Setup Guide - Auto Dealership Voice Assistant

This guide will walk you through setting up the production-grade voice assistant system.

Prerequisites

- Python 3.8 or higher
- pip package manager
- (Optional) API keys for speech services

Step-by-Step Setup

1. Environment Setup

Create and activate a virtual environment:

```bash
# Create virtual environment
python -m venv venv

# Activate (Linux/Mac)
source venv/bin/activate

# Activate (Windows)
venv\Scripts\activate
```

2. Install Dependencies

```bash
```

```
pip install -r requirements.txt
```

3. Configure Environment Variables

Copy the example environment file:

```
bash

cp .env.example .env
```

Edit `.env` with your configuration:

```
env

# Minimum configuration for testing
OPENAI_API_KEY=sk-your-key-here
STT_PROVIDER=simulated
TTS_PROVIDER=simulated
```

4. Test Installation

Run a quick test:

```
bash

python -c "from config.settings import settings; print(settings.get_summary())"
```

5. Run the Application

```
bash

python main.py
```

Configuration Options

Speech-to-Text Providers

Option 1: Simulated (No API Key Required)

```env
STT_PROVIDER=simulated
```

- Good for: Testing, development
- Limitations: No actual speech recognition

Option 2: OpenAI Whisper

```env
STT_PROVIDER=whisper
OPENAI_API_KEY=sk-your-key-here
```

- Good for: High accuracy, multiple languages
- Cost: ~$0.006 per minute

Option 3: Google Cloud Speech-to-Text

```env
STT_PROVIDER=google
GOOGLE_CLOUD_PROJECT=your-project-id
GOOGLE_APPLICATION_CREDENTIALS=/path/to/credentials.json
```

- Good for: Real-time streaming, high accuracy
- Cost: ~$0.016 per minute

Option 4: Azure Speech Services

```env
STT_PROVIDER=azure
AZURE_SPEECH_KEY=your-key
AZURE_SPEECH_REGION=eastus
```

- Good for: Enterprise integration
- Cost: ~$1 per audio hour

Text-to-Speech Providers

Option 1: Simulated (No API Key Required)

```env
TTS_PROVIDER=simulated
```

- Good for: Testing, development
- Limitations: No actual audio output

Option 2: OpenAI TTS

```env
TTS_PROVIDER=openai
OPENAI_API_KEY=sk-your-key-here
```

- Good for: Natural-sounding voices, quick setup
- Cost: ~$15 per 1M characters

Option 3: ElevenLabs

```env
TTS_PROVIDER=elevenlabs
ELEVENLABS_API_KEY=your-key
```

- Good for: Ultra-realistic voices
- Cost: Varies by plan

Option 4: Google Cloud Text-to-Speech

```env
TTS_PROVIDER=google
GOOGLE_CLOUD_PROJECT=your-project-id
GOOGLE_APPLICATION_CREDENTIALS=/path/to/credentials.json
```

- Good for: High quality, many voice options
- Cost: ~$16 per 1M characters

Production Deployment

1. Security

- Never commit `.env` file
- Use environment-specific configurations

- Rotate API keys regularly
- Implement rate limiting

2. Monitoring

Enable logging:

```env
LOG_LEVEL=INFO
DEBUG=False
```

Monitor logs:

```bash
tail -f dealership_assistant.log
```

3. Database Backup

Regularly backup your data:

```bash
cp data/bookings.json data/backups/bookings_$(date +%Y%m%d).json
```

4. Performance Optimization

- Use connection pooling for APIs
- Implement caching for frequent queries
- Consider async processing for STT/TTS

Troubleshooting

Issue: "ModuleNotFoundError"

Solution:

```bash
pip install -r requirements.txt
```

Issue: "OpenAI API Error"

Solution:

- Verify API key in `.env`
- Check API key has sufficient credits
- Ensure correct model name

Issue: "No such file or directory"

Solution:

```bash
# Create necessary directories
mkdir -p data output
```

Issue: Speech services not working

Solution:

- Verify API keys are correct
- Check provider is properly set in `.env`

- For testing, use `simulated` provider

Next Steps

1. Customize `data/vehicle_knowledge_base.json` with your inventory
2. Adjust business hours in settings
3. Integrate with your CRM system
4. Deploy to production server

Support

For issues or questions:

- Check the logs: `dealership_assistant.log`
- Review error messages
- Verify configuration settings

Production Checklist

- [ ] All API keys configured
- [ ] Database backup strategy in place
- [ ] Logging properly configured
- [ ] Error handling tested
- [ ] Security measures implemented
- [ ] Performance monitoring enabled
- [ ] Documentation updated