Codebase Indexing System

Overview

The AI Assistant now has access to your entire codebase through an intelligent indexing system. This allows the AI to provide accurate, context-aware help by referencing actual code files when answering questions.

Features

1. Automatic Code Scanning

- Scans all source code files in the project
- Supports: `.ts`, `.tsx`, `.js`, `.jsx`, `.json`, `.md`, `.prisma`
- Excludes: `node_modules`, `.next`, `build`, and other build artifacts
- Skips large files (> 1MB) to maintain performance

2. Smart Indexing

- Tracks file changes using content hashing
- Only re-indexes modified files
- Marks deleted files as inactive
- Stores file metadata (path, type, size, last modified)

3. Intelligent Search

- Full-text search across all indexed files
- Relevance scoring based on:
- File name matches (highest priority)
- File path matches
- Content matches (with frequency counting)
- Returns most relevant files with code snippets

4. Database Storage

The system uses a new `IndexedFile` model in Prisma:

```prisma
model IndexedFile {
  id            String   @id @default(cuid())
  filePath      String   @unique
  fileName      String
  fileType      String
  content       String
  fileSize      Int
  lastModified DateTime
  lastIndexed  DateTime
  hash          String
  isActive      Boolean
  metadata      String?
  createdAt     DateTime
  updatedAt     DateTime
}
```

Usage

Syncing the Codebase

1. Navigate to the AI Assistant page: `/ai-assistant`
2. Click the “Sync Codebase” button
3. Wait for the indexing to complete
4. View statistics about indexed files

Asking Questions

Once the codebase is indexed, you can ask the AI questions like:

- “How does the Matrix configuration work?”
- “Show me the audio processor control implementation”
- “Where is the CEC power control logic?”
- “How do I add a new API route?”
- “What’s the structure of the Prisma schema?”

The AI will search the indexed codebase and provide answers with references to specific files.

API Endpoints

POST `/api/ai-assistant/index-codebase`

Indexes the entire codebase.

Response:

```json
{
  "success": true,
  "stats": {
    "totalFiles": 150,
    "indexed": 10,
    "updated": 5,
    "skipped": 135,
    "deactivated": 0
  }
}
```

GET `/api/ai-assistant/index-codebase`

Gets current index statistics.

Response:

```json
{
  "success": true,
  "stats": {
    "totalFiles": 150,
    "totalSize": 2500000,
    "filesByType": [
      { "type": "typescript", "count": 80 },
      { "type": "typescript-react", "count": 50 },
      { "type": "json", "count": 15 },
      { "type": "markdown", "count": 5 }
    ],
    "lastIndexed": "2025-10-02T12:00:00Z"
  }
}
```

POST `/api/ai-assistant/search-code`

Searches the indexed codebase.

Request:

```json
{
  "query": "matrix configuration",
  "fileTypes": ["typescript", "typescript-react"],
  "maxResults": 10
}
```

Response:

```json
{
  "success": true,
  "results": [
    {
      "id": "...",
      "filePath": "src/components/MatrixConfiguration.tsx",
      "fileName": "MatrixConfiguration.tsx",
      "fileType": "typescript-react",
      "score": 25,
      "snippets": ["...code snippet..."],
      "lastModified": "2025-10-01T10:00:00Z"
    }
  ],
  "totalResults": 5
}
```

Enhanced AI Knowledge Base

The AI now uses an enhanced knowledge base system (`ai-knowledge-enhanced.ts`) that combines:

1. **Documentation Context**: From uploaded PDFs and markdown files
2. **Codebase Context**: From indexed source code files

When you ask a question, the AI:
1. Searches documentation for relevant information
2. Searches codebase for relevant files

3. Combines both contexts
4. Provides a comprehensive answer with file references

Performance Considerations

- **Initial Indexing**: May take 30-60 seconds for large codebases
- **Incremental Updates**: Only changed files are re-indexed (much faster)
- **File Size Limits**: Files > 1MB are skipped
- **Content Truncation**: Very long files are truncated in AI context to prevent token limits

Database Migration

To add the IndexedFile model to your database:

```
# Apply the migration
npx prisma db push

# Or create a new migration
npx prisma migrate dev --name add_indexed_files
```

Troubleshooting

Indexing Fails

- Check file permissions in the project directory
- Ensure database is accessible
- Check logs for specific error messages

AI Not Using Codebase

- Verify files are indexed (check stats on AI Assistant page)
- Try re-syncing the codebase
- Check that `useCodebase: true` is set in API calls

Missing Files

- Check if files are in excluded directories
- Verify file extensions are supported
- Check if files exceed size limit (1MB)

Future Enhancements

Planned improvements:
- Automatic background syncing
- Webhook-based updates on file changes
- Vector embeddings for semantic search
- Support for more file types
- Configurable exclusion patterns
- File content preview in search results