# SE4471 - Lab 1 Part 1: Vibe Coding a Web-Based Tic-Tac-To

**Due:** February 5th 2026, 11:59pm

**Submission Platform:** OWL and GitHub

---

## Purpose

This lab introduces **LLM-assisted software development** through a *vibe coding* approach.

You will build a small web application by interacting with a large language model and documenting how your solution evolves over time. The goal of this lab is **not** to test how well you can manually code Tic-Tac-Toe, but to examine:

- how you naturally interact with an LLM,
- how the application changes across interactions,
- how you evaluate partial or imperfect outputs,
- and how you make decisions as the system evolves.

A working application is required, but **your interaction process is the primary focus of assessment**.

---

## Allowed Tools

- Any LLM-assisted coding tool (e.g., Cursor, ChatGPT, Copilot Chat, Claude).
- Any programming language or framework **as long as the final application runs in a browser**.
- The application **must be hosted using GitHub Pages**.

---

## LLM Tool Disclosure (Required)

You must clearly state **which LLM tool(s)** you used for this lab.

Include this information:

- in your **LLM Interaction Log**, and
- in the **README.md** of your repository.

If you used more than one tool, list all of them and briefly note how they were used (e.g., “initial generation in ChatGPT, refinement in Cursor”).

This disclosure is for transparency only and **does not affect your grade**.

---

## Repository Setup (GitHub Classroom)

This lab must be completed using the **GitHub Classroom repository** provided to you.

- Accept the GitHub Classroom assignment here.
- Use the repository that is automatically created for you.
- Do **not** create a separate repository for this lab.

All development, commits, and deployment must occur from your Classroom repository.

---

## What You Are Building

A simple, web-based **Tic-Tac-Toe** game for two people playing on the same device.

The game should behave in a way that would feel familiar to someone who has played Tic-Tac-Toe before. Specific design and implementation decisions are intentionally left to you.

---

## Game Expectations (High-Level)

Your application should allow players to:

- interact with a visible 3×3 grid,
- take turns,
- reach a clear game outcome,
- and start a new game without refreshing the page.
- give the option to restart the game at any time.

The interface should communicate what is happening as the game progresses.

---

## Required Extra Feature (Choose ONE)

In addition to basic gameplay, implement **one** of the following:

- **Keyboard support**

  The game can be played without using a mouse.

**OR**

- **Appearance customization**

  A simple option that changes how the board or marks appear (e.g., theme toggle, mark style).

**OR**

- **Enhanced game feedback**

Additional visual feedback beyond plain text (e.g., highlighting, disabled states, hover effects, brief messages).

The feature must be **visible and testable** in the deployed application.

---

## Use of LLMs and Manual Coding

This lab is explicitly about **LLM-assisted coding**.

### What is allowed

- Using LLM-generated code as the primary source of your implementation.
- Making **small, localized manual edits**, such as:
    - fixing typos,
    - renaming variables,
    - minor adjustments after testing.

If you make a **non-trivial manual change**, you must:

- note it in your interaction log **or**
- add a short comment in the code explaining the change.

### What is not allowed

- Writing most of the application manually and then asking the LLM to review or explain it.
- Implementing core game logic entirely by hand.
- Prompting the LLM after the fact to justify already-written code.

If the majority of the implementation does not originate from LLM interactions, the lab objectives are not met.

---

## LLM Interaction Log

You must submit an **LLM Interaction Log** using the provided template.

---

### Full Interaction Record (Required)

You must provide a record of **all meaningful LLM interactions** used during development.

For **each interaction**, include:

- the **prompt** (verbatim), and
- a **brief summary of the LLM's response** (1–2 sentences describing what the model attempted to do).

Do **not** include full LLM output in this section.

There is **no expected or ideal number** of interactions.

Using many prompts will **not** reduce your grade.

---

## Selected Interaction Analysis (3 Required)

From your full interaction record, select **three (3)** interactions that were the **most meaningful** in shaping your final application.

For **each selected interaction**, you must include:

- the **prompt** (verbatim),
- the **LLM output** (verbatim or lightly trimmed), and
- answers to the following questions:
  1. What were you trying to achieve?
  2. What changed in the application as a result?
  3. What worked as expected?
  4. What did not work or required adjustment?
  5. What decision did you make next because of this outcome?

Among the three selected interactions, you must include:

- **one interaction** that led to the *initial version* of the application, and
- **one interaction** that resulted in a *visible UI change*. (You may need to ask the LLM to change something specifically for this)

The third interaction is your choice.

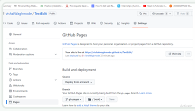
---

## GitHub Pages Deployment (Required, Not Graded)

You must deploy your application using **GitHub Pages** from your **GitHub Classroom repository**.

Deployment is required so the application can be run and reviewed, but **no marks are awarded specifically for deployment**.

You can use the following sites to deploy based on your chosen technologies:



Deployment of React Application using GitHub Pages - GeeksforGeeks

Your All-in-One Learning Portal: GeeksforGeeks is a comprehensive educational platform that empo…

Author **GeeksforGeeks**

Deploying a Frontend Project  on Github - GeeksforGeeks

Your All-in-One Learning Portal: GeeksforGeeks is a comprehensive educational platform that emp...

Author GeeksforGeeks

Your repository must include a README.md with:

- instructions for running the project locally,
- the GitHub Pages URL,
- the LLM tool(s) used.

---

## Submission on OWL

Submit **all of the following**:

1. **GitHub Pages link**
2. **LLM Interaction Log as a PDF**

---

## Assessment Breakdown (Out of 20)

| **Component** | **Points** |
|---|---|
| Working application (basic gameplay present) | **4 pts** |
| Required extra feature implemented | **2 pts** |
| Full LLM prompt record + response summaries | **4 pts** |
| Selected interaction analysis (3 × 3 pts) | **9 pts** |
| Evidence of LLM-assisted process (log ↔ code consistency) | **1 pt** |
| **Total** | **20 pts** |

> A polished application with weak or missing process documentation will not receive full credit.

---

## Important Notes

- You are **not graded** on how prompts are phrased.
- You are **not graded** on the number of prompts used.
- Honest trial-and-error is expected and encouraged.
- Transparency about tools and process is required.