COMM2020 Team Project
Project Specification (Project 4)
Single-Scan Traceability Quest

Usage of AI Tools: AI-Minimal

These tasks have been set to assess your problem-solving abilities, and so you may not use AI for generating code or autocomplete while completing the project.

The University of Exeter is committed to the ethical and responsible use of Generative AI (GenAI) tools in teaching and learning, in line with our academic integrity policies where the direct copying of AI-generated content is included under plagiarism, misrepresentation and contract cheating under definitions and offences in TQA Manual Chapter 12.3.

This assessment falls under the category of AI-Minimal in the University's Guidance on use of Gen AI in Assessment.

This means: You may use AI tools for checking spelling and grammar mistakes only, with no other impact on the structure or content of the assessment. This is because using GenAI tools outside of these uses prevents fair assessment of your ability to achieve module learning outcomes.

When writing your assessment, you must never use AI tools:

- For uses other than checking your spelling and grammar.
- To translate more than a word or short phrase into English.
- To upload sensitive or identifying material to an AI tool.
- To present material that has been generated by AI as your own work or the work of someone else.

When submitting your assessment, you must:

- Check the box during the submission process that confirms you have adhered to the university's academic conduct policy and the expectations on use of GenAI in your assessment brief.

NOTICE: Your coding activity must all be completed on the GitHub repository. This logs your activity as you code, and the logs will be checked for submissions that have made use of AI-generated code.

Guidelines

- This assessment is AI-Minimal: spelling/grammar checks only; no AI code generation or autocomplete.
- Your application will be demonstrated online. Provide a deployed URL for the live demo and a clear local run guide for markers.
- Your system must remain usable with a seeded dataset even if external services are unavailable.

1 Requirements

Build a web application that allows a consumer to scan/enter a product identifier and instantly view a product's traceability story. The story must show where a product's components/ingredients came from, how the product was assembled and transported, and which claims are supported by evidence.

1.1 The practical problem Sourcr solves (required framing)

Existing solutions (labels, certificates, brand websites, and emerging Digital Product Passport initiatives) provide incomplete and fragmented traceability information, often focused only on final assembly and not on all ingredients/components. Consumers have no single, simple tool that connects sources and presents the supply chain in understandable language.

Your system must explicitly support these two example scenarios (and you may add others):

- Food traceability scenario: 'EU' label is too broad. A consumer wants to know which countries contributed to an olive oil blend, and which portion came from each country.
- Luxury product scenario: a bag is manufactured mostly outside the EU but finished in an EU country. The consumer wants a clear breakdown (e.g., 90% produced in Country A; finishing and label in Country B).

1.2 The game concept: Traceability Quest (required)

Consumers do not always engage with long technical passports. To encourage exploration and comprehension, your application must include a lightweight 'Traceability Quest' game mode: users complete short missions that require reading and interpreting traceability information.

- Missions must be grounded in real passport data (e.g., 'Which country contributed the largest share of ingredient X?').
- Missions must include at least 3 difficulty tiers (basic, intermediate, advanced).
- Users earn points and badges for completing missions; the system must show progress over time.

2 Users and user profiles

Your application must support three user types:

2.1 Consumers (default users)

- Scan/enter a product identifier and view an understandable traceability story.
- Switch between ‘simple view’ and ‘evidence view’ (show confidence and sources).
- Play missions linked to the product and receive feedback on answers.

2.2 Verifiers / Product Editors (trusted users)

- Create and update product records and their supply chain stages.
- Upload/attach evidence (certificate, audit report, supplier statement, lab test).
- Set and justify confidence levels for claims (e.g., verified, partially verified, unverified).

2.3 Maintainers (developers)

- Maintain deployments, documentation, data pipelines, and testing.

3 Features

3.1 Core traceability journey (required)

1. Single scan/lookup: user enters or scans a product ID to reach a product page.
2. Traceability timeline: show stages from raw materials → processing → assembly → transport → retail.
3. Origin breakdown: for at least 2 product categories, show percentage breakdown across countries/regions for key inputs (e.g., olive oil blend; bag components).
4. Claim cards: show key claims (e.g., organic, recycled content, fair labour). Each claim must display a confidence label and allow drill-down to evidence.
5. Evidence view: list evidence items with issuer, date, and short summary; allow the user to open/download evidence where appropriate.
6. Comparison: compare two products on origin breakdown and claim confidence (side-by-side).

