Enhance Ollama Installation with Complete Model Downloads

Overview

This PR enhances the installation script to automatically download and verify **all required AI models** during the Ollama installation phase, ensuring the application is fully functional immediately after installation.

Problem Statement

The current installation script (`install.sh`) only downloads one model (`llama3.2:3b`) and does so in the background without proper verification. This leads to:

1. **Incomplete AI Setup**: Missing models required by different parts of the application
2. **Silent Failures**: Background downloads can fail without user awareness
3. **Delayed Functionality**: Users must manually download missing models later
4. **Poor User Experience**: No progress indicators or troubleshooting guidance

Current Behavior

```
# Old code - only downloads one model in background
ollama pull llama3.2:3b >> "$LOG_FILE" 2>&1 &
local pull_pid=$!
# Shows dots while waiting
wait $pull_pid
```

Solution

New `download_ollama_models()` Function

A comprehensive model download function that:

Downloads All 4 Required Models

1. **llama3.2:3b** - Primary model (2GB)
   - Used by: Enhanced Chat API, Tool Chat API, Log Analysis API
   - Files: `src/app/api/ai/enhanced-chat/route.ts` , `src/app/api/ai/tool-chat/route.ts` , `src/app/api/ai-assistant/analyze-logs/route.ts`

2. **phi3:mini** - Lightweight model (2.3GB)
   - Used by: General Chat API
   - Files: `src/app/api/chat/route.ts`

3. **llama2** - Backup model (3.8GB)
   - Used by: Device diagnostics
   - Reference: `install-local-ai.sh`

4. **mistral** - Fast model (4.1GB)
   - Used by: Quick queries
   - Reference: `install-local-ai.sh`

Total Download Size: ~12GB

Key Features

1. Sequential Downloads
- Downloads one model at a time to avoid resource contention
- Prevents network/disk I/O bottlenecks
- Ensures reliable completion

2. Retry Logic
- 3 attempts per model
- 5-second delay between retries
- Handles temporary network failures gracefully

3. Verification
- Checks if model already exists (skips download)
- Verifies successful download with `ollama list`
- Double-checks model availability after pull
- Tracks failed models for reporting

4. Progress Indicators

```
═══════════════════════════════════════
Model 2 of 4: phi3:mini
Purpose: Lightweight model for general chat interface
═══════════════════════════════════════
ℹ Downloading phi3:mini... (this may take several minutes)
✓ Model phi3:mini downloaded and verified successfully
```

5. Enhanced Service Readiness
- Replaced `sleep 5` with active polling of Ollama API
- 30-second timeout with proper error handling
- Verifies service is actually ready before downloading models

6. Comprehensive Error Handling
- Non-blocking failures (installation continues)
- Detailed troubleshooting guidance
- Lists all failed models at the end
- Provides manual recovery steps

7. User-Friendly Output

```
AI Features Ready:
  ✓ Enhanced Chat (llama3.2:3b)
  ✓ Tool Chat (llama3.2:3b)
  ✓ Log Analysis (llama3.2:3b)
  ✓ General Chat (phi3:mini)
  ✓ Device Diagnostics (llama2)
  ✓ Quick Queries (mistral)
```

Changes Made

1. Enhanced `install_ollama()` Function (Lines 354-386)

Before:

```
# Wait for Ollama to be ready
print_info "Waiting for Ollama to be ready..."
sleep 5

# Pull required models
print_info "Pulling required AI models (this may take a while)..."
ollama pull llama3.2:3b >> "$LOG_FILE" 2>&1 &
local pull_pid=$!
# Show progress dots
wait $pull_pid
print_success "AI models installed"
```

After:

```
# Wait for Ollama to be ready with active polling
print_info "Waiting for Ollama to be ready..."
local max_wait=30
local waited=0
while ! curl -s http://localhost:11434/api/tags > /dev/null 2>&1; do
    if [ $waited -ge $max_wait ]; then
        print_error "Ollama service failed to start within ${max_wait}s"
        return 1
    fi
    sleep 2
    waited=$((waited + 2))
done
print_success "Ollama service is ready"

# Download all required AI models
download_ollama_models
```

