Automatic TV Documentation System

Overview

The Automatic TV Documentation System is an intelligent feature that automatically fetches TV manuals and generates Q&A pairs when new TVs are discovered via CEC (Consumer Electronics Control). This creates a self-learning system where the AI assistant automatically becomes an expert on each TV model in your sports bar.

Features

1. Automatic Discovery Integration

- Triggers automatically when a new TV is discovered via CEC
- Extracts manufacturer and model information from CEC OSD name
- Runs documentation fetch in the background without blocking discovery

2. Intelligent Manual Search

- Searches the internet for official TV manuals (PDF format preferred)
- Uses multiple search queries to find the best sources
- Validates URLs before downloading
- Ranks results by relevance score

3. Manual Download & Storage

- Downloads PDF manuals or HTML documentation
- Saves to `docs/tv-manuals/` directory
- Uses safe filename format: `Manufacturer_Model_Manual.pdf`
- Handles edge cases (manual not found, download failures, etc.)

4. Content Extraction

- Extracts text content from PDF files using `pdf-parse`
- Converts HTML documentation to plain text
- Splits content into manageable chunks
- Identifies key sections (specifications, setup, troubleshooting, etc.)

5. Q&A Generation

- Generates template Q&A pairs for common questions
- Uses AI to create additional Q&A pairs from manual content
- Categorizes Q&A pairs by topic
- Saves to database for AI training

6. UI Integration

- Real-time status indicators for documentation fetching
- Manual fetch button for each TV model
- Summary statistics (total manuals, Q&A pairs)
- List of downloaded manuals with file sizes

Architecture

```
┌──────────────────────────────────────────────────────────────┐
│                    CEC Discovery Process                     │
└──────────────────────┬───────────────────────────────────────┘
                       │
                       ▼
┌──────────────────────────────────────────────────────────────┐
│              TV Detected (Brand + Model)                     │
└──────────────────────┬───────────────────────────────────────┘
                       │
                       ▼
┌──────────────────────────────────────────────────────────────┐
│           Auto-Fetch Documentation (Background)              │
│  ┌───────────────────────────────────────────────────────┐   │
│  │ 1. Search for Manual                                  │   │
│  │    - Multiple search queries                          │   │
│  │    - Rank by relevance                                │   │
│  │                                                       │   │
│  │ 2. Download Manual                                    │   │
│  │    - Try each source in order                         │   │
│  │    - Validate file size and type                      │   │
│  │    - Save to docs/tv-manuals/                         │   │
│  │                                                       │   │
│  │ 3. Extract Content                                    │   │
│  │    - Parse PDF or HTML                                │   │
│  │    - Split into chunks                                │   │
│  │    - Identify key sections                            │   │
│  │                                                       │   │
│  │ 4. Generate Q&A Pairs                                 │   │
│  │    - Template Q&A for common questions                │   │
│  │    - AI-generated Q&A from content                    │   │
│  │    - Save to database                                 │   │
│  └───────────────────────────────────────────────────────┘   │
└──────────────────────┬───────────────────────────────────────┘
                       │
                       ▼
┌──────────────────────────────────────────────────────────────┐
│              AI Knowledge Base Updated                       │
│         (Assistant now knows about this TV model)            │
└──────────────────────────────────────────────────────────────┘
```

File Structure

```
src/
├── lib/
│   └── tvDocs/
│       ├── types.ts              # TypeScript interfaces
│       ├── searchManual.ts       # Web search for manuals
│       ├── downloadManual.ts     # Download and save manuals
│       ├── extractContent.ts     # Extract text from PDFs
│       ├── generateQA.ts         # Generate Q&A pairs
│       └── index.ts              # Main service exports
│
├── app/
│   └── api/
│       ├── cec/
│       │   ├── fetch-tv-manual/
│       │   │   └── route.ts      # POST endpoint to fetch manual
│       │   └── tv-documentation/
│       │       └── route.ts      # GET endpoint for documentation list
│       ├── web-search/
│       │   └── route.ts          # Internal web search API
│       └── ai/
│           └── generate-qa/
│               └── route.ts      # AI Q&A generation API
│
└── components/
    └── TVDocumentationPanel.tsx  # UI component

docs/
└── tv-manuals/                   # Downloaded TV manuals
    ├── Samsung_UN55TU8000_Manual.pdf
    ├── LG_OLED55C1PUB_Manual.pdf
    └── Sony_XBR55X900H_Manual.pdf
```

API Endpoints

POST /api/cec/fetch-tv-manual

Fetch manual for a specific TV model.

