RAG System Installation Instructions

Files to Copy

Copy these files from `/home/claude/` to your project:

1. **documents_search.ex** → `lib/green_man_tavern/documents/search.ex`
2. **claude_client.ex** → `lib/green_man_tavern/ai/claude_client.ex`
3. **character_context.ex** → `lib/green_man_tavern/ai/character_context.ex`
4. **character_live_updated.ex** → `lib/green_man_tavern_web/live/character_live.ex` (REPLACE existing file)

Installation Steps

1. Create Directories

bash

```bash
cd ~/Projects/HyperArk-2025
mkdir -p lib/green_man_tavern/ai
```

2. Copy Files

bash

```bash
# Copy document search
cp /home/claude/documents_search.ex lib/green_man_tavern/documents/search.ex

# Copy AI modules
cp /home/claude/claude_client.ex lib/green_man_tavern/ai/claude_client.ex
cp /home/claude/character_context.ex lib/green_man_tavern/ai/character_context.ex

# Backup and replace CharacterLive
cp lib/green_man_tavern_web/live/character_live.ex lib/green_man_tavern_web/live/character_live.ex.backup
cp /home/claude/character_live_updated.ex lib/green_man_tavern_web/live/character_live.ex
```

3. Add HTTPoison Dependency

Edit `mix.exs` and ensure HTTPoison is in your deps (it might already be there):

elixir

```elixir
defp deps do
  [
    # ... other deps ...
    {:httpoison, "~> 2.0"}
  ]
end
```

Then run:

bash

```bash
mix deps.get
```

4. Set Your Anthropic API Key

Option A: Environment Variable (Recommended)

bash

```bash
export ANTHROPIC_API_KEY="your-api-key-here"
```

Add to your `~/.bashrc` or `~/.zshrc` to make permanent:

bash

```bash
echo 'export ANTHROPIC_API_KEY="your-api-key-here"' >> ~/.bashrc
source ~/.bashrc
```

Option B: Application Config Edit `config/dev.exs`:

elixir

```elixir
config :green_man_tavern,
  anthropic_api_key: "your-api-key-here"
```

⚠️ **NEVER commit API keys to git!** Add to `.gitignore` if using config file.

5. Compile and Test

bash

```bash
# Compile the new modules
mix compile

# Start the server
mix phx.server
```

6. Test the System

1. Navigate to http://localhost:4000
2. Click on a character (e.g., "The Student")
3. Ask a question like: "How do I start composting?"
4. The character should respond with information from your 44 PDFs!

What Changed

Removed (MindsDB)

- `GreenManTavern.MindsDB.Client.query_agent/3`
- `GreenManTavern.MindsDB.ContextBuilder`
- `GreenManTavern.MindsDB.MemoryExtractor`

Added (Direct Claude API)

- `GreenManTavern.Documents.Search` - Search your PDF knowledge base
- `GreenManTavern.AI.ClaudeClient` - Direct Anthropic API calls
- `GreenManTavern.AI.CharacterContext` - Build character prompts and context

Modified

- `CharacterLive` - Now uses Claude API instead of MindsDB

How It Works

1. **User asks a question** → Character LiveView receives message
2. **Search knowledge base** → Find relevant chunks from your 44 PDFs
3. **Build context** → Format chunks + character personality
4. **Call Claude API** → Send to Anthropic with full context
5. **Return response** → Character responds in their unique voice

Troubleshooting

"Anthropic API key not configured"

- Make sure you've set the `ANTHROPIC_API_KEY` environment variable
- Restart your terminal/server after setting it

No search results

- Check that documents were processed: `psql green_man_tavern_dev -c "SELECT COUNT(*) FROM document_chunks;"`
- Should show ~16,000 chunks

Character doesn't respond

- Check logs for errors: Look for "CLAUDE API" messages
- Verify API key is valid
- Check your Anthropic account has credits

Search returns irrelevant results

- Current implementation uses simple keyword matching
- For better results, implement vector embeddings (future enhancement)

Next Steps (Optional Enhancements)

1. **Vector Search** - Add pgvector for semantic search
2. **Conversation Memory** - Store and retrieve past conversations in prompts
3. **Streaming Responses** - Stream Claude's response word-by-word
4. **Response Caching** - Cache common questions to save API calls
5. **Multi-document Citations** - Show which documents were used