3.2 Quest gameplay (required)

- At least 18 missions across three tiers (6 per tier).
- At least 10 missions must be automatically graded (multiple choice / numeric / match).
- At least 4 missions must involve interpreting the origin breakdown (percentages across countries).
- After each mission, provide feedback explaining the correct answer with a link to the relevant passport section.

- A leaderboard is optional; however, personal progress tracking is mandatory.

3.3 Verification and moderation workflow (required)

- Verifiers must be able to create/update a product and its stages through an admin interface.
- Every claim must be linked to at least one evidence item OR explicitly marked as ‘unverified’ with an explanation.
- The system must log changes to products/claims (who changed what and when).
- The system must include a ‘report an issue’ mechanism for consumers (e.g., ‘claim seems false’, ‘evidence missing’).

3.4 Technical requirements (required)

- Authentication: role-based access control (consumer/verifier).
- Security: secrets via environment variables; no secrets in repository or ELE ZIP; avoid storing sensitive personal data.
- Accessibility: keyboard navigation for core screens, readable contrast, and mobile-friendly layouts.
- Professional practice evidence: use GitHub commits/issues/PRs; the repository URL must be included in 0_admin/submission.txt.
- Testing (mandatory): you must submit BOTH (A) an automated test suite that can be executed by the markers, AND (B) a manual end-to-end test plan with results evidence.
- Automated tests (A) must include at least 15 automated tests covering: product lookup, timeline rendering, origin breakdown calculations, evidence linking, quest grading, and verifier edit workflow.
- Manual tests (B) must include at least 8 end-to-end scenarios (happy path + failure cases). Include expected results and screenshots/logs of completed runs in 4_technical/testing_evidence.pdf.
- Your deployment_guide.pdf must include a ‘How to run tests’ section with the exact commands/steps.

4 Data and seeded dataset

You must include a seeded dataset so the system works during marking without requiring external data. You may enrich with open datasets/APIs, but the core demo must function using your seeded dataset.

4.1 Minimum dataset contents (required)

- At least 200 products across at least 3 categories (must include at least one food product and one luxury product category).
- At least 200 traceability timelines (one per product) with at least 5 stages each (raw → processing → assembly → transport → retail).

- At least 400 evidence items linked to claims or stages (files or records).
- At least 18 missions linked to products (6 per tier), with answer keys for auto-graded missions.
- At least 30 pre-seeded ‘issues’ (reports) to demonstrate the issue-review workflow.

4.2 Suggested entities (example)

Entity	Key fields (minimum)
Product	product_id, name, category, brand, description, image(optional)
Stage	stage_id, product_id, stage_type, location(country/region), start/end dates(optional), description
InputShare	input_id, product_id, input_name, country, percentage, notes
Claim	claim_id, product_id, claim_type, claim_text, confidence_label, rationale
Evidence	evidence_id, linked_to(stage/claim), type, issuer, date, summary, file_reference
QuestMission	mission_id, product_id, tier, question, answer_key, grading_type, explanation_link
IssueReport	issue_id, product_id, reported_by(anon), type, description, status, resolution_note
ChangeLog	log_id, entity_type, entity_id, changed_by, timestamp, change_summary

5 Measures of success

You must define and report success measures with evidence. Suitable measures include:

- Comprehension: can users correctly answer missions after reading the passport (accuracy by tier).
- Transparency: do users find the evidence view understandable (short survey or task-based observations).
- Data integrity: validation checks catch inconsistent origin percentages or missing evidence links.
- Usability: time to find key origin information and interpret confidence labels.

6 Process and deliverables

You must submit all group deliverables on ELE as a single ZIP. GitHub is used to evidence professional practice and must be referenced in submission.txt.

Intended Learning Outcomes assessed by the coursework

Coursework 1 and Coursework 2 assess all module Intended Learning Outcomes (ILOs):

- ILO1 – Function effectively as a member of a team.

- ILO2 – Apply an integrated or systems approach to the solution of complex problems.
- ILO3 – Apply knowledge of domain context, project and change management, and relevant legal matters including intellectual property rights.
- ILO4 – Select and apply appropriate materials, technologies, and processes, and recognise their limitations.
- ILO5 – Plan self-learning and development to support the activity of the wider team.
- ILO6 – Support an inclusive approach to teamwork and problem solving, recognising the responsibilities, benefits and importance of supporting equality, diversity, and inclusion.
- ILO7 – Evaluate the environmental and societal impact of solutions to complex problems and minimise adverse impacts.

Evidence is expected across reports, implementation, evaluation, ethical/legal materials, and teamwork/process artefacts in both sprints.

