COMM2020 Team Project
Project Specification (Project 5)
Campus Carbon Challenge Game and Dashboard

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.
- Use transparent calculations for carbon impacts and cite the source of conversion factors in your report.
- Your system must remain usable with a seeded dataset even if external services are unavailable.
- If you use third-party APIs or services, handle failures gracefully and avoid embedding secrets (use environment variables).

1 Requirements

Build a web application that helps campus users understand and reduce their carbon footprint through a challenge-based game. The system must allow users to log actions, estimate carbon savings, join challenges, earn points, and visualise progress at individual and cohort levels.

1.1 The practical problem

Universities are expected to reduce emissions and report progress. However, individual behaviour change is difficult because impacts are invisible, data is confusing, and motivation is low.

Your system addresses this by providing:

- A clear model for estimating carbon impact from everyday actions.
- Challenges and missions that make behaviours measurable and rewarding.
- Dashboards that show progress, uncertainty, and the assumptions behind calculations.
- A mechanism for groups (e.g., halls, societies, classes) to compete or collaborate.

1.2 The game concept: Campus Carbon Challenge (required)

Users participate in weekly and monthly challenges that focus on specific behaviours. You must implement:

- Personal missions: small actions logged by individuals (e.g., ‘cycled to campus’, ‘ate vegetarian meal’).
- Group challenges: teams compete (e.g., halls or societies) based on total verified points or estimated CO2e saved.
- Badges: earned for milestones (streaks, variety of actions, evidence submission).

- Season leaderboard: overall points across the term.

2 Users and user profiles

Your application must support three user types:

2.1 Participants (students/staff)

- Create a profile (non-sensitive), join a group, and log actions.
- Join challenges, see mission feedback, and view personal progress dashboards.
- Upload evidence where required (e.g., photo of reusable cup) with clear privacy guidance.

2.2 Challenge organisers (moderators)

- Create and manage challenges (rules, points, evidence requirements, start/end dates).
- Review flagged submissions and moderate disputes (approve/reject with a reason).
- View analytics (participation rates, common failure reasons).

2.3 Maintainers (developers)

- Maintain data models, calculations, deployments, and testing.

3 Features

3.1 Core workflow (required)

1. Action logging: users log actions against a taxonomy (travel, food, energy, waste).
2. Carbon estimation: the system converts actions into CO2e saved (or emitted) using explicit conversion factors and assumptions.
3. Challenge participation: users join challenges and submit actions to earn points; the system applies scoring rules.
4. Evidence and verification: some actions must require evidence (configurable by challenge) and go through moderation.
5. Dashboards: personal dashboard (actions, points, CO2e trend) and group dashboard (leaderboards, participation).
6. Transparency: for each action, show the factor(s) used and a short explanation of the calculation.

3.2 Challenge and moderation rules (required)

- At least 8 challenges seeded (mix of personal and group challenges).
- At least 4 challenges must require evidence and moderation decisions.
- Moderation workflow must track: submission status, moderator decision, decision reason, and timestamps.

- Anti-gaming checks: at least 6 checks such as duplicate submissions, unrealistic frequency, or contradictory logs.

3.3 Data visualisation (required)

- At least 6 dashboard views across the system (e.g., weekly CO2e, action breakdown by category, group leaderboard, verification rate).
- At least 2 views must include uncertainty or caveats (e.g., ranges, disclaimers, confidence levels).
- Users must be able to filter by date range and category.

3.4 Technical requirements (required)

- Authentication: role-based access control (participant/moderator).
- Security: secrets via environment variables; no secrets in repository or ELE ZIP; protect uploaded evidence files.
- 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: authentication/authorisation, action logging, carbon calculation correctness, challenge scoring, moderation workflow, and dashboards/filters.
- 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, but the core demo must function using your seeded dataset.

4.1 Minimum dataset contents (required)

- At least 500 action logs seeded across at least 60 users and 10 groups (synthetic data is acceptable).
- At least 8 challenges (as above) with at least 200 challenge submissions across them.
- At least 80 evidence submissions seeded and at least 40 moderation decisions seeded.

- A conversion factor table for CO2e calculations (with sources recorded in your report).
- At least 100 pre-seeded ‘edge case’ submissions to test anti-gaming checks and moderation.

4.2 Suggested entities (example)

Entity	Key fields (minimum)
User	user_id, role, display_name, group_id, settings
Group	group_id, name, type (hall/society/class), members
ActionType	action_type_id, category, name, unit, default_factor_id
ConversionFactor	factor_id, source, unit_in, unit_out, value, notes, uncertainty(optional)
ActionLog	log_id, user_id, action_type_id, quantity, date, evidence_required, calculated_co2e, confidence
Challenge	challenge_id, title, scope (personal/group), rules, scoring, start/end dates
Submission	submission_id, challenge_id, user_id/group_id, linked_action_logs, points, status
ModerationDecision	decision_id, submission_id, moderator_id, decision, reason, timestamp
AntiGamingFlag	flag_id, submission_id, flag_type, rule_triggered, status

5 Measures of success (evaluation)

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

- Calculation credibility: random sample audit comparing your calculations against expected factor usage and units.
- User understanding: short task-based evaluation (e.g., can users explain a calculation after reading transparency panel).
- Engagement: challenge participation rate and retention (repeat submissions).
- Moderation effectiveness: proportion of flagged submissions resolved; false positives/negatives discussion.

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.

Notes: If you include credentials, use low-privilege demo accounts only and change passwords after marking.

Sprint 1 deliverable expectations

- Prototype v0.1.0 (vertical slice): user logs an action → sees transparent carbon calculation → submits to a challenge → points awarded → dashboard updates.
- Moderator workflow: demonstrate one evidence-required submission being approved/rejected with a reason.
- 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, factor model design, initial evaluation evidence, and a sprint plan for CW2.
- Ethical/legal considerations must cover privacy for evidence uploads, data retention, fairness/incentives, accessibility, 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 sustainability problem you solve and why the game helps.
8. 6–7 minutes: demo the vertical slice including one calculation transparency panel and a challenge submission.
9. 1–2 minutes: show moderation workflow and anti-gaming checks.
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 challenge participation with multiple challenge types; robust dashboards and filtering; clear transparency panels.
- Moderation and anti-gaming: complete workflow with seeded cases and clear logs; demonstrate at least 3 anti-gaming rules triggering correctly.
- 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 environmental impact and unintended consequences (perverse incentives).
- 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 final system including dashboards, moderation, and challenge diversity.
13. 1–2 minutes: show handover pack and how a maintainer would update conversion factors and add a new challenge.
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 collect sensitive personal data. If you use photos as evidence, provide clear guidance and allow users to blur/redact identifying content.
- Be transparent about assumptions in carbon calculations; avoid overstating accuracy.
- Do not upload 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