Project Vision & Scope Document

Turkish Joke Generator with AI

Executive Summary

The Turkish Joke Generator is an innovative web application that leverages artificial intelligence to generate context-specific jokes in the Turkish language. The application focuses on three distinct Turkish joke categories: Karadeniz (Black Sea region), Nasreddin Hoca (a famous Turkish folk character), and Anadolu (Anatolian regional jokes). This project combines modern web technologies with local AI capabilities to deliver a culturally relevant and entertaining user experience.

Vision Statement

To create an accessible, engaging platform that preserves and celebrates Turkish humor through modern technology, while demonstrating the capabilities of local AI language models for Turkish content generation.

Business Objectives

- Create an entertaining application that showcases Turkish cultural humor
- Demonstrate practical implementation of on-premise AI models
- Provide a responsive, user-friendly interface accessible across devices
- Establish a foundation for future AI-powered Turkish language applications

Target Audience

- Turkish-speaking users seeking entertainment
- Language enthusiasts interested in Turkish humor
- Developers exploring AI implementation with Flask and Ollama
- Educational institutions teaching Turkish culture or language

Key Features

1. **AI-Powered Joke Generation**: Integration with Ollama and the DeepSeek-r1 language model
2. **Category Selection**: User ability to choose between three distinct joke types
3. **User Interface**: Modern, responsive design with light/dark mode
4. **Sharing Capabilities**: Social media integration for content sharing
5. **Visual Enhancements**: Animations and visual feedback for user interactions

Success Criteria

- Successful generation of contextually accurate Turkish jokes
- Responsive interface functioning across desktop and mobile devices
- System reliability with proper error handling
- Positive user feedback on joke quality and user experience

Out of Scope

- User account management
- Joke rating or feedback system
- Database storage of generated jokes
- Premium/paid features
- Multi-language support

Project Plan

1. Project Overview

- **Project Name**: Turkish Joke Generator (Türkçe Fıkra Üreticisi)
- **Project Duration**: 4 weeks
- **Project Owner**: [Your Name]

2. Phase Breakdown

Phase 1: Planning & Setup (Week 1)

- Requirements gathering and documentation

- Technology stack selection
- Environment setup (Python, Flask, virtual environment)
- Ollama configuration and model selection

Phase 2: Core Development (Week 2)

- Backend API development
- Integration with Ollama and DeepSeek model
- Basic frontend structure
- Initial testing

Phase 3: UI Enhancement (Week 3)

- Advanced UI implementation
- Responsive design
- Animation and visual effects
- Dark mode implementation
- Social sharing features

Phase 4: Testing & Deployment (Week 4)

- Comprehensive testing
- Performance optimization
- Bug fixes
- Documentation
- Production deployment

3. Milestones & Deliverables

Milestone	**Deliverable**	**Expected Completion**
Project Initiation	Project scope document, environment setup	End of Week 1
Core Functionality	Working API with Ollama integration	Mid-Week 2
Basic UI	Functional web interface with joke generation	End of Week 2

Milestone	Deliverable	Expected Completion
Enhanced UI	Complete responsive UI with animations	Mid-Week 3
Testing Complete	Fully tested application	Mid-Week 4
Production Ready	Deployed application	End of Week 4

4. Risk Management

Risk	Potential Impact	Mitigation Strategy
Ollama model compatibility issues	Core feature failure	Test multiple models, prepare fallback options
System resource limitations	Performance degradation	Optimize model selection, implement caching
API response quality	Poor user experience	Enhance prompt engineering, implement content filtering
Browser compatibility issues	UI/UX inconsistencies	Cross-browser testing, progressive enhancement

5. Quality Assurance

- Regular code reviews
- Automated testing for API endpoints
- UI testing across multiple devices and browsers
- User acceptance testing with sample audience
- Performance benchmarking

List of Resources

Human Resources

- **Project Manager**: Oversees project planning and execution
- **Backend Developer**: Python/Flask development, Ollama integration
- **Frontend Developer**: HTML/CSS/JavaScript implementation
- **UX/UI Designer**: Interface design and user experience

- **QA Tester**: Quality assurance and testing

Technical Resources

Development Environment

- **IDE**: Visual Studio Code
- **Version Control**: Git/GitHub
- **Local Environment**: Python 3.8+, virtual environment
- **Collaboration Tools**: Slack, Trello

Software Requirements

- **Backend**: Python 3.8+, Flask 2.x
- **AI Model**: Ollama with DeepSeek-r1:14b model
- **Frontend**: HTML5, CSS3, JavaScript (ES6+)
- **Frontend Libraries**:
  - Bootstrap 5.x
  - jQuery 3.6.0
  - Font Awesome 6.4.0
  - Animate.css 4.1.1

Hardware Requirements

- **Development**: Standard development machine (8GB+ RAM recommended)
- **Deployment**: Server with minimum 8GB RAM, preferably 16GB for optimal model performance
- **Storage**: Minimum 10GB for model storage and application code

API Requirements

- **Ollama API**: Local API running on port 11434
- **Model Specifics**: DeepSeek-r1:14b (or compatible alternative)

Documentation

- Project vision and scope document

- Technical specification
- API documentation
- User guide
- Developer onboarding document
- Deployment instructions

Testing Resources

- Testing framework for Python
- Cross-browser testing tools
- Mobile device testing
- Performance monitoring tools

Section Summary

This comprehensive documentation establishes a solid foundation for the Turkish Joke Generator project, outlining its scope, planning, and resource requirements in a professional manner. The project represents a unique intersection of cultural preservation and technological innovation by leveraging local AI capabilities to celebrate Turkish humor.

The careful planning and resource allocation detailed in this document create a framework for successful implementation while mitigating potential risks. By adhering to the outlined approach, the development team will be positioned to deliver a high-quality application that meets its objectives while demonstrating how thoughtful application of technology can preserve cultural traditions and create engaging digital experiences.

The Turkish Joke Generator serves as a model for implementing localized AI applications in non-English languages, addressing the challenge of language model performance for regional content while establishing a technical blueprint for similar cultural preservation initiatives.