Request Body:

```json
{
  "manufacturer": "Samsung",
  "model": "UN55TU8000",
  "forceRefetch": false
}
```

Response:

```
{
  "success": true,
  "manufacturer": "Samsung",
  "model": "UN55TU8000",
  "manualPath": "/path/to/Samsung_UN55TU8000_Manual.pdf",
  "documentationPath": "https://example.com/manual.pdf",
  "qaGenerated": true,
  "qaPairsCount": 25,
  "message": "Successfully fetched manual for Samsung UN55TU8000"
}
```

GET /api/cec/tv-documentation

Get all TV documentation records.

Response:

```
{
  "success": true,
  "documentation": [
    {
      "id": "Samsung-UN55TU8000",
      "manufacturer": "Samsung",
      "model": "UN55TU8000",
      "manualPath": "/path/to/manual.pdf",
      "fetchStatus": "completed",
      "qaGenerated": true,
      "qaPairsCount": 25,
      "createdAt": "2025-10-06T12:00:00Z",
      "updatedAt": "2025-10-06T12:05:00Z"
    }
  ],
  "totalManuals": 5,
  "totalQAPairs": 125,
  "manuals": [
    {
      "filename": "Samsung_UN55TU8000_Manual.pdf",
      "size": 2458624,
      "sizeFormatted": "2.34 MB"
    }
  ]
}
```

Usage

Automatic Mode (Recommended)

The system works automatically when you run CEC discovery:

1. Navigate to the CEC Discovery page
2. Click “Run Discovery” or “Discover All TVs”
3. When a TV is detected, documentation fetch starts automatically in the background
4. Check the TV Documentation panel to see progress
5. Q&A pairs are automatically added to the AI knowledge base

Manual Mode

You can also manually fetch documentation for specific TVs:

1. Navigate to the TV Documentation panel
2. Find the TV model you want to fetch documentation for
3. Click “Fetch Manual” button
4. Wait for the process to complete
5. View the generated Q&A pairs count

Re-fetching Documentation

If you want to update documentation for a TV:

1. Click “Re-fetch” button next to the TV model
2. The system will download the latest manual
3. New Q&A pairs will be generated and added

Configuration

Search Configuration

Edit `src/lib/tvDocs/searchManual.ts` to customize search behavior:

```
// Number of search results to try
const queries = [
  `${manufacturer} ${model} manual PDF`,
  `${manufacturer} ${model} user guide PDF`,
  `${manufacturer} ${model} instruction manual`,
]

// Relevance scoring weights
if (lowerTitle.includes('manual')) relevanceScore += 3
if (lowerTitle.includes('user guide')) relevanceScore += 3
if (isPDF) relevanceScore += 2
```

Q&A Generation Configuration

Edit `src/lib/tvDocs/generateQA.ts` to customize Q&A generation:

```
// Chunk size for content splitting
const chunks = splitContentIntoChunks(content, 2000)

// Maximum chunks to process (to avoid overload)
const maxChunks = Math.min(chunks.length, 10)
```

Edge Cases Handled

1. Manual Not Found

- System tries multiple search queries
- Falls back to alternative sources
- Records “not_found” status in database
- User can retry manually

2. Download Failures

- Validates URL before downloading
- Checks file size and content type
- Retries with next best source
- Logs detailed error messages

3. Multiple Models

- Uses exact model number from CEC OSD name
- Handles variations in model naming
- Prevents duplicate downloads

4. Large Files

- Validates file size (max 50MB)
- Streams downloads to avoid memory issues
- Shows progress in logs

5. Corrupted PDFs

- Validates PDF structure after download
- Checks for minimum content length
- Deletes invalid files automatically

Database Schema

The system uses the existing `QAPair` model:

```
model QAPair {
  id          String   @id @default(cuid())
  question    String
  answer      String
  category    String
  source      String   // e.g., "Samsung UN55TU8000 Manual"
  isActive    Boolean  @default(true)
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}
```

Performance Considerations

Background Processing

- Documentation fetch runs asynchronously
- Does not block CEC discovery process
- Uses Promise.catch() to handle errors gracefully

Rate Limiting

- 1-second delay between chunk processing
- Prevents overwhelming AI service
- Configurable in code

Caching

- Downloaded manuals are cached locally
- Checks for existing files before downloading
- Uses `forceRefetch` flag to override cache

Memory Management

- Streams large files instead of loading into memory
- Processes content in chunks
- Cleans up temporary data

Troubleshooting

Manual Not Downloading

Problem: Manual fetch fails with “No manual found online”

