**📋 Risk Management Report**

**Project Title:** Turkish Joke Generator with Fine-Tuned Language Models  
**Prepared by:** Team 8   
**Date:** 10.04.2025

**📌 1. Introduction**

This risk report aims to identify, assess, and prepare mitigation strategies for potential risks that may affect the development of the Turkish joke generation AI model. Managing risks is a critical part of ensuring the project proceeds on time, meets quality expectations, and maintains team coordination throughout its lifecycle.

**📍 2. Risk Management Approach**

Each risk in this report is evaluated using the following criteria:

* **Risk Description:** A short explanation of the risk
* **Category:** Classification (e.g., Technical, Organizational, Resource, Schedule)
* **Likelihood:** Probability of the risk occurring (Low, Medium, High)
* **Impact:** The level of effect on the project if the risk occurs (Low, Medium, High)
* **Mitigation Strategy:** Actions to prevent or reduce the severity of the risk

All risks are summarized in a table at the end of this report.

**⚠️ 3. Identified Key Risks**

**🔸 R1 - Model Training Takes Longer Than Expected**

* **Category:** Technical
* **Description:** Fine-tuning large-scale models like Mistral v1 can take considerable time, especially on limited platforms such as Kaggle or Colab.
* **Impact:** Medium
* **Likelihood:** High
* **Mitigation:** Use checkpoint saving, early stopping, experiment with reduced parameters, and explore prompting as an alternative strategy.

**🔸 R2 - Low Output Quality Despite Training**

* **Category:** Technical
* **Description:** The model may produce low-quality or contextually weak outputs, especially with limited or inconsistent training data.
* **Impact:** High
* **Likelihood:** Medium
* **Mitigation:** Test with more robust models (e.g., GPT-4), increase training dataset size and diversity.

**🔸 R3 - Kaggle GPU / Memory Limitations**

* **Category:** Resource
* **Description:** Due to hardware limitations, large models like Mistral may fail to load or train properly on Kaggle.
* **Impact:** Medium
* **Likelihood:** High
* **Mitigation:** Use smaller model configurations, switch to alternative platforms (Colab Pro, local machine).

**🔸 R4 - Insufficient or Unbalanced Dataset**

* **Category:** Data
* **Description:** A dataset limited to 1000–5000 jokes may restrict the model’s ability to generalize effectively.
* **Impact:** Medium
* **Likelihood:** Medium
* **Mitigation:** Continue data collection, refine and balance the dataset through cleaning and preprocessing.

**🔸 R5 - API Usage Limitations or Cost Issues**

* **Category:** Resource
* **Description:** Using commercial APIs like GPT-4 may lead to cost or quota limitations.
* **Impact:** Low
* **Likelihood:** Medium
* **Mitigation:** Track token usage carefully, use minimal input prompts during testing, utilize free quota efficiently.

**🔸 R6 - Uneven Task Distribution Within the Team**

* **Category:** Organizational
* **Description:** Poor task distribution may result in workload imbalances among team members.
* **Impact:** Medium
* **Likelihood:** Medium
* **Mitigation:** Define roles clearly at the beginning of each sprint and ensure regular task tracking.

**🔸 R7 - Delay in Frontend / Web Interface Delivery**

* **Category:** Schedule
* **Description:** While focusing on backend model training, frontend development may be delayed or overlooked.
* **Impact:** Medium
* **Likelihood:** Medium
* **Mitigation:** Start frontend tasks early, use simple UI templates, and assign dedicated team members.

**📊 4. Risk Summary Table**

Code which we used for generatemetin, ekran görüntüsü, makbuz, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

|  |
| --- |
| import pandas as pd  # Risk tablosu verisi  data = {  "Risk ID": ["R1", "R2", "R3", "R4", "R5", "R6", "R7"],  "Description": [  "Model training takes much longer than expected",  "Low output quality despite training",  "Kaggle environment limitations (GPU, storage)",  "Insufficient or inconsistent dataset",  "API usage cost or quota issues",  "Uneven task distribution in team",  "Frontend/web interface not delivered on time"  ],  "Category": [  "Technical", "Technical", "Resource", "Data", "Resource", "Organizational", "Schedule"  ],  "Likelihood": ["High", "Medium", "High", "Medium", "Medium", "Medium", "Medium"],  "Impact": ["Medium", "High", "Medium", "Medium", "Low", "Medium", "Medium"],  "Mitigation Strategy": [  "Use checkpoints, early stopping, parallel prompting",  "Try different models (e.g., GPT-4), use more data",  "Use alternative platforms (Colab, local), reduce data",  "Continue data collection and cleaning",  "Control token usage, test with small batches",  "Clearly assign tasks at sprint start, track progress",  "Start UI early, use simple templates or libraries"  ]  }  # DataFrame oluştur  df = pd.DataFrame(data)  # 🔵 Renkleme fonksiyonu  def highlight\_severity(val):  if val == "High":  return "background-color: #ff9999" # Kırmızı  elif val == "Medium":  return "background-color: #fff29e" # Sarı  elif val == "Low":  return "background-color: #b3ffb3" # Yeşil  return ""  # Styler ile renklendir  styled = df.style \  .applymap(highlight\_severity, subset=["Likelihood", "Impact"]) \  .set\_caption("Software Engineering Risk Table")  styled |

**📎 5. Conclusion and Recommendations**

The risks outlined in this report cover the major areas of concern for the current project phase. To ensure a successful outcome:

* Maintain both model training and prompting as parallel strategies
* Monitor team roles and responsibilities throughout the sprint
* Manage time and resources carefully, especially when training large models or using external APIs

Risk assessment should be a continuous process and revisited at the start and end of each sprint.