Use forms F10.

Write an estimate someone can approve

A useful estimate names the sender, customer, date, validity, exact deliverables, assumptions, price and next decision. F10 contains a complete fictional estimate from Casey Rowan, Clear Path Web Studio in Salem, Oregon, casey@clearpath.example.invalid, to Morgan Lee, Juniper Window Care in Salem, Oregon, morgan@juniper.example.invalid. Dated November 23, 2026, it is valid through November 30. Client content is due November 24, prototype review December 1, conditional live acceptance December 4 and follow-up December 11. The full live stage is contingent on separately evidenced account ownership, publication permission, domain control, appropriate provider terms and successful hosted tests. These fictional assumptions establish no real client authority.

Included: Home, Services and Contact; owner-approved text and one authorized image; local fictional-data rehearsal; static hosted adaptation using the documented managed Forms route; one owner email notification; client-owned domain/DNS/TLS setup within demonstrated competence; local and hosted acceptance ledgers; one content revision up to 90 minutes; a rollback rehearsal; editable source; local and live owner handoffs; and one 60-minute used follow-up allowance. Excluded: account purchases and provider renewal charges, booking, payments, uploads, sensitive/regulated records, automatic visitor replies, custom server validation/idempotency in the hosted route, analytics, ranking guarantees and ongoing maintenance. The optional prototype-only first offer requires its own clearly reduced scope and independently estimated price.

Make terms and responsibilities concrete

The full fictional price is $1,800 service, $0 modeled Oregon general sales tax, $1,800 total, payable once after accepted live handoff on December 4. No deposit is modeled. Both fictional parties and delivery are in Oregon; verify actual tax obligations and any cross-state facts before a real invoice [S18]. Client owns and directly pays domain, hosting and mailbox services, supplies content/asset rights, confirms recipients and publication, and controls recovery. Provider uses limited authorized access, protects records and delivers test/rollback evidence. Approval for local review alone never authorizes publication, purchases or sending real messages.

Changes require a written description, effect on price and date, and client approval before work. For this invented agreement, cancellation before work has no fee; after authorized work begins, charge only documented completed stages up to the $1,800 service cap, deliver the paid work and credit any prepaid unearned amount. Actual cancellation terms require review for the jurisdiction and customer relationship. Correct a reproducible departure from the approved scope reported within seven days of handoff without an extra service charge; new features and later third-party changes require a separate estimate. No term promises continuous monitoring.

Record approval without inventing permission

The fictional approval record says: “Morgan Lee approved CP-026 on November 23, 2026: $1,800 full three-page website/inquiry project, subject to verified client-owned accounts and separate written publication/notification authorization on December 3 after prototype review. No card processing, bookings or subscription purchases by provider authorized.” The case assumes those later gates and live tests are fulfilled; that assumption is explicitly fictional, not actual evidence. Your real project remains on hold until its own authorized scope, provider access, privacy and readiness checks are established. Silence or a design compliment is not a substitute.

For a real client, separate the two decisions in ordinary language: “First you review the local design and inquiry demonstration. Then, after ownership and tests are ready, you approve public pages and one owner notification.” If the client buys only the optional prototype, state that visitors cannot use it publicly and it sends no email. If they buy the full scope, name the additional hosted work, recurring costs and

acceptance conditions before quoting. Keep the approved version and every change. A precise estimate makes the system testable and preserves the client's ability to operate or transfer it later.

1. Read the prelaunch boundary and total aloud to the decision maker.
2. Store the actual approved version before beginning paid work.
3. Keep deployment, purchases and real messaging outside prototype approval.



Design feedback covers a specific page version. Public launch needs separately evidenced account ownership, supported tests and actual publication authority.

Practice

The client approves the colors and says "looks good." Is that approval to buy a domain and publish?

Worked answer

No. Design feedback is not the separately defined authority to spend, select account ownership, publish or process real inquiries. Record the design feedback and seek the actual launch authorization only after a concrete launch scope and readiness evidence exist.

Sources: S10, S14, S18. Full links and limits are in the source register.

L12 Control changes and cost an adverse outcome

Estimate,agree and manage the project. Study and exercise allowance 50 minutes.

Produce: A scope-change ledger and a fully costed failure branch

Use forms F11.

Classify the request before doing it

At review, Morgan asks for a new emergency-service page and automatic appointment booking. Compare the request with the approved three pages and owner-review inquiry route. Both are new scope; booking also crosses an excluded boundary. Record the exact request, why it changes the workflow, the safe alternative and a decision. Do not insert a booking promise into the existing receipt while deciding. A small sentence can create a large operational obligation. Explain that the prototype cannot confirm availability or dispatch urgent service.

The accepted fictional change is narrower: one ordinary FAQ section on Services with approved copy. Estimate 60 minutes to prepare/build, 30 to review and 30 to retest: 120 extra minutes. Quote an invented $180 service change, $0 Oregon general sales tax, $180 gross. If collected as a second payment using the same fictional fee assumption, its fee is $5.70. Owner value adds $60. With no other extra cost, incremental contribution is $114.30. The changed full project is $1,980 service and 2,220 owner minutes. Total fees are $60, total management cost $1,317, profit $663 and after the base $90 reserve $573. Keep this optional approved-change scenario separate from the base.

Handle a real functional failure honestly

A different adverse branch begins when the fictional final review finds that accepted requests were never written to the owner inbox. The apparent success screen is insufficient evidence. Stop acceptance, preserve the run and tell the client exactly what is unproved. Do not describe the prototype as delivered or silently add the missing time to a future maintenance bill. The provider corrects a failure to meet the agreed function under the correction terms, with no extra service charge. Retest the entire inquiry ledger after changing persistence behavior.

The adverse branch adds 240 minutes: diagnosis 60, correction 90, full retest 60 and client explanation 30. Total time becomes 2,340 minutes, 39 hours, and owner value $1,170. Pay another $20 independent retest, bringing cash job cost to $65. Keep service price $1,800, Oregon general sales tax $0 and fee $54.30. Management cost is $1,170 + $65 + $30 + $72 + $54.30 = $1,391.30. Profit before reserve is $408.70 and after the $90 reserve $318.70. Closing cash is $2,580.70. The successful base remains unchanged in its own column.

Show the downside all the way through

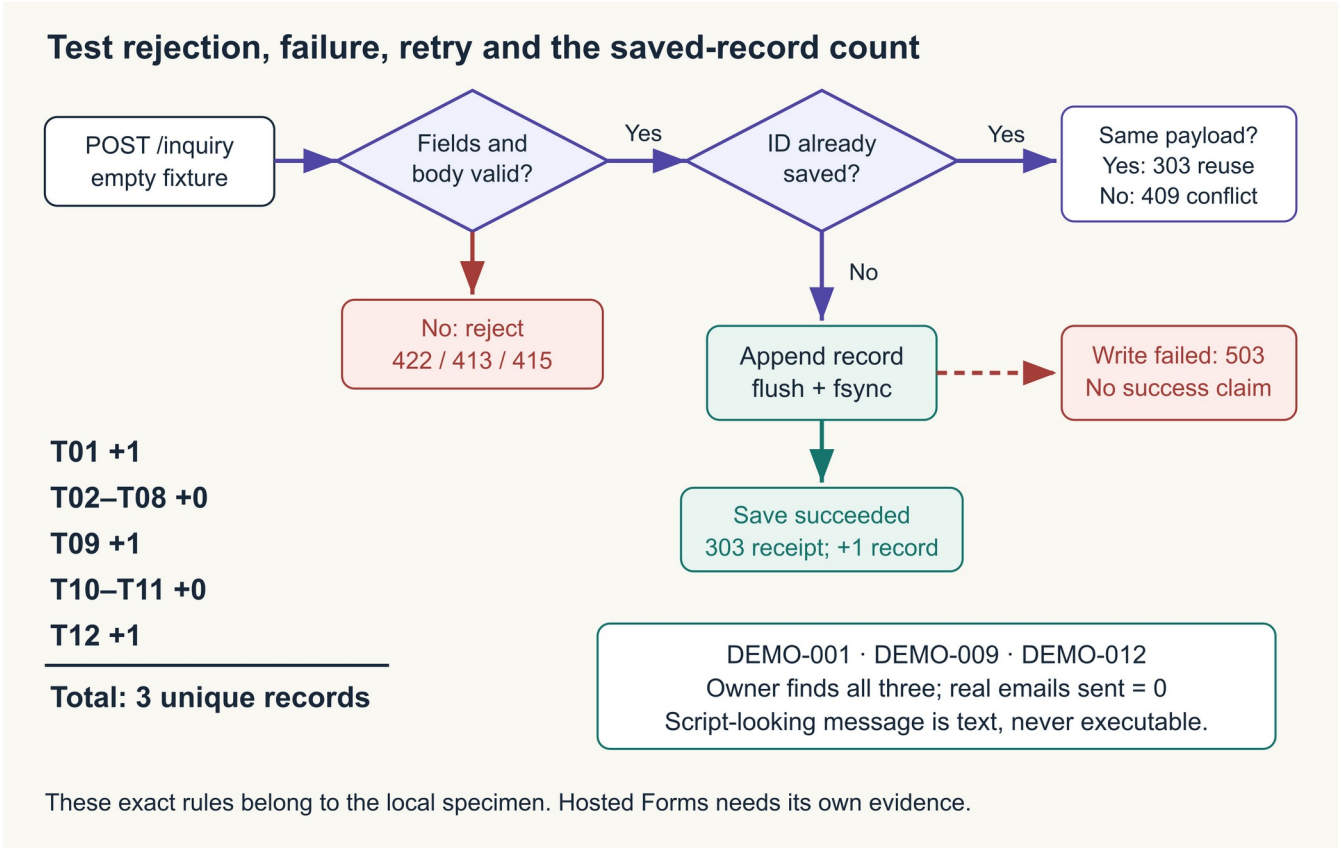
If correction still fails acceptance and the provider gives a full $1,800 refund under an agreed resolution, revenue returns to zero. Assume the original $54.30 fee is retained and there is no invented extra refund fee. With 2,340 minutes and $65 cash job cost, the economic loss is $1,391.30; after a $90 cash designation the management remainder is negative $1,481.30, although reserve is not another expense. Closing cash is $780.70: $900 + $1,800 − $54.30 − $65 − $1,800. No Oregon general sales tax was collected, remitted or refunded in this case. A different jurisdiction would need its own lawful tax adjustment, not an assumed instant recovery.

