

# Project Charter: Custom ChatGPT-Style Assistant for Company Documentation

Department: AI Solutions | Project Lead: [Your Name]  
Version: 1.0 | Last Updated: [Date]

---

## 1. Business Case

**Problem:**  
Employees waste 20+ hours/month searching internal docs (HR policies, technical manuals, meeting notes). Existing intranet search yields 42% irrelevant results (Q3 2024 internal audit).

**Solution:**  
AI assistant that:

- ✓ Answers natural language questions about company docs
- ✓ Cites sources for compliance
- ✓ Learns from user feedback

**ROI:**

- Estimated **30% reduction** in employee search time
- **\$250K/year** saved in operational overhead

---

## 2. Technical Specifications

### Core Features

Feature	Implementation
Document	PDF/PPT/Word → Text extraction (PyMuPDF,
Ingestion	Tesseract OCR)

Knowledge Base	Vectorized storage (FAISS) + metadata tagging
Query Engine	Hybrid: GPT-3.5-turbo + custom fine-tuned DistilBERT
User Interface	Slack/Teams integration + web dashboard (Gradio)

Data Sources

```
markdown
- 📁 **HR**: Employee_Handbook_2024.pdf, Benefits_Guide.pptx
- 📁 **Engineering**: API_Specs.md, Incident_Reports_2023.zip
- 📁 **Leadership**: All_Hands_Recordings/ (Whisper transcripts)
```

System Architecture

```
Diagram
Code
```

---

3. Project Roadmap

- Phase 1 (Weeks 1-2):
- Ingest sample docs (HR handbook + 10 engineering specs)
  - Benchmark baseline accuracy (target: 85% correct answers)
- Phase 2 (Weeks 3-4):
- Implement feedback loop (👍/👎 button to train reward model)
  - Add audit logging for compliance
- Phase 3 (Week 5+):
- Dockerize for on-prem deployment
- 

4. Compliance & Security

- 🛡️ **Data Isolation**: All docs remain on company servers (no OpenAI training)

- 📄 **Approvals Required:** Legal (Doc Usage Policy v3.2), Infosec (Form AI-114)
- 

## 5. Example Use Cases

### HR Scenario:

User: "How many sick days do I get?"

Assistant:

"You accrue 8 sick days annually (page 12, [HR Handbook](#)). Unused days roll over up to 16. [🔍 See Related: Leave Policy]"

### Engineering Scenario:

User: "Show error codes for API 500 responses"

Assistant:

"From [API\\_Specs.md](#):

- 500.1: Database timeout
  - 500.3: Auth token expired
- [⚠️ 3 recent incidents logged]"
- 

## 6. Mock Performance Metrics

```
python
{
  "accuracy": 0.87, # Tested on 200 QA pairs
  "latency": 1.2s, # P95 response time
  "cost": "$0.02/query", # AWS + OpenAI estimated
  "user_satisfaction": 4.6/5 # Pilot group survey
}
```

---

### Why This Works for Your Portfolio

1. **Realistic Structure:** Mirrors actual enterprise docs (shows you understand biz needs)
2. **Balanced Tech/Biz:** Proves you can bridge AI and stakeholder requirements
3. **Scalable Design:** Docker mention signals deployment awareness