Name: Student ID:

Agentic AI for Business and FinTech (FTEC5660)
Reproducibility Work, Due date: 1 March, midnight.

Instructions

- **Modality:** This is an individual homework. Students are encouraged to discuss with other students, but each student should submit their own work.
- **What you are doing:** Pick an agentic system project (code on GitHub), get it running, reproduce at least one reported result/behavior, then make one small, well-scoped modification and report what changed.

  This assignment is modeled after research-community reproducibility efforts: the goal is to verify what works, what does not, and why. This exercise is *not* about "pass/fail" a paper.

1 Choice of Project

You have two options:

1. Pick from the instructor-provided list (link on the course page). Link: `https://shorturl.at/yx112`, or
2. Choose your own, as long as it is an agentic system and has enough documentation/results to evaluate.

Definition (practical): an "agentic system" is a program that plans and acts over multiple steps (often with tools, memory, search, code execution, multi-agent loops, etc.), not just a single prompt-response.

Minimum eligibility checklist:

- Public GitHub repo (or shareable private repo) with a runnable pipeline.
- A target to reproduce: a table/figure/metric, or at least a clear claimed capability with an evaluation script.
- You can run some meaningful subset within your compute/time budget.

2 Scope: you do not need to reproduce everything

Many agent papers are huge. You are allowed to reproduce one small task/experiment that is clearly defined (e.g., one dataset split, one benchmark, one ablation, one environment, one table row). This is normal in reproducibility work—community guidelines explicitly note you don't have to reproduce all experiments, especially if resource-heavy, and you can focus on a subset like baseline results.

Pick a "target claim" early. Example targets: "Table 2 accuracy on X", "success rate on Y tasks", "tool-call reliability", "cost per solved task", "latency", "win-rate vs baseline", etc.

3 You are not re-implementing from scratch

You do not need to re-build the method from the paper description. Using the authors' repo is explicitly acceptable in standard reproducibility challenges. Minimum expectation: run the repo (or a subset) and produce your own measured outputs, then compare them to what the paper/repo reports.

4 Edge case A: the repo uses an old/unsupported LLM → you may switch to DeepSeek

Many repos hard-code older models or deprecated endpoints. You may replace the model with DeepSeek (or another accessible model), as long as you document the change and treat it as a controlled variable. DeepSeek's API is designed to be OpenAI-compatible, so many repos can switch by changing config (model name + base URL + key) rather than rewriting the whole stack.

When you swap models:

- Record: model name, provider, base URL, decoding params (temperature/top_p/max_tokens), and any prompt/template changes.
- Be explicit: "Results are not directly comparable to the paper because the underlying LLM differs." Then compare as fairly as possible (same tasks, same metrics, same evaluation script).

5 Edge case B: the paper has many tasks → extract one small task and report

If the repo includes multiple benchmarks, pick one and go deeper:

- Reproduce the reported number(s) for that benchmark
- Identify hidden assumptions (dataset version, prompt format, seed, tool availability)
- Run 3–5 trials (if stochastic) and report mean/variance

This aligns with how reproducibility reports emphasize "what parts reproduce, at what cost, and under what conditions," rather than a binary outcome.

6 What counts as a "modification"?

Make one change that is small but meaningful, such as:

- Model change (e.g., to DeepSeek) with careful logging
- Prompt/tool policy change (e.g., stricter tool-use rules, different system prompt)
- Ablation (remove memory, remove planner step, disable a tool)
- Parameter change (max steps, reflection on/off, retrieval k, etc.)
- Evaluation tightening (fix a bug, pin dataset version, add seeds, add missing metric)

Your modification should be: (i) Isolated (one main change), (ii) Measurable (you can quantify impact).

7 Submission package (required)

Submit all three:

- Brief report (max 10 pages). Suggested structure:
  - Project summary + what you tried to reproduce

  - Setup notes (env, data, keys, compute)
  - Reproduction target(s) + metric definition
  - Results: your numbers vs reported numbers (tables/plots ok)
  - Your modification + results after modification
  - Debug diary: main blockers + how you resolved them
  - Conclusions: what is reproducible, what isn't, and why
- GitHub link that contains:
  - Clear README with install + run steps
  - Your changes in commits (no giant unreviewable dump)
  - No secrets/API keys committed
- Presentation link (maximal 15 minutes). A recorded Zoom presentation where you explain:
  - What you attempted to reproduce
  - Your modification and its measured impact
  - What worked / what didn't
  - Key lessons + recommendations to future users