F11 separates requested change, approved change, correction and cancellation/refund. These are different events. The optional FAQ change is profitable only under its own assumptions; it cannot erase the delivery failure. Use the ledger to improve future estimates: more persistence evidence earlier, a clear test fixture and a realistic review allowance. When a prospect needs a feature you cannot support, a reasoned decline is a valid business outcome. The aim is to deliver a promise you can verify, not to keep every inquiry at any cost.

1. Record the FAQ alternative and written approval before changing scope.

2. Cost diagnosis, correction, retest and communication in the failure branch.

3. Keep tax recovery pending until actual lawful evidence exists.



Original worked local request logic. Exact statuses and limits are specimen rules; managed hosted Forms does not inherit them. Sources S01, S04, S06–S09, S16.

Practice

Can the provider charge the client the $180 FAQ fee for fixing an inbox save that was already included?

Worked answer

No. The FAQ is optional new content; restoring the promised save function is a correction in this fictional agreement. Preserve separate records and obtain approval only for actual additional scope. The failure branch adds time and cash to the provider's cost without increasing the base price.

Sources: S17. Full links and limits are in the source register.

L13 Handoff the prototype and define production evidence

Handoff,host conditionally and operate. Study and exercise allowance 55 minutes.

Produce: An owner-operated handoff and a specific launch gap register

Use forms F12.

Teach operation then observe it

Handoff is a demonstration by the owner, not a file attachment followed by "let me know." Deliver editable pages, styles, server source, the README, test fixtures, actual test report, approved content

inventory, scope/change log and the local inbox instructions. Mark the whole set prelaunch and fictional where appropriate. In the fictional 120-minute handoff, spend 30 minutes on files and starting/stopping, 30 on submitting and finding a request, 30 on failure/recovery and 30 on ownership, limits and next decisions. Keep passwords and real customer records out of the pack.

Ask the owner role to start a clean copy, change an approved sentence, submit a new fictional inquiry and locate it in the inbox without your prompts. Then stop the server and restore the clean copy. Record what they did and where they needed help. Explain that a computer being off means this local prototype is unavailable. The inbox file is not a hosted mailbox, backup service or multi-user dashboard. The owner must know these limits before signing off. The corrected acceptance record describes observed functions and remaining gaps, not “everything works.”

Specify the production adapter without inventing settings

A live stage needs a client-owned public domain, HTTPS hosting, a protected receiving store, an approved notification recipient, appropriate abuse controls, backup/recovery and an operator. The selected conditional production path in L14–L15 is static Netlify hosting plus Netlify Forms storage and one owner email notification. It replaces the local Python receiver; Python’s teaching http.server is not recommended for production [S01]. No service has been provisioned here. Verify current provider terms, plan, roles, limits, recipient confirmation and data handling against the client’s actual needs before any purchase or publication.

Separate states remain essential: browser submitted, managed record observed, notification observed in the mailbox, owner reviewed and follow-up assigned. The selected simple hosted route is not claimed to provide custom transactional idempotency, local status codes or a programmable retry queue. The owner reconciles the managed submissions list and mailbox at the agreed interval and resolves duplicate-looking requests manually. If those limits do not fit the client’s needs, stop and estimate a supported alternative receiver. A successful page or a configured notification address does not prove arrival or reading; preserve stored inquiries during investigation.

Make launch a concrete separate decision

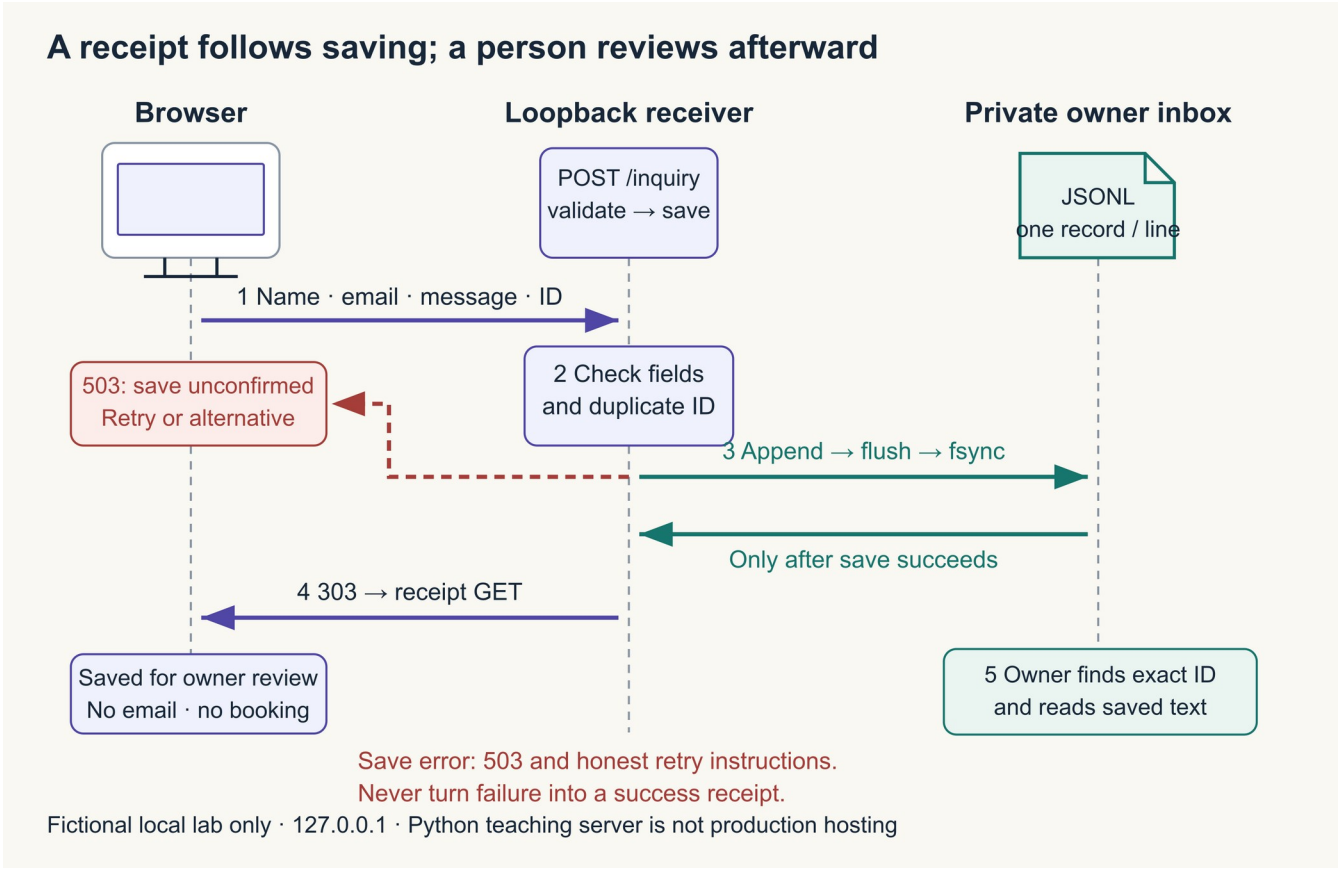
Before any public stage, run actual end-to-end tests with client-approved test data and the real destination. Record the request identifier, server acceptance, durable record, notification event, mailbox location including spam/quarantine where applicable, and the owner's successful review. Test an invalid request, an unavailable receiving dependency, notification failure, duplicate retry, private-data exposure, backup restore and account recovery through the selected documented system. Revisit privacy wording and retention across every copy. FTC's guidance supports minimizing data and access [S10]; a short test list is not comprehensive security certification.

