Assignment Template: Project Assignment 1

Team Name: _18_________________________
Team Members: ___Elang Sisson, Andrew Varkey_____________________
Course Code: __421________________________
Date: ___9/21/2025______________________________

1. Functional Requirements (FRs)

(List at least 8–10. Use FR-01, FR-02 … numbering)

- **FR-01:** The system shall allow an Analyst to upload one or more code artifacts in at least C/C++, Python, Java, and JavaScript, up to 10 MB per artifact.
- **FR-02:** The system shall perform authorship attribution and return a ranked list of top 5 candidate authors with associated probabilities and confidence intervals.
- **FR-03:** The system shall provide authorship verification mode: given a claimed author and code artifact, output ACCEPT/REJECT with a calibrated score in [0,1].
- **FR-04:** The system shall support an adversarial transformation toolkit (e.g., identifier renaming, comment/style perturbation, control-flow rewriting, dead-code insertion) and re-evaluate attribution post-transform.
- **FR-05:** The system shall export results as a digitally signed PDF and JSON report including inputs, configuration, metrics, and explanation artifacts.
- **FR-06:** The system shall compute and display feature level explanations that contributed most to the decision.
- **FR-07:** The system shall provide an audit log of all actions (uploads, model runs, exports, and dataset changes) with user, timestamp, and hash of artifacts.
- **FR-08:** The system shall provide model management: select baseline and advanced models, record hyperparameters, and log versioned runs with metrics.
  …

2. Non-Functional Requirements (NFRs)

(List at least 5–7. Use NFR-01, NFR-02 … numbering)

- **NFR-01:** Uploaded code shall be stored encrypted at rest and in transit.
- **NFR-02:** Every run shall be re-creatable from stored dataset snapshot, model version, and configuration hash; checksum validation must pass.
- **NFR-03:** Shall be able to complete an end-to-end attribution in ≤ 10 minutes.
- **NFR-04:** On the baseline evaluation corpus, top-1 attribution accuracy shall be ≥ 70% and verification AUROC ≥ 0.85.
- **NFR-05:** For a 300 KB code artifact, attribution shall complete in ≤ 5 seconds.
  …

3. User Stories with Acceptance Scenarios

(Write as many as reasonably possible at this stage. Each story must be peer-reviewed in the team. Add ***at least two scenarios per story*** *— one positive, one negative/edge case.)*

US-01:

As an Analyst, I want to upload a code snippet and select an attribution model so that I can identify the most likely author.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Scenario: Successful attribution for supported language

Given I am on the Attribution page

And I have a valid Python file under 300 KB

And I select the "Baseline Stylometry" model

When I click "Run Attribution"

Then I should see a ranked list of top-5 authors with probabilities

And the job status should be "Completed" within 5 seconds

Scenario 2: Negative/Edge Case

Scenario: Rejected upload for unsupported type

Given I am on the Attribution page

And I attempt to upload a binary .exe file

When I click "Run Attribution"

Then I should see an error message "Unsupported file type"

And no attribution job should be created

US-02:

As a Researcher, I want to verify a claimed author for a code file so that I can confirm or refute authorship with a calibrated score.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Scenario: Accept verification for matching author

Given I am on the Verification page

And I provide author "A.Smith" and a Java file written by A.Smith

When I click "Run Verification"

Then I should see result "ACCEPT" with a score ≥ 0.75

Scenario 2: Negative/Edge Case

Scenario: Reject verification for non-matching author

Given I am on the Verification page

And I provide author "A.Smith" and a Java file written by B.Lee

When I click "Run Verification"

Then I should see result "REJECT" with a score ≤ 0.25

US-03:

As a Red-Team user, I want to apply adversarial transformations to code so that I can assess how robust the attribution is.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Scenario: Re-evaluation after identifier renaming

Given I have completed an attribution run with top-1 author "A.Smith"

And I open the Adversarial Toolkit

When I apply "Identifier Renaming" and re-run attribution

Then I should see updated probabilities and a robustness delta report

Scenario 2: Negative/Edge Case

Scenario: Invalid transformation configuration

Given I open the Adversarial Toolkit

And I set dead-code insertion rate to 200%

When I apply transformations

Then I should see an error "Invalid parameter range" and no changes applied

US-04:

