- **Person 1: Dataset + Token-Level Watermarking (Approach B)**

  Tentative- Aime Cesaire Mugishawayo

Task	What It Means
Finalize and curate ~1,000 prompts	Gather about 1,000 short texts/questions from two sources: OpenAI Evals and WikiText-103. These prompts will be used as inputs to generate text with and without watermarks.
Implement token-level watermark	Re-create the watermarking method from Kirchenbauer et al. (2023): modify the way an LLM (like Llama-2-7B-Chat) picks words when generating text so that some words are more likely (greenlist) based on a secret key.
Handle token-level watermark embedding	Actually run the Llama model and generate watermarked text using the modified sampling process.
Detection experiments for token watermark	After generating text, use statistical tests (like a z-test) to check if you can detect the watermark — meaning: is there an unusually high number of greenlisted words?
Robustness experiments (token)	Test how well the token-level watermark survives attacks like paraphrasing, editing, etc. (Does the watermark still show up after changes?)

- **Person 2: Prompt-Based Watermarking (Approach A)**

  Tentative -Admire Madyira

Task	What It Means
Design and code the prompt wrapper	Create a program that adds hidden instructions into prompts (like "Use at least one word with double letters") that cause the generated text to carry a subtle, machine-detectable pattern.
Generate prompt-watermarked texts	Send these wrapped prompts to GPT-3.5-Turbo and collect the text it generates — this will be your watermarked dataset for Approach A.
Implement BERTa classifier training	Train a small BERT-like model to automatically tell apart normal text vs. text with hidden prompt-based watermarks.
Zero-shot detection (Zhong et al.)	Try detecting the watermark without any special training by cleverly prompting another LLM ("Does this text seem watermarked?").
Robustness and ablation experiments (prompt)	Experiment with different rules (double-letters, rare synonyms, etc.) and test which rules are easiest or hardest to detect.

- **Person 3: Analysis, Evaluation, and Write-up**

  Tentative- Miro Babin

Task	What It Means

Set up evaluation framework	Decide how to fairly compare prompt-based vs. token-based methods — such as: which one is easier to detect, which one hurts output quality less, etc. Build the scripts to do these evaluations.
Analyze results	After text generation and detection experiments, carefully look at the numbers: how accurate are detections? how much does the watermark hurt the quality of the text? Summarize insights.
Write comparative analysis section	Write a formal section in the report explaining which method is better, under what conditions, and why (strengths, weaknesses, deployment ideas).
Manage final deliverables	Make sure all parts — code, report, poster datasets, detection tools — are clean, complete, and ready to submit.