The handoff inventory names the owner, payer, recovery contact, maintenance duty and renewal dates for every service. Remove temporary access only after the client demonstrates operation and recovery. Agree whether follow-up is the base project’s scheduled check or a separate maintenance contract. The course’s actual lab remains local, and no real production readiness is established. The fictional full case assumes independent live evidence for teaching economics; your project must supply its own evidence. The optional prototype-only delivery closes with the production gap register open, while a real full project closes only after the additional hosted acceptance.

1. Observe the owner starting, using, finding and restoring the local project.

2. Deliver the open production gap register beside the accepted prototype.

3. Require real storage, notification and mailbox evidence before a later launch promise.



Original local system diagram. HTML/Python components supported by S01–S09 and303 by S16. Local save is not email delivery or public readiness.

Practice

A selected future email API reports accepted, but the owner cannot find the message. Can launch delivery testing be marked complete?

Worked answer

No. Preserve the accepted request, investigate the documented notification path and mailbox controls, and keep delivery unresolved until the agreed destination is observed. API acceptance and owner review are separate states. The local prototype never claims either real email arrival or public readiness.

Sources: S01, S10, S11. Full links and limits are in the source register.

L14 Prepare a documented live website and inquiry route

Handoff,host conditionally and operate. Study and exercise allowance 60 minutes.

Produce: A conditional deployment plan for client-owned hosting, forms, DNS and TLS

Use forms F13.

Choose one supported hosted architecture

The full business can offer a live three-page informational website with one inquiry notification after the additional gates are met. This lesson selects a concrete conditional path: static HTML on Netlify, Netlify Forms as the managed receiving store, and one email notification to the client's approved recipient. It

replaces the local Python receiver; it does not upload it. The course and review site remain local and unpublished. No account, purchase, domain change, deploy or email is performed by reading this lesson. Actual client authorization and provider suitability must be established separately.

Start a live project only in a client-owned account with client-controlled billing and recovery. Check current plan limits, total recurring charges, data terms, roles and ownership before purchase; do not assume that a Forms feature price means the entire site is free. Netlify documents different Forms billing for different plan generations [S25]. Obtain the smallest supported access role that can complete the authorized work and plan its removal. Save a clean static output folder containing only the approved public HTML, stylesheet, authorized image and receipt page. Exclude the Python server, private data, tests, credentials and business records.

Adapt the form and confirm receiving

In the hosted copy, use a named static HTML form with method POST and data-netlify="true". Enable automatic form detection in Forms, then deploy the form-bearing HTML. Netlify parses HTML at deploy time and injects a hidden form-name field matching the form's name [S19]. A custom success page is configured through the form action under that documented setup. Use name, email and message controls with the same clear labels and browser assistance as the lab. The local request identifier and Python validation contract do not transfer automatically. Do not claim the managed service enforces your local length, rejection-code or duplicate rules without separate implementation and tests. For this original static copy, use <form name="inquiry" method="POST" data-netlify="true" action="/received.html">, keep the named name, email and message controls, and create received.html in the public output. Remove the local server-generated request_id field; it has no automatic hosted duplicate-handling effect.

For this bounded managed route, the owner handles duplicate-looking inquiries manually by comparing the recorded sender, time and message. If automatic idempotency or strict custom server validation is a requirement, this simple selected route no longer meets the scope; define and support a separate receiver instead. Netlify's documented spam filtering separates verified and spam submissions [S24]. Review both during acceptance. Choose the documented honeypot only after assessing usability and privacy; do not add a CAPTCHA or another provider casually. The form receipt acknowledges submission, not email arrival, a booking or a personal response.

Deploy deliberately and connect the address

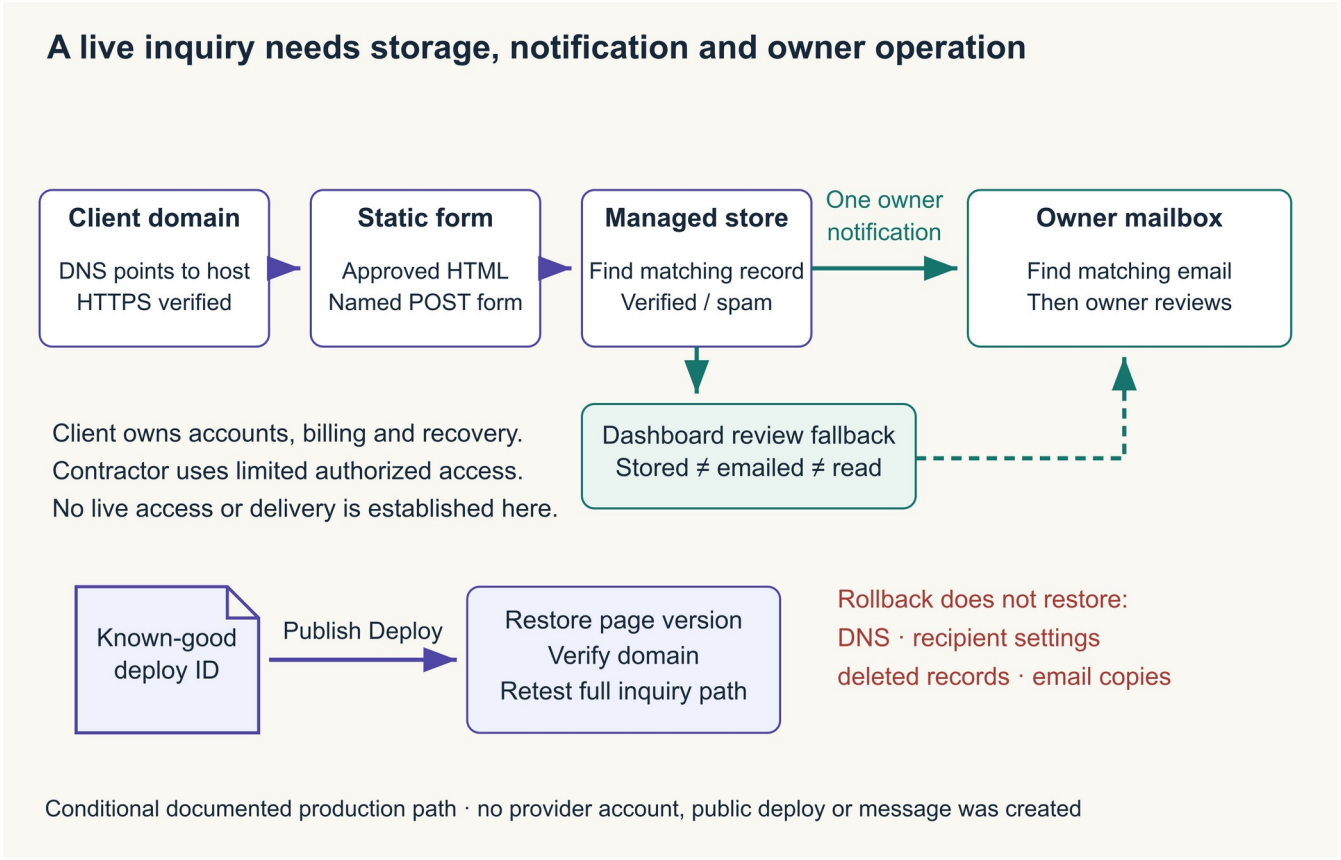
Manual folder deployment is a concrete option for this static project: after explicit publication approval, use the client team's project/deploy interface to upload the approved output folder, record the deploy identifier and inspect every page at the resulting URL [S22]. Treat any preview URL as potentially public unless its actual protection is verified. Use fictional test data and no confidential drafts. Before adding a custom domain, record the existing DNS configuration and confirm the owner controls it. Do not replace nameservers or remove mail records as an experiment.