As an Analyst, I want explanation artifacts so that I can understand why the model predicted a given author.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Scenario: Feature contribution visualization available

Given an attribution result is available

When I open the "Explanation" tab

Then I should see top contributing features

Scenario 2: Negative/Edge Case

Scenario: Explanation unavailable for legacy model

Given I run attribution with a legacy model lacking explainability support

When I open the "Explanation" tab

Then I should see a notice "Explanation not available for selected model"

US-05:

As a Project Admin, I want dataset and model versioning so that runs are reproducible and auditable.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Scenario: Reproducing a prior run

Given a completed run with dataset snapshot v1.2 and model v0.9

When I click "Reproduce Run"

Then a new job is created with identical configuration hashes

And the checksum validation should pass

Scenario 2: Negative/Edge Case

Scenario: Dataset mutation blocked without version bump

Given a dataset snapshot v1.2 is marked immutable

When I attempt to overwrite files in v1.2

Then I should see an error "Snapshot is immutable — create a new version"

4. Brainstorming and GenAI Reflection

Step A: Team Brainstorming

(Document initial FRs, NFRs, stories, and scenarios created without AI. Record 3–4 key decision points from your discussion.)

- Decision Point 1: Prioritize both attribution and verification modes.
- Decision Point 2: Require reproducibility via dataset and model versioning.
- Decision Point 3: Provide explanation artifacts to increase analyst trust and support reportings

Step B: GenAI-Assisted Brainstorming

(Summarize what GenAI suggested — new FRs, NFRs, user stories, or scenarios.)

- GenAI Suggestion 1: Add audit logging with cryptographic hashes for chain-of-custody.
- GenAI Suggestion 2: Define measurable NFRs (p95 latency, AUROC targets, availability SLO).
- GenAI Suggestion 3: Include export to digitally signed PDF/JSON for evidence packages.

Step C: Refined Requirements & Stories

*(Update Step A with insights from GenAI. Clearly mark new/modified items — e.g., FR-06 **[Added after GenAI]**.)*

- **FR-06 [Added after GenAI]:** Export digitally signed PDF/JSON reports.
- **NFR-04 [Added after GenAI]:** Add accuracy targets (Top-1 ≥ 70%, AUROC ≥ 0.85).
- **NFR-05 [Modified after GenAI]:** Set explicit p95 latency target (≤ 5 seconds for 300 KB).

Step D: Reflection (200–300 words)

- How did you feel about using GenAI in this exercise (e.g., empowering, surprising, confusing, over-reliant)?
- In what ways did GenAI change or improve your brainstorming compared to your team's initial work? (Consider clarity, creativity, and coverage.)
- Did GenAI help you uncover new functional, non-functional, or business requirements that you had not considered before? Provide examples.

Since our team is still getting familiar with the project, using GenAI helped us clarify the scope of the project. At first we focused on the two big pieces, authorship attribution and verification. AI helped us to come up with clear requirements and to think about how our project as a product can be used on a day to day.

With AI's help we were able to create measurable goals such as the availability and reproducibility expectations, and accuracy of models and data. It also reminded us about important stuff such as audit logs, result explanations from the models, and file compatibility checking. These features are basic, but make the tool easier to use and more reliable.

GenAI didn't change any of our ideas, instead it helped us organize them. For example, suggested edge case scenarios like unsupported files, invalid parameters, and rate limits. This improved our Gherkin scenarios and made the stories easier to create and analyze. We also thought about the protection of the corpuses which is made up of uploaded code and being transparent about the model's capabilities.

Overall, AI sped up drafting and made our plans clearer. This assignment left us with a more defined set of requirements and scenarios that we can build on and against.

Final Submission Checklist

- Functional Requirements (8–10+)
- Non-Functional Requirements (5–7+)
- User Stories (all possible at this stage, with acceptance scenarios)
- Gherkin Scenarios (≥2 per story, reviewed)
- Brainstorming & GenAI Section (Steps A–D complete)
- Reflection (200–300 words)
- PDF format, file named correctly

File name format:
TeamName_Assignment1_Requirements.pdf