2. New `download_ollama_models()` Function (Lines 388-523)

Complete implementation with:
- Model definitions with descriptions
- Sequential download loop
- Retry logic (3 attempts per model)
- Verification checks
- Progress tracking
- Error collection and reporting
- Troubleshooting guidance
- Success summary

Benefits

1. **Complete Installation**: All AI features work immediately after installation
2. **Reliability**: Retry logic handles temporary network issues
3. **Transparency**: Users see exactly what's being downloaded and why
4. **Robustness**: Graceful handling of failures with recovery guidance

5. **User-Friendly**: Clear progress indicators and helpful messages
6. **Time Estimate**: Users know to expect 10-30 minutes for downloads
7. **Verification**: Ensures models are actually available before continuing

Error Handling

If Models Fail to Download

The installation continues with a warning:

```
⚠ Some models failed to download:
  ✗ mistral

⚠ The application will still work, but some AI features may be limited.

ℹ Troubleshooting steps:
  1. Check your internet connection
  2. Verify Ollama service is running: sudo systemctl status ollama
  3. Check available disk space: df -h
  4. Try manually pulling models later: ollama pull <model-name>
  5. View detailed logs: tail -f /tmp/sportsbar-install-*.log

ℹ You can continue with the installation. Failed models can be downloaded later.
```

Testing Recommendations

1. **Fresh Install**: Test on system without Ollama
2. **Existing Install**: Test with Ollama already installed
3. **Partial Models**: Test with some models already present
4. **Network Issues**: Test retry logic with simulated failures
5. **Disk Space**: Test behavior with insufficient space

Files Changed

- `install.sh` (Lines 341-523)
- Enhanced `install_ollama()` function
- New `download_ollama_models()` function
- Added comprehensive documentation
- `OLLAMA_MODEL_ENHANCEMENT.md` (New file)
- Complete documentation of changes
- Technical details and rationale
- Testing recommendations

Related Files

Models are used in:
- `src/app/api/chat/route.ts` - Uses phi3:mini
- `src/app/api/ai/enhanced-chat/route.ts` - Uses llama3.2:3b
- `src/app/api/ai/tool-chat/route.ts` - Uses llama3.2:3b

- `src/app/api/ai-assistant/analyze-logs/route.ts` - Uses llama3.2:3b
- `install-local-ai.sh` - Reference for model list

Backward Compatibility

✅ Fully backward compatible:
- Existing installations with models already downloaded will skip re-downloading
- Script continues even if some models fail
- No breaking changes to installation process
- All existing functionality preserved

Future Enhancements

Potential improvements for future versions:
- Parallel downloads with proper resource management
- Model size information before download
- Download progress bars (if Ollama API supports it)
- Configurable model list via environment variables
- Optional vs. required model categories

Commit Message

```
feat: enhance Ollama installation with complete model downloads

- Add download_ollama_models() function for comprehensive model management
- Download all 4 required models: llama3.2:3b, phi3:mini, llama2, mistral
- Implement retry logic (3 attempts) for network resilience
- Add verification checks for each downloaded model
- Improve service readiness check with active polling
- Provide clear progress indicators and troubleshooting guidance
- Ensure non-blocking failures with helpful error messages

This ensures all AI features are immediately functional after installation,
eliminating the need for manual model downloads.

Closes #[issue-number]
```

Installation Time Impact

Before: ~5 minutes (Ollama install only, models downloaded in background)
After: ~15-35 minutes (Ollama install + all models downloaded and verified)

The increased time is necessary to ensure complete functionality. Users are informed upfront with the message:

> "Downloading 4 AI models (this may take 10-30 minutes depending on your connection)..."

Verification

After installation completes, users can verify all models are installed:

```
ollama list
```

Expected output:

```
NAME              ID              SIZE      MODIFIED
llama3.2:3b       a80c4f17acd5    2.0 GB    X minutes ago
phi3:mini         4abea9e2f5e0    2.3 GB    X minutes ago
llama2:latest     78e26419b446    3.8 GB    X minutes ago
mistral:latest    f974a74358d6    4.1 GB    X minutes ago
```