For externally managed DNS, first assign the owned domain to the site, then open Domain management → Production domains → Pending DNS verification for that site's exact required records [S23]. A subdomain uses the provided CNAME target; the apex follows its separately documented supported record method. Never copy an IP address from an old tutorial when the account provides current values. After propagation, verify the intended domain and redirects. Netlify documents managed TLS certificates for hosted content [S26]; confirm actual certificate validity and HTTPS operation on every intended hostname. DNS and TLS success do not establish form saving or email delivery. Those are separate acceptance tasks in L15.

1. Prepare public-only output and record ownership/permission before any upload.

2. Enable form detection, deploy the adapted static form and inspect its detected name.

3. Use the project-specific DNS instructions; verify HTTPS and preserve the old DNS record.



Original conditional hosted architecture based on Netlify official docs S19–S27. No actual account, deploy, DNS, TLS or mailbox evidence is established.

Practice

Can the local /inquiry handler and its 422/409 rules be assumed to work after uploading only HTML to Netlify?

Worked answer

No. The selected live path uses a different managed receiver. Its form detection, storage, spam handling and notifications follow Netlify documentation; the local server's custom validation and duplicate behavior are not inherited. Either accept and test the bounded managed behavior or scope a separately supported custom receiver.

Sources: S19, S22, S23, S24, S25, S26. Full links and limits are in the source register.

L15 Prove notifications recover a release and maintain the site

Handoff,host conditionally and operate. Study and exercise allowance 60 minutes.

Produce: A live acceptance ledger, rollback rehearsal and owner maintenance agreement

Use forms F13, F14.

Configure one real notification destination

Under actual client authorization, open Forms → Submission notifications and configure the agreed form's email notification to the approved owner recipient. Netlify requires an active team-member address or an outside address that has confirmed receipt where that rollout is available. Its documentation says an unconfirmed outside recipient receives no notifications; some teams may instead require a team-member address [S20]. Record the state you actually see. Do not invent a menu option, bypass confirmation or add yourself permanently as the client's only recipient. This is one owner notification, not a marketing sequence or automated reply to visitors.

Submit a clearly marked, client-approved test through the real public domain. Record the visible response, form name, timestamp and exact test marker. In Forms, select the active form and inspect the matching verified submission; also inspect spam when the test does not appear as expected. The documented submissions screen can export verified records as CSV and permanently delete selected submissions [S21]. A saved record and the notification email are separate evidence. Have the owner find the matching email, including its actual folder, then open the saved record and explain the manual follow-up. Never label a confirmation email sent to the recipient as proof that a later inquiry notification arrived.

Test missing delivery and rollback

Use a temporary authorized test recipient or an agreed controlled mailbox rule to rehearse notification not being observed without disrupting the real business inbox. Preserve the stored test request, mark notification unresolved, and use the owner's documented dashboard-review fallback. Do not create repeated visitor submissions to compensate blindly. Reconcile stored inquiries against reviewed tasks at an agreed interval; in the fictional example, Morgan checks the dashboard at 9 a.m. each business day and escalates any missing notification. This interval is an invented operating choice, not a service guarantee. A provider outage needs a separately available, approved contact alternative.

Before a later update, retain the known-good deploy ID and approved source. Netlify's documented rollback uses Publish Deploy on a previous successful deploy that is still available [S27]. In an authorized rehearsal, publish the approved prior version, inspect the live domain and run a fresh inquiry test, then restore the agreed current version. If Git auto-publishing exists, account for later builds overwriting the rollback; use documented locking when appropriate. Restoring page files does not undo DNS changes, notification settings, deleted submissions or outside email copies. Record those recovery procedures separately. A local file backup alone does not prove hosted recovery.

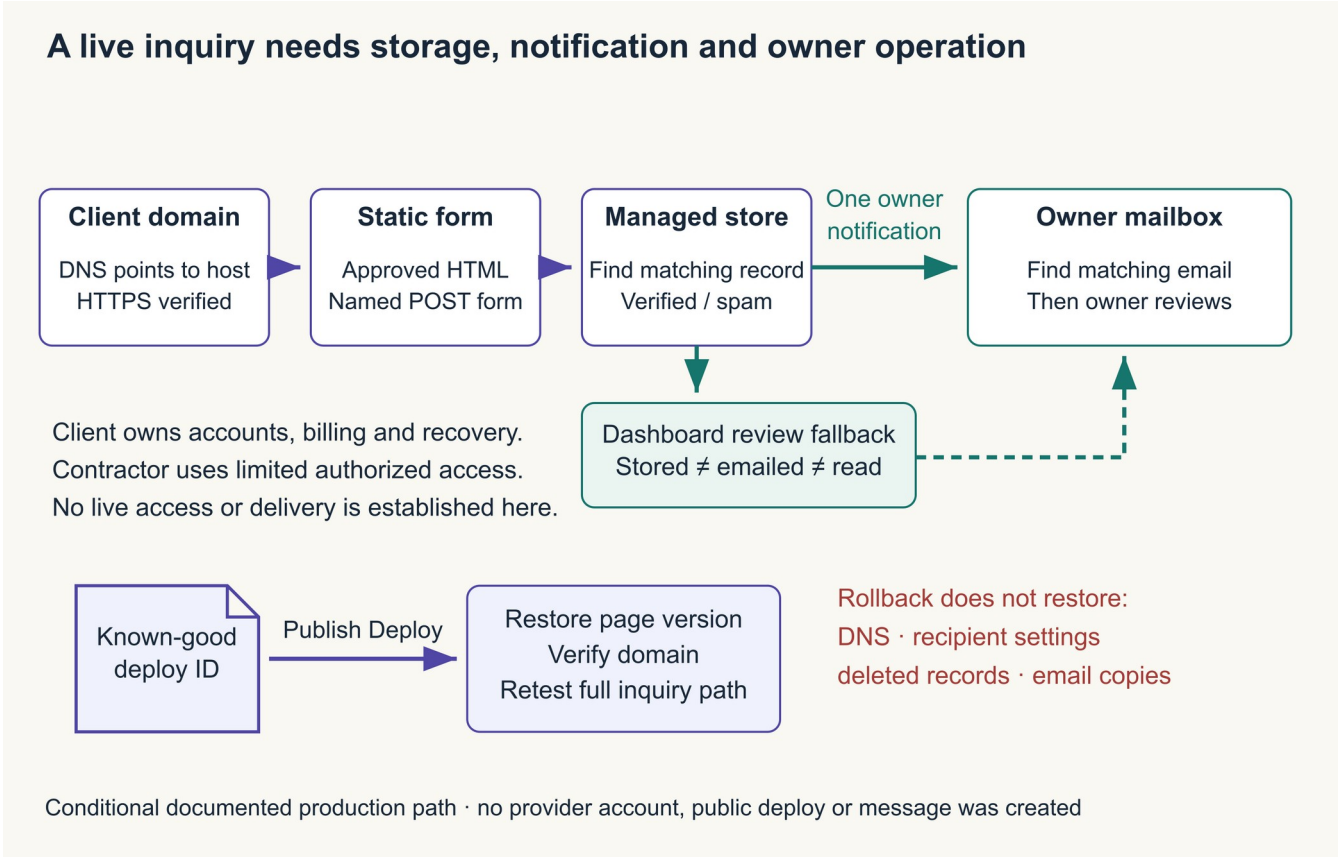
Operate and price maintenance separately

For acceptance, record at least a successful public-domain request, a malformed browser submission, a spam-path review, an unobserved-notification exercise, a duplicate-looking request review, private-file access checks and the rollback/retest. Do not carry the local lab's expected HTTP codes into this hosted ledger. Record actual behavior and compare it with the hosted scope. If a requirement is unmet, correct or narrow the promise before acceptance. Have the owner perform review, export and recovery tasks with their own account; then remove temporary contractor access and verify their continued control.