6.1 Sprint 1 (Coursework 1) – Prototype v1, report and demonstration (week 5)

Submission to ELE is a single ZIP file named GroupX_CW1.zip.

Sprint 1 deliverable expectations

- Prototype v0.1.0: product lookup → traceability timeline → at least one claim with evidence → complete one mission and receive feedback → progress updates.
- Verifier workflow: demonstrate creating/editing one product stage and attaching evidence to a claim.
- Testing evidence (CW1): include at least 5 automated tests running successfully and document how to run them in deployment_guide.pdf; include one manual end-to-end test run with evidence in testing_evidence.pdf.
- Prototype report must include: executive summary, prioritised requirements, architecture v1, dataset design and provenance, initial evaluation evidence, and a sprint plan for CW2.
- Ethical/legal considerations must cover privacy/GDPR (even if no personal data), misinformation/consumer harm risk, accessibility, security controls, and IP/licensing implications.
- Software/data inventory must list all dependencies and datasets (licence, provenance, cost model, versions).

Live demo and presentation (10 minutes) – what to cover (CW1)

7. 30 seconds: what traceability problem you solve and what ‘single scan’ gives the user.
8. 6–7 minutes: demo the vertical slice including evidence view and one mission.
9. 1–2 minutes: show verifier workflow and change log/issue reporting.
10. Final minute: what will be completed in Sprint 2 and your top risks.

6.2 Sprint 2 (Coursework 2) – Final prototype, client handover and presentation (week 11)

Submission to ELE is a single ZIP file named GroupX_CW2.zip.

Individual reflection (submitted separately by each student on ELE): reflection.pdf (800–1,000 words; includes evidence links to commits/issues/PRs; and an AI-Minimal compliance statement).

Sprint 2 deliverable expectations

- Final prototype v1.0.0: stable end-to-end traceability across categories; complete evidence linking; robust origin breakdown and comparison features.
- Quest mode: all 18 missions implemented across tiers with clear feedback and progress tracking; at least one mission-set should be product-category specific.
- Verification/moderation: verifier workflow complete; issue review workflow demonstrated with statuses and resolution notes; change log working.
- Client handover pack: clear deployment, operations, and maintenance guidance so another team could run and extend the system.
- Testing evidence (CW2): meet the full testing requirement (15+ automated tests + 8+ manual scenarios) and include clear pass/fail evidence in testing_evidence.pdf. Marks will be reduced if tests cannot be run by markers.
- Final evaluation: report success measures with method and limitations; include an explicit discussion of consumer harm risks and mitigations (misinformation, overconfidence, misleading visualisations).
- Updated ethical/legal and licensing materials consistent with the final system and all dependencies.

Live demo and presentation (10 minutes) – what to cover (CW2)

11. 1 minute: recap problem and what is now delivered.
12. 6–7 minutes: demo traceability across at least two categories, including comparison and multiple missions.
13. 1–2 minutes: show handover pack and how a maintainer would add a new product category and mission set.
14. Final minute: evaluation highlights, limitations, and next steps.

Individual reflection (Coursework 2 – individual deliverable)

Each student must submit an individual reflection on ELE (not in the group repository). Suggested length: 800–1,000 words. This reflection is used to evidence individual learning and contribution and may be used to resolve contribution disputes.

15. Your role and contributions: describe what you owned (features, testing, documentation, deployment). Reference concrete evidence (PR links, issue IDs, commits).
16. What you learned: at least three specific technical or professional learning points linked to module outcomes (e.g., requirements negotiation, risk management, testing, deployment).
17. Challenges and how you addressed them: one technical challenge and one teamwork/process challenge; what changed as a result.
18. Responsible computing: what ethical/legal risk you personally focused on (privacy, accessibility, safety) and how you mitigated it.
19. AI-Minimal compliance statement: confirm you adhered to the brief and did not use GenAI for code generation or content generation beyond spelling/grammar checks.

7 Safety and responsible use (must comply)

- Do not claim your demo reflects real brands unless your data is clearly synthetic or derived from openly licensed sources with correct attribution.
- Avoid defamatory or misleading product claims. Your 'confidence' labels must be explained and must not imply official certification unless evidence supports it.
- Do not upload sensitive personal data or third-party confidential information.

8 Marking Rubric

The same COMM2020 marking rubric applies across all project options. The rubric will be provided by the module team on ELE and used consistently for CW1 and CW2 (including the individual reflection in CW2).

[END OF SPECIFICATION]

Document owner: Module team (COMM2020)

Date: 23 December 2025