Project 3: Smart Knowledge Repository

Project 3: Smart Knowledge Repository **1**
Objective (Why?) 2
Core Requirements (Must-have for Initial 2-Day Project) 2
Development Approach: Milestone-Based Progression 3
Technical Specifications 5
Measurable Goals & Review Template Compliance 5
Task Tracking & Project Management Integration 6
Sample Interactions 8
Stretch Goals (For this Project or as a Foundation for Project 4) 9
Deliverables 10
Evaluation Rubric 10
Testing Scenarios 11
Quick Start Resources 12

Objective (Why?)

Build an intelligent knowledge base that scrapes specific website sections, stores information locally, and answers questions within scope in just 2 days. This accelerated timeline builds efficiently on your Projects 1-2 foundation. You will practice:

- Structured Data Extraction: Scraping and organizing specific information
- Data Storage & Retrieval: Building a simple knowledge database
- Context-Aware AI: Creating scope-limited AI responses
- Intelligent Querying: Matching user questions to relevant data

Core Requirements (Must-have for Initial 2-Day Project)

Layer	Requirement
Data Collection	➢ Targeted Web Scraping: Scrape a *pre-defined* leadership team page (e.g., amzur.com/leadership-team/). ➢ Extract structured data: name, role, bio, photo, contact. ➢ Store data in a local SQLite database.
Knowledge Base	➢ Keyword-Based Search: Implement a local SQLite database with Full-Text Search (FTS5) for efficient keyword retrieval. ➢ Data should be categorized by role or department.

Chat Interface	➢ Streamlit Application: Use `st.chat_input()` and `st.chat_message()` for conversations. ➢ Use `st.sidebar` for filtering and `st.tabs()` for different views (Chat, Browse Profiles).
Scope Management	➢ Rule-Based Scope Detection: Implement a function to check if a user's question contains keywords related to the knowledge base (e.g., "who", "CEO", "team"). ➢ Provide helpful "I don't know" responses for out-of-scope questions.

Development Approach: Milestone-Based Progression

Philosophy: Focus on deliverable quality and comprehensive review compliance rather than rigid timelines. Each milestone must pass all relevant review templates from our Templates folder before proceeding.

Milestone 1: Data Collection & Knowledge Storage

Deliverables:

- Structured web scraping for leadership profile data
- JSON/SQLite knowledge storage system with proper schema
- Data validation and normalization pipeline
- Basic search functionality with keyword matching
- Profile browsing interface with categorization

Review Requirements (Must Pass to Proceed):

- Security Review: Safe data scraping, input validation, storage security
- Code Quality Review: Clean data modeling and storage architecture
- Performance Review: Efficient data extraction and retrieval

Milestone 2: Intelligent Query & Scope Management

Deliverables:

- Scope-aware chatbot with domain detection
- Context-aware AI responses using stored knowledge
- Out-of-scope question handling with helpful suggestions
- Advanced search with relevance scoring
- Chat interface with profile data integration

Review Requirements (Must Pass to Proceed):

- AI Integration Review: Effective scope management and context awareness
- Architecture Review: Clean separation of knowledge and AI components
- Code Quality Review: Maintainable search and query systems

Milestone 3: Production Features & Knowledge Management

Deliverables:

- Advanced knowledge management (add, update, delete profiles)
- Multi-view interface (Chat, Browse, Data Management, Analytics)
- Export/import functionality for knowledge base
- Comprehensive documentation and testing procedures
- Production deployment preparation

Review Requirements:

- Architecture Review: Scalable knowledge management system
- AI Integration Review: Production-ready scope-aware AI
- Security Review: Complete data security and access control
- Code Quality Review: Final production-ready code quality

Milestone Progression Rules:

- Cannot advance to next milestone without passing all review requirements
- Flexible timing allows for learning at individual pace
- Quality gates ensure each milestone meets professional standards

- Mentor support available for concept clarification and review failures

Technical Specifications

Data Structure

- Design JSON schema for storing profile information including name, role, bio, and contact details
- Implement data validation and normalization for consistent profile storage