The fictional maintenance option is separate from the base project: $90 per month for one scheduled 60-minute content/link/form/notification review and a written report, with any unused allowance expiring under explicitly reviewed terms. At the invented $30 owner value, 60 minutes cost $30; a $6 overhead allocation and one fictional $3 fee leave $51 before any reserve. This is a pricing exercise, not a subscription sale or a current provider quote. It includes no round-the-clock monitoring, unlimited revisions, new features or paid provider renewals. Client-owned hosting/domain/mailbox subscriptions remain direct client costs. Set renewal responsibility, retention/deletion tasks, incident contacts and what

happens when either party ends maintenance. A verified process and clear ownership make the small automation supportable.

1. Record dashboard saving, mailbox arrival and owner review as separate observed events.
2. Rehearse recovery with approval and rerun the live inquiry route.
3. Confirm the client owns operation, renewals and recovery before removing your access.



Original conditional hosted architecture based on Netlify official docs S19–S27. No actual account, deploy, DNS, TLS or mailbox evidence is established.

Practice

Rolling back restores the old contact page. Does it restore deleted form records or the previous notification recipient?

Worked answer

No such restoration is established by a page-deploy rollback. Treat stored data, notification settings, DNS and mailbox copies as separate systems with their own recovery evidence. Retest the complete inquiry route after the rollback.

Sources: S20, S21, S24, S27. Full links and limits are in the source register.

L16 Close the job and decide whether to offer it

Handoff,host conditionally and operate. Study and exercise allowance 40 minutes.

Produce: A measured closeout, follow-up and real readiness decision

Use forms F01, F09, F12, F14.

Use the promised follow up

The fictional full project CP-026 includes 60 used follow-up minutes: a 30-minute review with Morgan on December 11 and 30 minutes to record results and close the evidence pack. Recheck the included full scope: a permitted ordinary submission reaches the managed form record; Morgan can sign in through Morgan's own approved access, find the exact request and observe the one matching notification in the approved mailbox. If the record exists but the notification is missing, record those two outcomes separately, use the agreed dashboard-review fallback and escalation route, correct the included notification function and retest before treating the defect as resolved. The fallback does not turn a failed email test into a pass. Hosting and that one notification are already included in CP-026; fixing an included regression is not a new service charge. Only new scope or separately agreed continuing support requires a new estimate. A prototype-only follow-up instead checks the local starter and private file, with zero real emails; do not apply that limited check to the full project. The separately counted 30-minute backup/reset/closure stage still belongs in the whole owner cycle.

For the full hosted offer, retain separate evidence of receiving, notification delivery, owner review and recovery. The selected provider's submissions area can be reviewed independently of email, and notification recipients must satisfy the provider's current team-membership or confirmation rules. Confirm those conditions in the client's actual authorized environment before claiming delivery [S19–S21]. This course supplies only fictional hosted acceptance assumptions; no live account, recipient confirmation or real message is activated by the local exercise.

Reconcile actual receipts, fees, cash expenses and invoice status against the financial worksheet. Preserve the original baseline and compare stage-level actual minutes. A late client content delivery may move the calendar without adding labor; an unplanned rewrite adds labor even if the final page count stays three. Record both separately. Keep sufficient business records and supporting evidence for your real reporting obligations [S17]. Your own income tax, self-employment tax and insurance costs are not solved by a service-price calculator.

Evaluate the business with evidence

Review the two-week campaign and delivery together. The fictional paid channel acquired one booking for $60 cash plus $120 owner value. The full base project's $548.70 management profit before reserve leaves $368.70 after that $180 acquisition cost, or $278.70 after the $90 reserve. The correction branch leaves $408.70 before reserve, $228.70 after acquisition and $138.70 after reserve. A full refund instead produces a substantial loss. Do not report the $1,800 receipt as earnings or use these invented figures to promise income.

Ask which part of the service prospects valued, which accepted tasks you can repeat and where estimates were wrong. If prospects mainly want live hosting and bookings, a prototype-only offer may have weak demand. Either develop the additional supported capability with permitted practice and a new cost model or choose a better-fitting service. If the owner-review workflow works but content approval consumes twice the time, improve discovery, dependencies and revision terms before buying more advertising. Demand evidence, delivery evidence and economic evidence all matter.

Make a real readiness decision

Before paid prototype work, independently reproduce the local build, explain the request path, pass actual success/failure tests and complete the scope/cost/handoff records. Before selling the full live scope, also demonstrate the selected provider's receiving, mailbox delivery, owner operation, DNS/TLS and recovery under real permission. Confirm business requirements, tax treatment, appropriate terms

and insurance. Your reviewer should be able to challenge a saved count, ask for a rollback and inspect the recipient confirmation without relying on fictional answers. Lesson completion alone establishes none of that practical readiness.

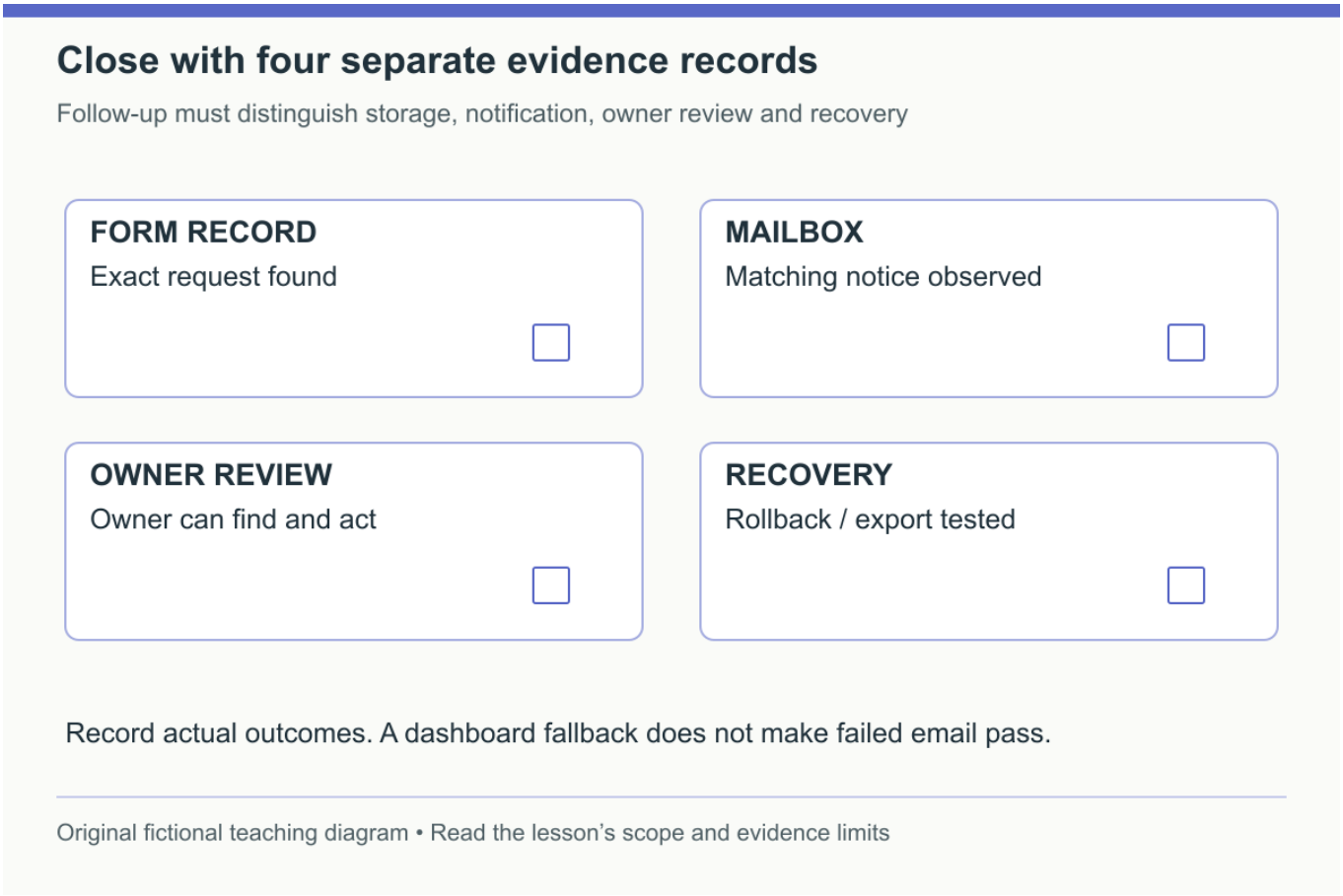
The optional foundations summary is simple: sell a defined result, document permission, keep truthful records and reserve money for obligations before treating cash as available. Price follows the whole work cycle, including sales, correction and follow-up. Keep personal and business transactions traceable and ask qualified advisers about legal or tax questions that depend on your actual facts.

