Take-Home Assessment: AI R&D Software Engineer

To: [Candidate Name]
From: Bally's R&D Team
Subject: Bally's R&D - Take-Home Assessment

Hi [Candidate Name],

Thank you for your interest in the R&D Software Engineer role at Bally's! As a next step, we have a short, practical take-home assessment.

The goal of our R&D unit is to rapidly explore "what's next" through timeboxed prototyping. This test is designed to reflect that environment. It's less about production-perfect code and more about your approach to problem-solving, your pragmatism, and your ability to quickly leverage AI tools to build and validate an idea.

We estimate this task will take approximately **4 hours**. Please don't spend more than that. We're most interested in seeing what you can accomplish within a focused timeframe.

The Challenge: LLM-Powered Game Recommender Prototype

Imagine a player on a Bally's online casino site has just finished playing a slot game. To enhance their experience and encourage discovery, we want to show them a component that says, **"Because you played [Game X], you might like..."** followed by a few similar games.

Your task is to build a small web application prototype that demonstrates this concept, using a Large Language Model (LLM) to determine game similarity based on a rich, generated dataset.

Core Objectives

1. AI-Powered Data Generation:

- To make the problem realistic, we need a good-sized dataset. Your first task is to **programmatically generate a dataset of at least 100 fictional casino slot games.**
- Write a script (e.g., generate_data.py) that uses an LLM to accomplish this.
- **This is a key part of the exercise.** You must first define the schema for a "game" by choosing the features you believe are important for determining similarity (e.g., game theme, volatility, special features, art style, etc.).
- Then, instruct the LLM to create a varied and plausible set of 100+ games based on your schema. The goal is a rich and diverse catalog.
- The script should save this data to a structured file (e.g., games.json or games.csv).

2. Build the Similarity Engine:

- With your generated dataset, build the core of the application: a function that takes a "played game" as input and returns a list of the top 3-5 most similar games.
- The method you choose for determining similarity is a key part of the assessment. You will need to design and implement a robust way to compare games from your dataset using an LLM.

3. Create a Simple UI:

- Build a simple, interactive user interface using **Streamlit** or **Gradio**.
- The UI should allow a user to:
    - Select a game from a dropdown/list of all available games (from your generated dataset).
    - Upon selection, the app should display the top 3-5 recommended games.
    - **For each recommendation, include a brief, LLM-generated explanation of why it's a good match.** (e.g., "Like Pharaoh's Fortune, this game also features an Ancient Egypt theme and a high-volatility free spins round.")

What We're Looking For

- **Problem-Solving & Creativity:** Your choices for the game schema are crucial. How did you decide to model a "game"? What assumptions did you make?
- **AI for Data Synthesis:** Your approach to programmatically generating the dataset is a key evaluation point. How effectively did you use an LLM to create a rich and varied set of mock data?
- **Pragmatism & Focus:** How did you use the 4 hours? We're looking for a working proof-of-concept. Your technical choices for the similarity engine are important—did you select a method appropriate for the dataset size and the time-boxed nature of the task?
- **Technical Implementation:** Is the code clean and understandable? Is the use of the LLM effective and logical? Did you think about practicalities like managing an API key (e.g., using environment variables)?
- **Communication:** Your README.md file is as important as your code. It should clearly explain your approach, design decisions, any trade-offs you made, and instructions on how to run your application.

Technical Guidelines

- **Language:** Python
- **UI Framework:** Streamlit or Gradio
- **LLM:** You can use any major LLM provider you're comfortable with (e.g., OpenAI, Anthropic, Google). Please use your own API key.
- **Deliverable:** A link to a Git repository (e.g., GitHub, GitLab) containing:

    1. All your source code.
    2. Your data generation script (e.g., generate_data.py).
    3. The final generated data file (e.g., games.json).
    4. A requirements.txt file.
    5. A detailed README.md file.

How to Submit

Please reply to this email with the link to your public Git repository when you are finished. We'd appreciate it if you could complete it within the next few days.

We are genuinely excited to see how you approach this challenge. Good luck!

Best regards,

The Bally's R&D Team