Test Case Report for LLM Project

1. Introduction

The purpose of this document is to provide a comprehensive test case report for the **Fıkra Üreten LLM** project, which aims to generate Turkish jokes (fıkra) using a fine-tuned large language model (LLM). This report outlines the testing methodology, test scenarios, test cases, expected outcomes, and actual outcomes, ensuring that the system operates according to functional and non-functional requirements.

The project leverages **transformers** library from Hugging Face, fine-tuned on a Turkish jokes dataset using **PyTorch**, **SentencePiece** for tokenization, and is deployed via **Streamlit** for interactive use.

2. Scope of Testing

The testing focuses on the following aspects:

- Functional correctness of generated jokes
- Tokenization accuracy
- Model inference stability
- UI (Streamlit) interaction and usability
- Response time and latency
- Filtering of invalid characters (e.g., symbols like “??”, “??”)
- Category-based generation validation (e.g., çocuk, doktor, temel)

3. Test Methodology

The testing process followed:

✅ **Unit testing** for tokenizer, preprocessing, and model inference pipeline

✅ **Integration testing** for end-to-end flow (input → model → output → UI display)

✅ **User acceptance testing (UAT)** with non-technical users evaluating joke quality

✅ **Manual and automated tests** for specific output constraints

4. Test Environment

- **Model**: Fine-tuned LLaMA-7B Turkish (from alpkylm/llama-2-7b-chat-turkish)
- **Inference Hardware**: NVIDIA RTX 4090 / CUDA 12.3
- **Web Framework**: Streamlit 1.32
- **Tokenizer**: SentencePiece (trained vocab=10k)
- **Backend**: Python 3.10 + PyTorch 2.0
- **Frontend**: Streamlit interface hosted locally

5. Test Cases

Test Case 1: Valid Joke Generation (Baseline Functional Test)

Test Case 1: Valid Joke Generation (Baseline Functional Test)

Field	Value
Test ID	TC-001
Objective	Validate that the model generates at least 3 lines of valid Turkish text without empty output
Input	Random prompt (or no prompt for unconditional generation)
Expected Output	A joke consisting of 3+ lines, Turkish characters, no placeholder tokens
Actual Output	✅ Passed – model generated 3-5 lines in 98% of trials
Status	Passed

Test Case 2: Character Filter Validation

Field	Value
Test ID	TC-002
Objective	Ensure that symbols like ??, ⁇, or <unk> are not present in output
Input	Random prompt; sample output 100 jokes
Expected Output	0% occurrence of invalid symbols
Actual Output	❌ Failed (observed 3% ?? tokens in raw output)
Action	Added post-processing regex filter to clean output; retested with 0% occurrence
Status	Passed after fix

Test Case 3: Category Classification Accuracy

Field	Value
Test ID	TC-003
Objective	Verify that jokes generated under çocuk category contain relevant vocabulary
Input	Prompt: "çocuk kategorisinden bir fıkra üret"
Expected Output	Joke mentioning child-related words (e.g., "çocuk", "öğretmen", "okul")
Actual Output	✅ Passed – 92% outputs semantically aligned with child context
Status	Passed

Test Case 4: UI Interaction Test

Field	Value
Test ID	TC-004
Objective	Ensure "Generate Joke" button triggers backend model inference and displays result
Input	Button click
Expected Output	Joke displayed in < 3 seconds
Actual Output	✅ Passed – average latency 2.2s
Status	Passed

Test Case 5: Input Parameter Handling

Field	Value
Test ID	TC-005
Objective	Confirm empty or invalid inputs do not crash the model or UI
Input	Empty input, special characters input
Expected Output	Default unconditional generation or sanitized input
Actual Output	✅ Passed
Status	Passed

Test Case 6: Performance Stress Test

Field	Value
Test ID	TC-006
Objective	Evaluate response time under 50 concurrent requests
Input	50 parallel Streamlit clients calling inference
Expected Output	Each response < 10s; no crash
Actual Output	❌ Observed 20% requests > 12s; 1 timeout error
Action	Added async queue, optimized batch size
Retest	✅ Passed (100% requests < 9s)
Status	Passed after optimization

6. Summary of Results

Test Category	Total	Passed	Failed	Retested
Functional Tests	5	4	1	1
UI Tests	1	1	0	0
Performance Tests	1	0	1	1

✅ **All critical tests passed after iteration**

✅ **No blocking issues remain for deployment**

7. Known Issues

- In rare cases (~1%) joke outputs may include low-quality humor or repetition due to small dataset size
- Category overlap between doktor and çocuk jokes observed (expected for certain scenarios)

---

8. Recommendations

- Expand dataset with more diverse fıkras across regions and subcategories
- Further fine-tune with reinforcement learning for humor metrics
- Add user feedback loop in UI for joke quality ranking

Name	Role
Muharrem Şimşek	
Simay Aydın	
Aslımay Mısra Kandar	
Taha Demirhan Gökhan Mert Demirok	