For this service specifically, never let attractive pages outrun the receiving system. State whether a request was stored, whether a notification was sent and whether the owner reviewed it. The next practical action is a permitted local rehearsal using only fictional data, followed by an independent person finding the saved record. If that cannot be repeated, stay in practice and resolve the cause before taking responsibility for a client's inquiries.

1. Close the actual timeline and reconcile money without rewriting the baseline.

2. Compare contribution after the matching campaign cost.

3. Mark each real readiness criterion demonstrated, unresolved or out of scope.



Receipt storage, mailbox notification, owner operation and recovery need separate actual evidence. A fallback review route does not turn an email failure into success.

Practice

You can explain the fictional ledger but have not run the supplied system. Are you ready to sell this technical result?

Worked answer

No. Explanation is useful study evidence, but practical readiness requires your own permitted build, actual tests, owner-operation rehearsal and complete scope/cost record. Continue with the local rehearsal; no public deployment, account purchase or real messages are required to practice the bounded task.

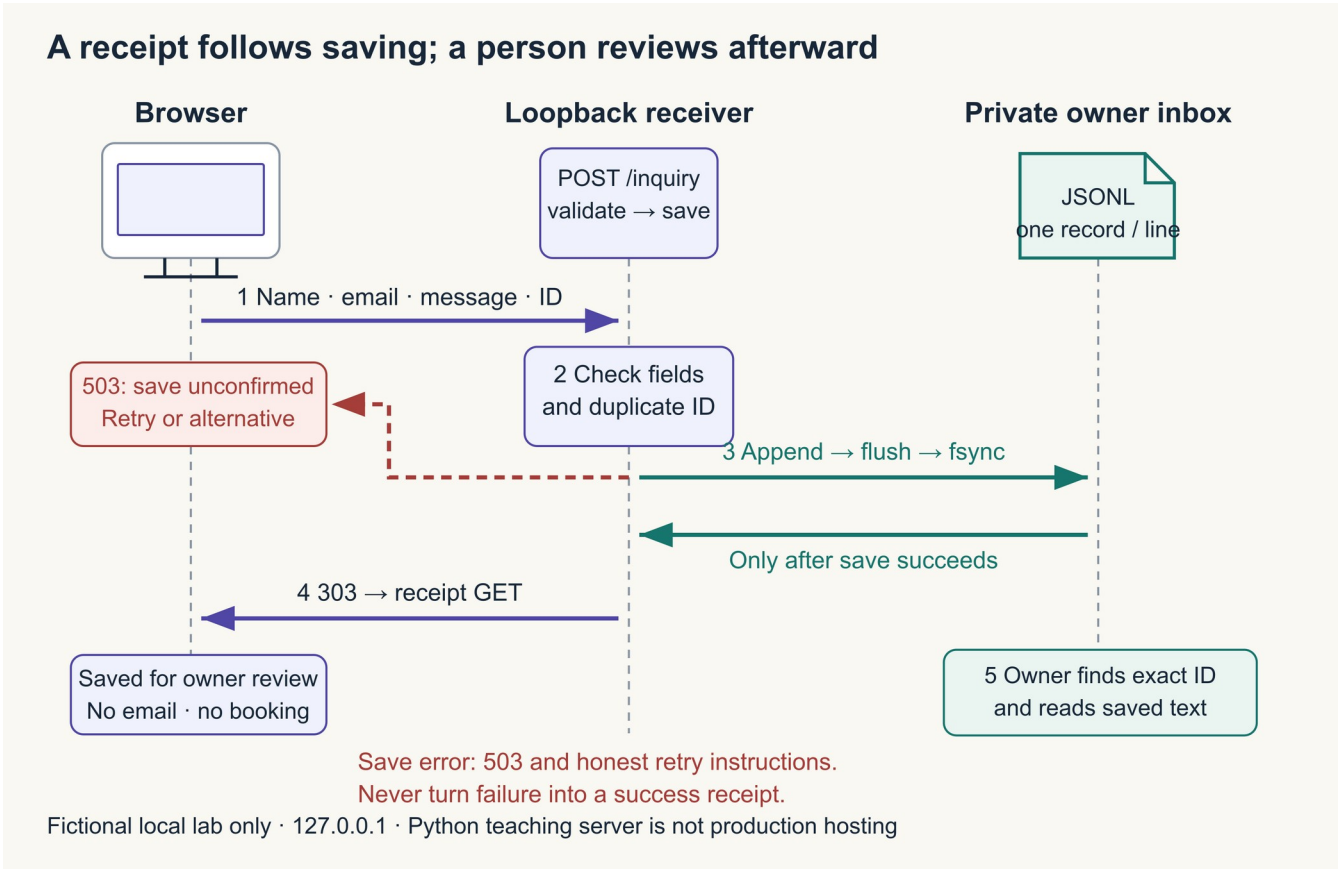
Sources: S01, S12, S14, S17, S19, S20, S21. Full links and limits are in the source register.

One complete readiness decision

- Actual permission-based demand conversations and a stop-before-purchase decision are recorded.
- Own permitted local rehearsal has repeatable navigation,validation,saving,failure,retry,escaping and private-file evidence.
- Independent owner role finds a saved request and restores the local project without hidden steps.
- A measured whole-cycle estimate,actual tax/business checks and reviewed scope/terms exist before paid work.
- Full live service additionally requires client ownership/access,appropriate provider terms,explicit publication/notification approval,DNS/TLS,managed storage,mailbox arrival,owner review and rollback evidence.
- No local pass or fictional client approval is treated as actual production readiness.
- Client renewal,recovery,retention,incident and maintenance responsibilities are documented and temporary access is removed.

