Analysis Report

Names: Mrigank, Mohit, Shazi, Mayan

Section: F

SRNs: PES2UG23CS352, CS350, CS348, CS332

1. Strategies

- Our project implements the mandated hybrid system by combining a fast statistical "oracle" with a Q-learning "brain". This approach was designed for a strategic balance of high-speed performance and intelligent decision-making.

HMM Design (The "Oracle")

- Instead of a computationally heavy HMM, our `HangmanHMM` functions as a high-speed "smart frequency calculator." During its one-time training, it sorts all 50,000 corpus words into "buckets" based on their length. When the agent requests a prediction for a pattern like `_ P P _ E`, the oracle instantly filters the 5-letter bucket for all possible matching words. It then counts the letter frequencies *only* in the blank spots of this new candidate list and returns this probability distribution to the RL agent as statistical advice.

RL Design (The "Brain")

- The "brain" is a Q-Learning agent that learns from experience. We defined the game **state** as a unique, hashable string: `"{pattern}:{guessed_letters}:{lives_left}"`. This is memory-efficient and allows us to use a simple Q-table instead of a complex neural network. The **reward function** was designed to directly optimize the hackathon's scoring formula : it gives a large `+50` reward for winning, small penalties for wrong (`-1`) or repeated (`-5`) guesses, and a large penalty for losing (`-10`) . This teaches the agent to prioritize winning efficiently.

2. Exploration:

We manage this trade-off with a hybrid epsilon-greedy strategy. The agent **exploits** its knowledge 85% of the time by choosing an action based on a 50/50 weighted average of its own Q-table "experience" and the HMM's statistical advice. For the other 15% of the time, it **explores**. This exploration is not random; instead, it's a "smart exploration" where the agent samples a guess from the HMM's probability distribution. This ensures even our exploratory moves are statistically likely to be good, speeding up the learning process significantly.

3. Key Observations:

The main challenge was the HMM mandate . A full HMM would be too slow, so our key insight was to build a fast, pattern-matching frequency model that fulfills the *purpose* of the HMM—providing a probability distribution—without the computational cost. A second challenge was the massive state space. Using a simple string-based key instead of a complex vector was a critical decision that allowed us to use a fast Q-table rather than a more complex DQN. This hybrid approach is robust, as the HMM provides strong guidance early in the game, while the RL brain learns to handle specific, tricky late-game scenarios

4. Future Improvements:

If we had more time, we would first replace the Q-table with a **Deep Q-Network (DQN)**, as suggested in the PDF. This would allow the agent to generalize from similar states and be far more memory-efficient. Second, we would implement an **epsilon-decay** schedule. The agent would start with a high exploration rate that gradually decreases, allowing it to explore broadly at first and then confidently exploit its learned knowledge for a higher final score.

```
================================================================================
HMM-RL HYBRID AGENT RESULTS (Time/Success Optimized)
================================================================================

Games Played: 2000
Games Won: 1961
Success Rate: 98.05%

Total Wrong Guesses: 2717
Average Wrong Guesses: 1.36

Total Repeated Guesses: 0
Average Repeated Guesses: 0.00

================================================================================
FINAL SCORE: -11624.00
================================================================================
```