Solutions:
1. Check if the TV model name is correct in the database
2. Try searching manually for the manual online
3. Check network connectivity
4. Review search query patterns in `searchManual.ts`

Q&A Generation Fails

Problem: Manual downloads but Q&A pairs are not generated

Solutions:
1. Check if AI service is running
2. Verify `/api/ai/generate-qa` endpoint is working
3. Check manual content extraction (PDF might be corrupted)
4. Review logs for detailed error messages

PDF Extraction Errors

Problem: “Failed to extract PDF content” error

Solutions:
1. Verify `pdf-parse` package is installed
2. Check if PDF is password-protected
3. Try downloading the PDF manually to verify it’s valid
4. Check file permissions in `docs/tv-manuals/` directory

Duplicate Q&A Pairs

Problem: Same Q&A pairs appear multiple times

Solutions:
1. Check database for duplicate entries
2. Add unique constraint on question+source
3. Implement deduplication logic in `generateQA.ts`

Testing

Manual Testing

1. **Test Discovery Integration:**
   ```bash
   # Run CEC discovery
   curl -X POST http://localhost:3000/api/cec/discovery

# Check if documentation fetch started
tail -f logs/app.log | grep "TV Docs"
```

1. **Test Manual Fetch:**
   ```bash
   curl -X POST http://localhost:3000/api/cec/fetch-tv-manual \
     -H "Content-Type: application/json" \
     -d '{"manufacturer":"Samsung","model":"UN55TU8000"}'
   ```

2. **Test Documentation List:**
   ```bash
   curl http://localhost:3000/api/cec/tv-documentation
   ```

Automated Testing

Create test files in `__tests__/tvDocs/`:

```
// __tests__/tvDocs/searchManual.test.ts
import { searchTVManual } from '@/lib/tvDocs/searchManual'

describe('TV Manual Search', () => {
  it('should find manuals for Samsung TV', async () => {
    const results = await searchTVManual('Samsung', 'UN55TU8000')
    expect(results.length).toBeGreaterThan(0)
    expect(results[0].url).toBeDefined()
  })
})
```

Future Enhancements

1. Enhanced Search

- Integrate with manufacturer support APIs
- Use computer vision to extract model from TV photos
- Support for multiple languages

2. Advanced Q&A Generation

- Use more sophisticated AI models
- Generate troubleshooting flowcharts
- Create video tutorials from manual content

3. Knowledge Base Management

- Deduplicate similar Q&A pairs
- Rank Q&A pairs by usefulness

- Allow manual editing of Q&A pairs

4. Analytics

- Track which manuals are most accessed
- Monitor Q&A pair usage in AI responses
- Identify gaps in documentation

5. Integration

- Sync with manufacturer support portals
- Auto-update when new firmware is released
- Share knowledge base across multiple locations

Security Considerations

File Validation

- Validates file types before saving
- Checks file sizes to prevent DoS
- Sanitizes filenames to prevent path traversal

Content Filtering

- Removes scripts and malicious content from HTML
- Validates PDF structure
- Limits content size for Q&A generation

API Security

- Rate limiting on fetch endpoints
- Authentication required for manual operations
- Input validation on all parameters

Deployment

Prerequisites

- Node.js 18+ with npm
- Prisma database configured
- Write permissions to `docs/tv-manuals/` directory

Installation Steps

1. **Install Dependencies:**
   `bash`
   ```
      npm install pdf-parse cheerio axios
   ```

2. **Create Manuals Directory:**
   `bash`
   ```
      mkdir -p docs/tv-manuals
      chmod 755 docs/tv-manuals
   ```

3. **Run Database Migrations:**
   `bash`
   ```
      npx prisma migrate dev
   ```

4. **Build Application:**
   `bash`
   `  npm run build`

5. **Start Server:**
   `bash`
   `  npm start`

Environment Variables

No additional environment variables required. The system uses existing configuration.

Monitoring

Monitor the system using logs:

```
# Watch for documentation fetch activity
tail -f logs/app.log | grep "TV Docs"

# Check for errors
tail -f logs/app.log | grep "ERROR.*TV Docs"

# Monitor disk usage
du -sh docs/tv-manuals/
```

Support

For issues or questions:
1. Check the troubleshooting section above
2. Review logs in `logs/app.log`
3. Check GitHub issues
4. Contact the development team

License

This feature is part of the Sports Bar TV Controller system and follows the same license.

Last Updated: October 6, 2025
Version: 1.0.0
Author: Sports Bar AI Assistant Team