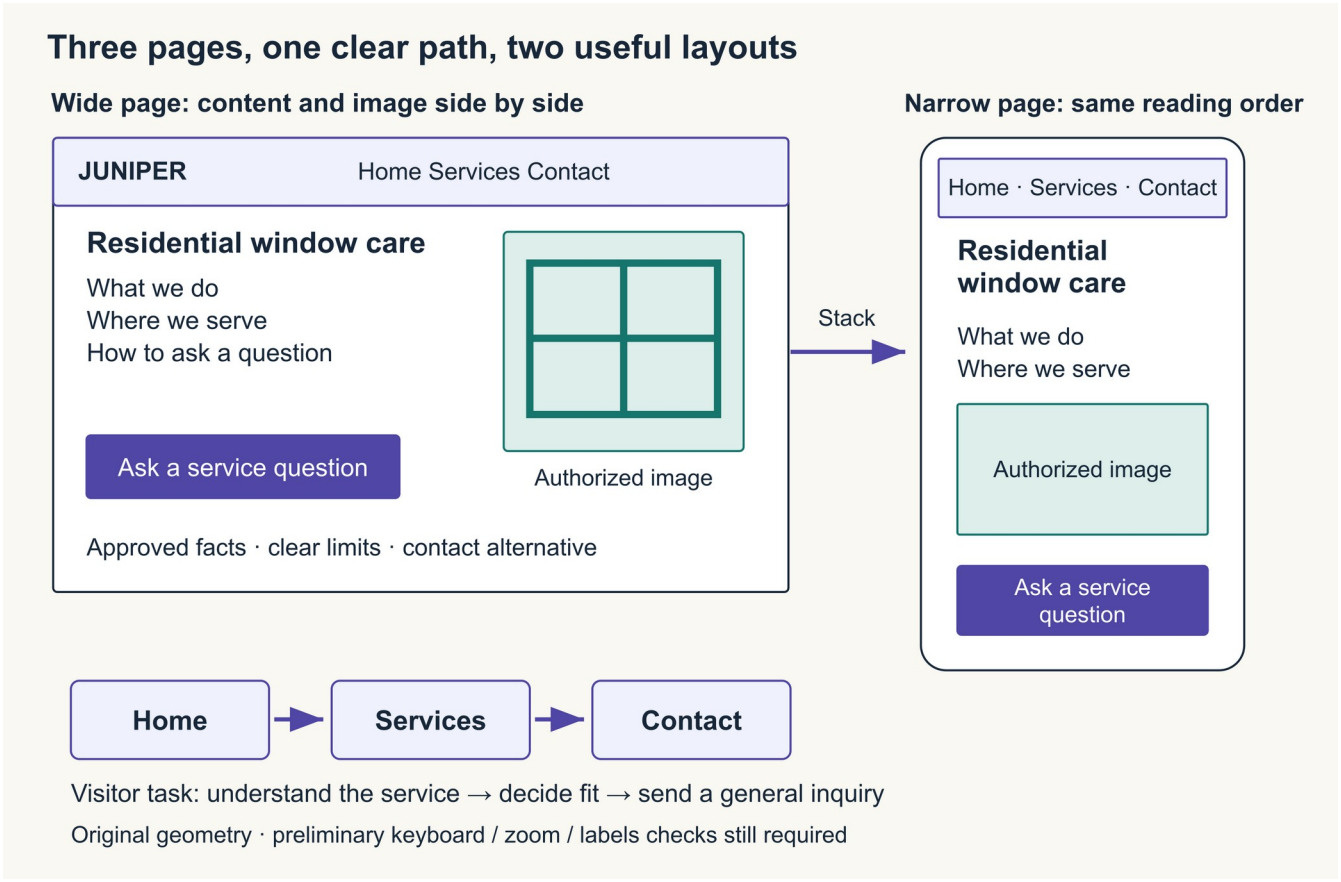
Original visual reference

Local inquiry browser receiver and owner review



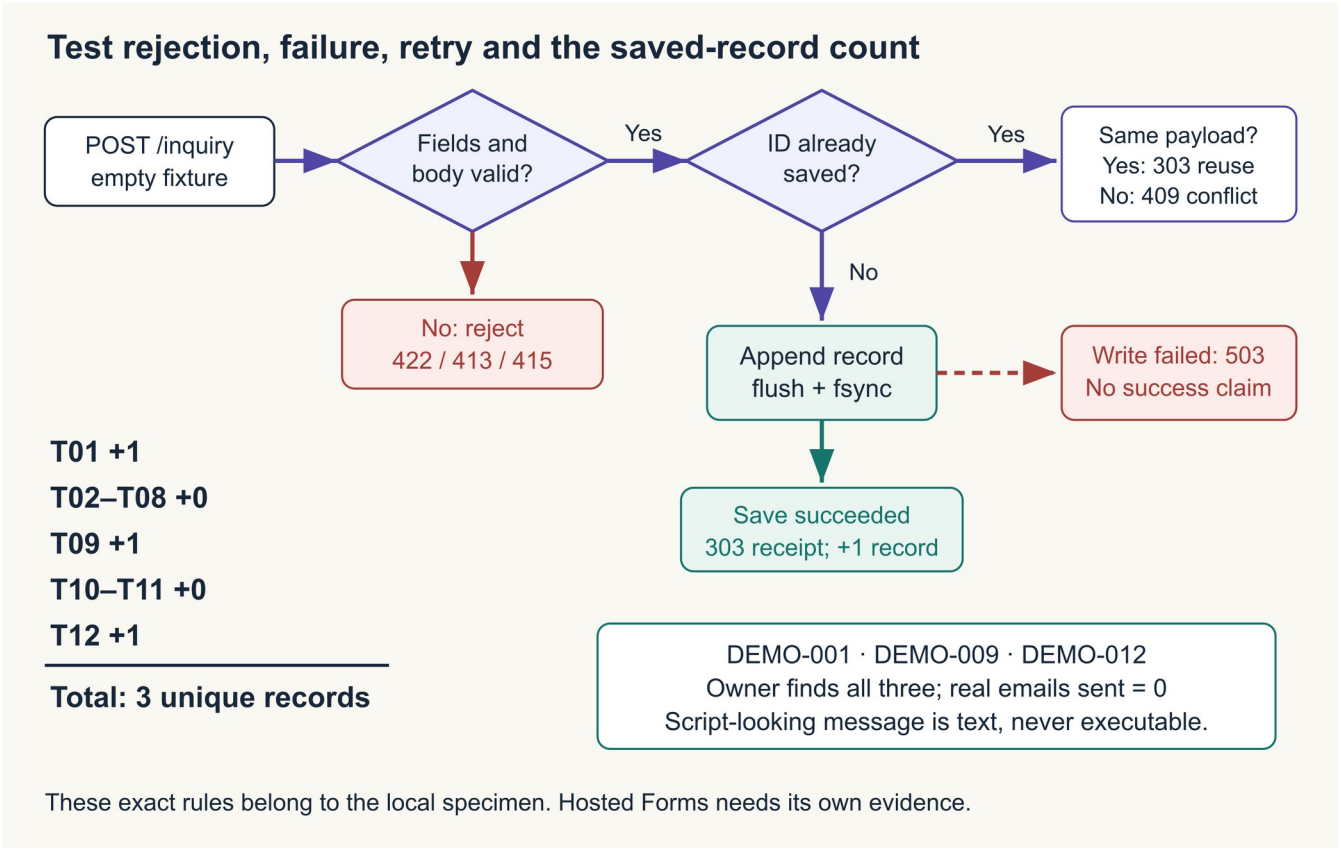
Original local system diagram. HTML/Python components supported by S01–S09 and303 by S16. Local save is not email delivery or public readiness.

Three pages and one reading route



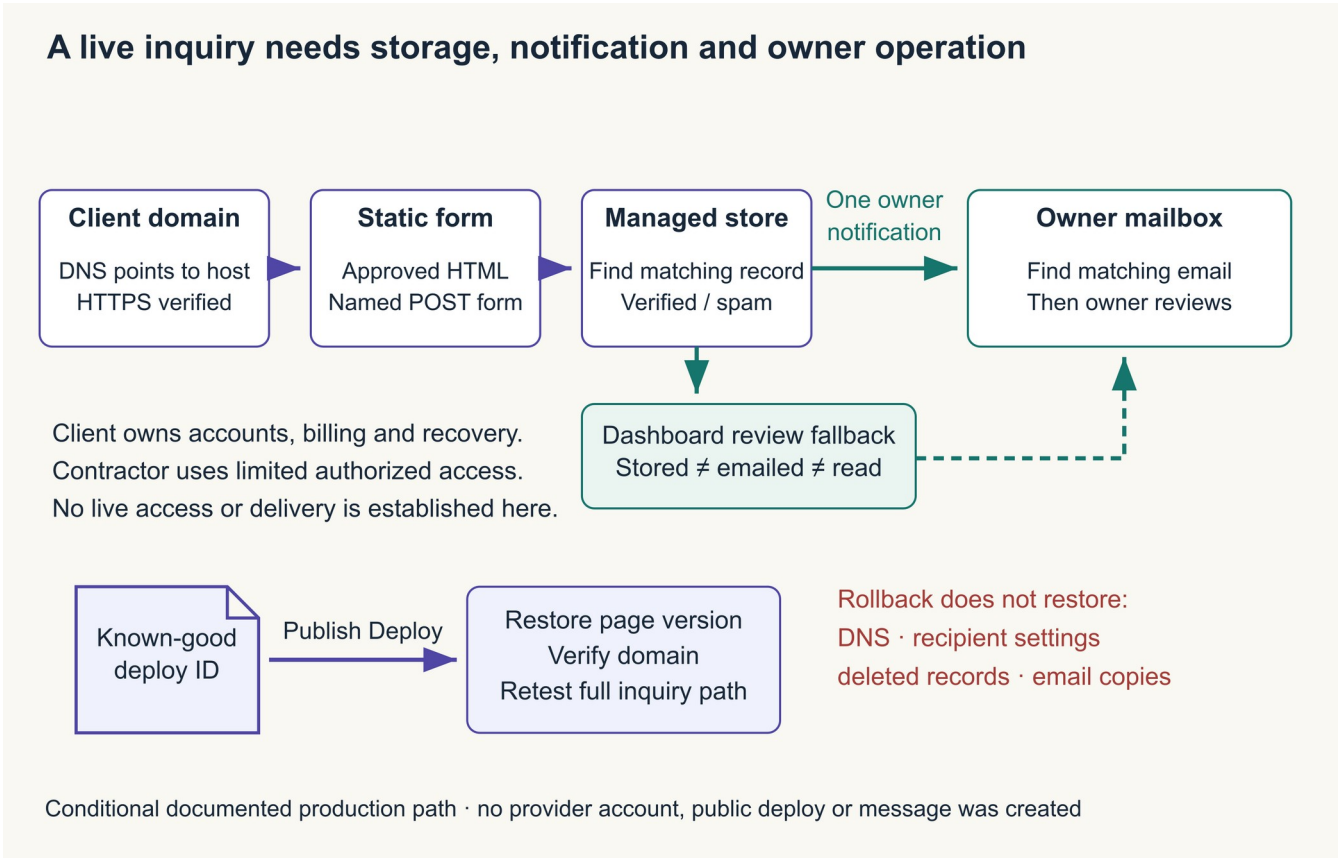
Original page geometry for the fictional business. W3C preliminary review guidance S11; design choices do not establish accessibility conformance.

