

# Vision and Scope Document

**Project Name:** Joke-Telling AI

**Version:** 1.0

**Date:** [03/04/2025]

**Prepared by:**Project Team 1

**Muharrem Şimşek - 210706001**

**Taha Demirhan - 210706002**

**Simay Aydın - 200706040**

**Aslımay Mısra Kandar – 210706043**

**Gökhan Mert Demirok - 210706311**

---

## 1. Business Needs

### 1.1 Background

In the age of artificial intelligence, humor generation remains a unique and complex challenge due to its deep ties with culture, language, and context. As part of our business group assignment, we are developing a Joke-Telling AI that can generate and deliver jokes interactively. The project allows us to explore the creative side of Natural Language Processing (NLP) while designing an entertaining tool that showcases AI's capacity to mimic human-like humor.

### 1.2 Business Opportunity

- Demonstrates AI's ability to understand, replicate, and deliver humor.
- Opens possibilities for integration into broader applications such as virtual assistants, chatbots, or entertainment platforms.
- Provides team members with hands-on experience in AI development, data handling, and user interface design.

### 1.3 Business Objectives

- Build a functioning AI model capable of generating contextually appropriate jokes.
- Ensure joke output is culturally sensitive and non-offensive.
- Deliver a user-friendly interface enabling seamless interaction.
- Lay the foundation for potential multilingual joke delivery.

---

## 2. Vision of the Solution

## 2.1 Vision Statement

"To create an AI-powered joke generator that delivers humorous, engaging, and culturally appropriate jokes through a simple and interactive platform."

## 2.2 Key Features

- **Joke Generation:** AI generates jokes across predefined categories (e.g., puns, one-liners, knock-knock).
- **User Interaction:** Users interact with the system via a text-based or chatbot interface.
- **Joke Rating (Optional):** Users can rate jokes to help improve future performance and relevance.
- **Multi-language Support:** Initial version in Turkish, with scalability to other languages (e.g., English).

## 2.3 Assumptions & Dependencies

- Utilizes pre-trained NLP models such as GPT, BERT, or custom fine-tuned models.
  - Requires a curated, clean dataset of jokes for training and filtering.
  - Deployment relies on a lightweight frontend and backend integration (e.g., Flask, Node.js, or Telegram bot framework).
  - Internet or cloud access may be required for model inference if not hosted locally.
- 

## 3. Scope

### 3.1 In-Scope

#### Core Functionalities:

- Joke generation using AI models.
- User inputs for category/joke request.
- Categorized joke output via UI/chatbot.

#### Technical Deliverables:

- NLP model integration and backend logic.
- Simple frontend interface (e.g., web app or chatbot).
- Joke dataset curation and cleaning.

#### Development Tools and Frameworks:

- NLP tools (OpenAI, Hugging Face Transformers, etc.)
- Flask/Django or Bot API (e.g., Telegram).
- Simple cloud or local deployment setup.

3.2 Out-of-Scope

Advanced Functionalities (Future Work):

- Speech recognition for voice-activated joke requests.
- Personalized humor recommendations based on user profiles.
- Real-time emotion or sentiment-based joke customization.

Complex AI Capabilities:

- Reinforcement learning for adaptive joke generation.
- Sentiment analysis beyond keyword-based filtering.

4. Stakeholders

Role	Responsibility
Developers	Build, integrate, and test the AI model and user interface.
Data Collectors	Gather, filter, and clean the joke dataset.
Testers	Evaluate the quality, usability, and functionality of the system.
Instructor/Client	Provide project guidance, evaluation, and feedback.

5. Risks & Challenges

Risk	Mitigation Strategy
Humor is subjective	Use diverse datasets and allow user feedback to adapt content.
Dataset quality	Manually review and filter jokes for appropriateness.
Technical limitations	Start with pre-trained models and focus on basic integration.
Cultural differences	Start with a local language (Turkish) and test with a targeted audience.

6. Timeline & Milestones

Week	Milestone
Week 1	Literature review, tool selection, and dataset gathering.
Week 2	Model selection/training and joke generation logic.
Week 3	Build user interface and connect with backend.

## Week Milestone

Week 4 Testing, feedback incorporation, and final presentation.

---

### 7. Success Metrics

- **Joke Quality:** At least **70%** of users rate generated jokes as "funny" or "acceptable".
  - **User Engagement:** On average, each user requests **2 or more jokes** per session.
  - **Technical Completion:** Functional prototype with integrated frontend and backend by deadline.
  - **Feedback Score:** Achieve at least **80% satisfaction** in the final demo or evaluation.
- 

### 8. Approval

Approved by (Instructor/Client): Ensar Gül