Knowledge Base Implementation

- Build simple text-based search functionality using keyword matching and relevance scoring
- Create data categorization system for efficient profile retrieval by role, department, or name

Measurable Goals & Review Template Compliance

Primary Objectives

- Knowledge Management Excellence: 90%+ accurate scope detection and responses
- Data Architecture Quality: Pass Architecture Review with 8.5/10+ score
- AI Integration Sophistication: Pass AI Integration Review with 8.5/10+ score
- Search Performance: Sub-second search response times with relevant results
- Code Quality Standards: Pass Code Quality Review with 8/10+ score

Performance Standards

- Search Response Time: < 3 second for keyword searches
- Scope Detection Accuracy: 90%+ correct in-domain/out-of-domain classification

- Knowledge Retrieval: Relevant results in top 3 for 85% of queries
- Data Integrity: 100% consistency in profile data storage and retrieval

Task Tracking & Project Management Integration

Project 3 - Smart Knowledge Repository

Epic ID: P3-KNOWLEDGE
Priority: High
Dependencies: Project 2 completion

Milestone 1: Data Collection & Knowledge Storage

Feature 3.1: Knowledge Data Infrastructure

Task ID: P3-M1-DATA
Priority: Critical
Dependencies: None

Sub-tasks:

- P3-M1-DATA-01: Design knowledge schema
  - Description: Create JSON/SQLite schema for profile data
  - Acceptance Criteria: Normalized schema with proper relationships
- P3-M1-DATA-02: Implement data collection service
  - Description: Leadership profile scraping with structured extraction
  - Acceptance Criteria: Extract name, role, bio, contact, photo URLs
- P3-M1-DATA-03: Build search functionality
  - Description: Keyword-based search with relevance scoring
  - Acceptance Criteria: Full-text search with ranking algorithm

Milestone 2: AI Intelligence & Scope Management

Feature 3.2: Intelligent Query System

Task ID: P3-M2-AI
Priority: High
Dependencies: P3-M1-DATA completion

Sub-tasks:

- P3-M2-AI-01: Scope detection engine
  - Description: Domain classification for in/out-of-scope queries
  - Acceptance Criteria: 90%+ accuracy in scope detection
- P3-M2-AI-02: Context-aware response system
  - Description: Knowledge-based AI responses with profile context
  - Acceptance Criteria: Relevant, accurate responses using stored data
- P3-M2-AI-03: Chat interface implementation
  - Description: Streamlit chat interface with profile integration
  - Acceptance Criteria: User-friendly conversation flow

Milestone 3: Production Features & Management

Feature 3.3: Knowledge Management System

Task ID: P3-M3-MGMT
Priority: Medium
Dependencies: P3-M2-AI completion

Sub-tasks:

- P3-M3-MGMT-01: Data management interface
  - Description: Add, update, delete profile functionality
  - Acceptance Criteria: Complete CRUD operations for profiles
- P3-M3-MGMT-02: Multi-view interface
  - Description: Chat, Browse, Data Management, Analytics views
  - Acceptance Criteria: Seamless navigation between all views
- P3-M3-MGMT-03: Export/import functionality
  - Description: Knowledge base backup and restore capabilities
  - Acceptance Criteria: JSON export/import with data validation

Phase Progression Requirements

Project 3 → Project 4 Advancement Requirements

Mandatory Review Template Compliance:

- Architecture Review: Minimum 8.5/10 score (knowledge system design)
- AI Integration Review: Minimum 8.5/10 score (scope management)
- Security Review: Minimum 8.5/10 score (data security)
- Code Quality Review: Minimum 8.0/10 score (maintainable systems)

Project 4 Preparation Requirements

- Full-Stack Fundamentals: Learn React basics and component architecture
- Database Design: PostgreSQL fundamentals and relationship modeling
- Authentication Patterns: JWT, bcrypt, and security best practices
- API Design: RESTful API principles and FastAPI advanced features