Request decisions and the three record answer



Original worked local request logic. Exact statuses and limits are specimen rules; managed hosted Forms does not inherit them. Sources S01, S04, S06–S09, S16.

Conditional hosted inquiry and recovery path



Original conditional hosted architecture based on Netlify official docs S19–S27. No actual account, deploy, DNS, TLS or mailbox evidence is established.

Source register

These sources support the identified product, platform and jurisdiction only. Recheck changing rules and exact model instructions before work. Case evidence and illustrations are original fictional teaching material, not field validation.

S01 http server HTTP servers

Python Software Foundation. Checked 2026-11-21. Local HTTP handlers and explicit warning against production use.

Local teaching component only; no public hosting/security certification.

https://docs.python.org/3.13/library/http.server.html

S02 HTML Standard Forms

WHATWG. Checked 2026-11-21. Form interface, named controls, method/action and separate server processing.

Normative HTML behavior does not implement the original server or guarantee delivery.

https://html.spec.whatwg.org/multipage/forms.html

S03 Labeling Controls

W3C Web Accessibility Initiative. Checked 2026-11-21. Visible associated form labels and understandable control purpose.

Technique guidance, not comprehensive accessibility evaluation.

https://www.w3.org/WAI/tutorials/forms/labels/

S04 Validating Input

W3C Web Accessibility Initiative. Checked 2026-11-21. Browser assistance and need for separate server validation.

Original specimen bounds/email syntax choices are not universal or production certification.

https://www.w3.org/WAI/tutorials/forms/validation/

S05 User Notification

W3C Web Accessibility Initiative. Checked 2026-11-21. Clear success and error feedback and correction instructions.

A message is not proof that storage or email occurred.

https://www.w3.org/WAI/tutorials/forms/notifications/

S06 urllib parse parse_qs

Python Software Foundation. Checked 2026-11-21. Form URL decoding, keep_blank_values and max_num_fields.

Parser alone does not establish application validation or security.

https://docs.python.org/3.13/library/urllib.parse.html#urllib.parse.parse_qs

S07 json JSON encoder and decoder

Python Software Foundation. Checked 2026-11-21. JSON encoding and warning that repeated dumps do not frame one JSON document.

JSONL line format is an original application choice; not a transactional database.

https://docs.python.org/3.13/library/json.html

S08 html escape

Python Software Foundation. Checked 2026-11-21. Escaping HTML-significant characters when displaying text.

Output-context technique, not complete application security.

https://docs.python.org/3.13/library/html.html#html.escape

S09 os fsync

Python Software Foundation. Checked 2026-11-21. Flush file object before fsync to write buffers to disk; OSError possibility.

No guarantee against every storage/hardware failure or distributed concurrency issue.

https://docs.python.org/3.13/library/os.html#os.fsync

S10 Protecting Personal Information A Guide for Business

Federal Trade Commission. Checked 2026-11-21. Data inventory, minimization, access, retention and incident planning.

General U.S.guidance; actual duties depend on data, business and jurisdiction.

https://www.ftc.gov/business-guidance/resources/protecting-personal-information-guide-business

S11 Easy Checks A First Review of Web Accessibility

W3C Web Accessibility Initiative. Checked 2026-11-21. Preliminary page/title/image/keyboard/zoom/forms checks and explicit limits.

Passing quick checks is not comprehensive conformance or legal certification.

https://www.w3.org/WAI/test-evaluate/preliminary/

S12 Plan your business

U.S.Small Business Administration. Checked 2026-11-21. Research customer need and calculate startup costs.

Does not establish local demand, market rates or earnings.

https://www.sba.gov/counseling/plan-your-business/

S13 CAN SPAM Act A Compliance Guide for Business

Federal Trade Commission. Checked 2026-11-21. Commercial email rules include B2B; accurate headers/subject, ad identification, postal address and opt-out.

Unsent conversation examples are not complete compliant campaign templates; actual channel/legal duties need review.

https://www.ftc.gov/business-guidance/resources/can-spam-act-compliance-guide-business

S14 Launch your business

U.S.Small Business Administration. Checked 2026-11-21. Registration, licenses/permits, banking and insurance are separate launch decisions.

Requirements are fact/location specific; not professional authorization.

https://www.sba.gov/counseling/launch-your-business/

S15 SEO Starter Guide

Google Search Central. Checked 2026-11-21. Useful descriptive content and page organization; no indexing/ranking guarantee.

No promise of traffic, leads or sales; no accounts or search services used.

https://developers.google.com/search/docs/fundamentals/seo-starter-guide

S16 RFC9110 HTTP Semantics

IETF/RFC Editor. Checked 2026-11-21. 303 directs retrieval of a separate resource after submission.

Protocol semantics alone do not prove persistence or safe retries.

https://www.rfc-editor.org/rfc/rfc9110.html#name-303-see-other

S17 Recordkeeping

Internal Revenue Service. Checked 2026-11-21. Records clearly show income/expenses and preserve supporting evidence.

Management model is not a tax return or individualized tax advice.

https://www.irs.gov/businesses/small-businesses-self-employed/recordkeeping

S18 Sales Tax in Oregon

Oregon Department of Revenue. Checked 2026-11-21. Oregon has no general sales or use/transaction tax.

The case assumes Oregon provider, customer and delivery with no other-state nexus or special tax. Income, local and other-state obligations are not resolved.

https://www.oregon.gov/dor/programs/businesses/Pages/sales-tax.aspx

S19 Forms setup

Netlify. Checked 2026-11-21. Enable form detection; named static HTML POST form with data-netlify; deploy-time parsing and hidden form-name; custom success-page action.

Conditional read-only teaching. No actual account, detection or deployment. The managed receiver is different from the local Python lab.

https://docs.netlify.com/manage/forms/setup/

S20 Form notifications

Netlify. Checked 2026-11-21. Forms → Submission notifications. Email recipients must be active team members or confirmed outside addresses where available; pending confirmation receives no notifications.

Rollout varies. Configuration does not prove mailbox arrival or human reading.

https://docs.netlify.com/manage/forms/notifications/

S21 Form submissions

Netlify. Checked 2026-11-21. Active-form verified and spam views, verified CSV export and permanent deletion of selected submissions.

Exports and mailbox messages are additional copies. The UI does not establish a complete backup or lawful retention policy.

https://docs.netlify.com/manage/forms/submissions/

S22 Create deploys

Netlify. Checked 2026-11-21. Manual static-folder deployment within the client team and updating through the project's deploy area.

An upload can publish. Actual authorization is required. No anonymous or account-bound upload was performed.

https://docs.netlify.com/deploy/create-deploys/

S23 Configure external DNS for a custom domain

Netlify. Checked 2026-11-21. Assign the owned domain; fetch project-specific Pending DNS verification values; subdomain CNAME and distinct apex configuration.

Fetch exact values from the actual account and DNS provider. No DNS record was changed.

https://docs.netlify.com/manage/domains/configure-domains/configure-external-dns/

S24 Spam filters

Netlify. Checked 2026-11-21. Verified/spam separation and a documented optional honeypot field.

Filtering can suppress legitimate tests. Not security certification; no CAPTCHA provider was engaged.

https://docs.netlify.com/manage/forms/spam-filters/

S25 Forms usage and billing

Netlify. Checked 2026-11-21. Forms usage and billing rules depend on plan generation.

No actual plan or rate quote. Confirm the total hosting, domain and mailbox costs before a purchase.

https://docs.netlify.com/manage/forms/usage-and-billing/

S26 HTTPS SSL

Netlify. Checked 2026-11-21. Managed TLS for Netlify-hosted content and custom domains.

Actual DNS and certificate readiness must be observed. HTTPS alone is not comprehensive privacy or security.

https://docs.netlify.com/manage/domains/secure-domains-with-https/https-ssl/

S27 Manage deploys Rollbacks

Netlify. Checked 2026-11-21. Publish Deploy on an available prior successful deploy; Git auto-publishing may overwrite rollback; documented deploy locking.

Page rollback does not restore DNS, deleted submissions, notification settings or email copies.

https://docs.netlify.com/deploy/manage-deploys/manage-deploys-overview/