# Voice Testing Guide

Complete guide for testing the voice functionality in Multilingual Mandi.

## Overview

The voice pipeline consists of three main components:

1. **SARVAM STT** - Converts audio to text (Speech-to-Text)
2. **OpenRouter AI** - Extracts intent and parameters from text
3. **Kisaan Bot** - UI component that orchestrates the pipeline

## Prerequisites

### 1. Node.js Installation

Ensure Node.js is installed and available in your PATH:

```
node --version
npm --version
```

### 2. API Keys Configuration

Check that API keys are configured in `backend/.env`:

```
SARVAM_API_KEY=sarvam-key
OPENROUTER_API_KEY=openrouter-key
```

Paras

### 3. Install Dependencies

```
cd backend
npm install
```

## Testing Methods

### Method 1: Complete Pipeline Test (Recommended)

Test the entire voice pipeline with a single command:

```
node test-voice-complete.js test/sample_add_listing.m4a
```

This will:

1. Send audio file to SARVAM STT
2. Get transcription
3. Send transcription to OpenRouter
4. Extract intent and parameters
5. Display complete results

**Expected Output:**

```
================================================================================
STEP 1: SARVAM Speech-to-Text
================================================================================

 SARVAM API Key found
 API URL: https://api.sarvam.ai
 Audio file: test/sample_add_listing.m4a
 File size: 45.23 KB
 Content type: audio/mp4

 Sending audio to SARVAM STT API...
 SARVAM Response received
{
  "transcript": "□□□□ 100 □□□□ □□□□□ □□□□□ □□"
}

 Transcription successful!
 Transcribed Text: □□□□ 100 □□□□ □□□□□ □□□□□ □□

================================================================================
STEP 2: OpenRouter Intent Extraction
================================================================================

 OpenRouter API Key found
 API URL: https://openrouter.ai/api/v1
 Model: qwen/qwen3-vl-32b-instruct

 Query: "□□□□ 100 □□□□ □□□□□ □□□□□ □□"

 Sending to OpenRouter for intent extraction...
 OpenRouter Response received
{
  "intent": "create_listing",
  "cropType": "wheat",
  "quantity": "100 kg",
  "price": null,
  "location": null,
  "qualityTier": null,
  "confidence": "high"
}
```

Paras

```
 Intent Extracted Successfully!
   Intent: create_listing
   Crop: wheat
   Quantity: 100 kg
   Price: N/A
   Location: N/A
   Quality: N/A
   Confidence: high

================================================================================
 PIPELINE TEST COMPLETE
================================================================================
```

## Method 2: Individual Component Tests

### Test SARVAM STT Only

```
node backend/test-sarvam-standalone.js test/sample_add_listing.m4a
```

### Test OpenRouter Only

```
node backend/test-openrouter-standalone.js "□□□□ 100 □□□□ □□□□□ □□□□□
□□"
```

## Method 3: Test with Kisaan Bot UI

1. Start the backend server:

Paras

```
cd backend
npm start
```

2. Start the frontend:

```
cd frontend
npm run dev
```

3. Open browser to http://localhost:3000
4. Click the Kisaan Bot icon (bottom right)
5. Click the microphone button
6. Speak your query in Hindi or English
7. Check the console for detailed logs

# Audio File Requirements

## Supported Formats

- WAV (`.wav`)
- MP3 (`.mp3`)
- M4A (`.m4a`)
- OGG (`.ogg`)
- WebM (`.webm`)

## Recommendations

- **Sample Rate**: 8000 Hz or higher
- **Channels**: Mono or Stereo
- **Duration**: 1-30 seconds
- **Language**: Hindi (primary), English (supported)
- **Quality**: Clear audio without background noise

## Test Audio Samples

The repository includes a test audio file:

- `test/sample_add_listing.m4a` - Hindi query about selling wheat

# Intent Types

The system recognizes these intents:

## 1. price_query

User wants to know crop prices

- Example: "□□□□□ □□ □□□ □□□□ □□?"
- Example: "What is the price of tomato?"