Sample Interactions

In-Scope Questions

```
None
User: "Who is the CEO?"
Bot: "Bala Nemani is the founding partner and CEO of Amzur Technologies. An
entrepreneurial spirit with over 25 years of experience, Bala’s experience in multiple
domains spans across Information Technology, Engineering and Energy, with a
strong focus on product development, IT innovation, and business strategy. As a
serial entrepreneur who has started and led several companies, Bala has
established various partnerships with seed-stage, mid-market, and enterprise
companies. He is based in Tampa, FL, and is known for his extensive work in helping
businesses cut costs and improve process efficiencies with emerging technologies
for enterprise application development, enterprise resource planning (ERP), and
```

wireless IoT solutions. He is also a firm believer in CSR (Corporate Social Responsibility) and contributes actively to the community through a multitude of programs that he leads at Amzur."

User: "Tell me about the Leadership Team in Amzure"
Bot: "The Bot should return as
Bala Nemani President – Group CEO
Ganna Vadlamaani - CEO, Growth Markets
Sam Velu - Director, Workforce Solutions
Gururaj Gokak - Director, Finance
Muralidhar Veerapaneni - Director, Operations
Rakesh Mantrala - Head of Marketing & Corporate Communications
Sunil Kodi - Head of Pre Sales, Partner Management & Customer Success"

Out-of-Scope Questions

None
User: "What's the weather today?"
Bot: "I only have information about our leadership team and can't help with weather questions. I can tell you about our executives, their backgrounds, roles, or experience. Try asking something like 'Who are the company founders?' or 'Tell me about the CTO.'"

User: "What's the stock price?"
Bot: "I don't have access to financial or stock information. My knowledge is focused on our leadership team profiles. I can share details about our executives' backgrounds, roles, and experience instead."

Stretch Goals (For this Project or as a Foundation for Project 4)

- Semantic Search: Upgrade from keyword search to vector-based semantic search using sentence transformers.
- Retrieval-Augmented Generation (RAG): Implement a formal RAG pipeline for more contextually accurate answers.
- Auto-Discovery: Enhance the scraper to automatically find new leadership profile pages.
- Data Visualization: Create a leadership organization chart.
- Auto-Updates: Schedule re-scraping to keep data current.

Deliverables

1. GitHub Repository Link (public or invite @mentor)
2. Live Demo with sample Q&A session
3. KNOWLEDGE_DEMO.md - Include:
    - Sample extracted data (5+ profiles)
    - 10+ example Q&A interactions (in-scope and out-of-scope)
    - Screenshots of the application
    - List of data sources used
4. Technical_Learnings.md

Evaluation Rubric

Criterion	Points	Details
Data Extraction	30 pts	Successfully scrapes structured profile data Handles multiple profiles automatically Data quality and completeness
Knowledge Search	25 pts	Accurate search and retrieval Relevant results for queries Handles name/role variations

Scope Management	20 pts	Correctly identifies in-scope questions Graceful out-of-scope responses Helpful redirection suggestions
AI Integration	15 pts	Context-aware LLM responses Natural conversation flow Accurate information presentation
Code Quality	10 pts	Clean project structure Good documentation Error handling

Testing Scenarios

Data Collection Testing

- Target: https://amzur.com/leadership-team/
- Expected: Extract all leadership profiles with relevant information
- Validate: Names, roles, bios, contact info, images

Question Categories to Test

In-Scope Questions:

- "Who is the CEO?"
- "Tell me about the CTO"
- "List all VPs"
- "What's Sarah's background?"
- "Who leads the engineering team?"

Out-of-Scope Questions:

- "What's the weather?"
- "How's the stock doing?"
- "What's for lunch?"
- "Current news updates?"

- "Company financials?"

Quick Start Resources

- Web Scraping: https://realpython.com/beautiful-soup-web-scraper-python/
- JSON Handling: https://docs.python.org/3/library/json.html
- Text Search: https://docs.python.org/3/library/re.html
- SQLite (optional): https://docs.python.org/3/library/sqlite3.html