Paras

## 2. create_listing

User wants to sell/list a product

- Example: "□□□□ 100 □□□□ □□□□□ □□□□□ □□"
- Example: "I want to sell 100 kg wheat"

## 3. search_listings

User wants to search/buy products

- Example: "□□□□ □□□□□ □□□□□□ □□"
- Example: "Show me rice listings near Delhi"

## 4. make_offer

User wants to make an offer on a listing

- Example: "□□□ 50 □□□□□ □□□□□ □□□□ □□□□ □□□□□ □□□"
- Example: "I want to offer 50 rupees per kg"

### 5. general_help

General questions

- Example: "□□ □□□□ □□□ □□□□ □□?"
- Example: "How does this work?"

## Extracted Parameters

The AI extracts these parameters when mentioned:

- **cropType**: wheat, rice, tomato, onion, potato, cotton, etc.
- **quantity**: Amount with unit (e.g., "100 kg", "5 quintal")
- **price**: Price mentioned (number only)
- **location**: Place name (e.g., "Delhi", "Mumbai")
- **qualityTier**: premium, standard, or basic

## Troubleshooting

### SARVAM API Issues

**Error: API key not configured**

```
□ SARVAM_API_KEY not configured in backend/.env
```

Solution: Add valid API key to `backend/.env`

Paras

**Error: Request timeout**

```
□ Request timed out - check your internet connection
```

Solution: Check internet connection, try smaller audio file

**Error: Invalid audio format**

```
□ Unsupported audio format
```

Solution: Convert audio to supported format (WAV, MP3, M4A)

### OpenRouter API Issues

**Error: API key not configured**

```
 OPENROUTER_API_KEY not configured in backend/.env
```

Solution: Add valid API key to `backend/.env`

**Error: Response is not valid JSON**

```
⚠ Response is not valid JSON
```

Solution: Check model configuration, ensure model supports JSON responses

**Error: Rate limit exceeded**

```
Status: 429
```

Solution: Wait and retry, check API quota

### Kisaan Bot UI Issues

**Issue: Microphone not working**

- Check browser permissions for microphone access
- Use HTTPS or localhost (required for microphone API)
- Check browser console for errors

**Issue: No response after recording**

- Check backend server is running
- Check browser console for API errors
- Verify API keys are configured

Paras

**Issue: Mock response shown**

- This means the API call failed
- Check backend logs for error details
- Verify API keys and internet connection

## API Response Examples

### SARVAM STT Response

```
{
  "transcript": "□□□□ 100 □□□□ □□□□□ □□□□□ □□",
  "language": "hi",
```

```
  "duration": 3.5
}
```

OpenRouter Intent Response

```
{
  "intent": "create_listing",
  "cropType": "wheat",
  "quantity": "100 kg",
  "price": null,
  "location": null,
  "qualityTier": null,
  "confidence": "high"
}
```

Kisaan Bot API Response

```
{
  "transcription": "□□□□ 100 □□□□ □□□□□ □□□□□ □□",
  "intent": "create_listing",
  "cropType": "wheat",
  "quantity": "100 kg",
  "price": null,
  "location": null,
  "qualityTier": null,
  "confidence": "high",
  "response": "I can help you create a listing for 100 kg of wheat.
Would you like to proceed?"
}
```

Paras

## Performance Benchmarks

Typical response times:

- **SARVAM STT**: 2-5 seconds (depends on audio length)
- **OpenRouter Intent**: 1-3 seconds
- **Total Pipeline**: 3-8 seconds

## Next Steps

After successful testing:

1. Test with different audio samples
2. Test with different languages
3. Test error handling (invalid audio, network issues)
4. Test UI integration

5. Test with real users

## Additional Resources

- SARVAM AI Documentation
- OpenRouter Documentation
- Web Speech API