**Finalized Project Plan: GenAI-based Document Navigator for ABB**

**Project Goal:**

Develop an AI-powered system that helps **ABB employees** quickly find, summarize, and extract insights from internal documents in **Azerbaijani and English**. The system will handle **machine-readable documents** (PDFs, Word, Excel) and **scanned documents** (using OCR), enabling **efficient document retrieval** and **automated insights extraction**.

**Core Features:**

1. **Document Ingestion and Parsing:**
   * **Upload**: Users can upload **PDFs**, **Word documents**, and **Excel files**.
   * **Text Extraction**:
     + For **machine-readable documents**, **pdfplumber** (for PDFs), **python-docx** (for Word), and **openpyxl** (for Excel) will be used for **text extraction**.
     + For **scanned PDFs** or image-based documents, **OCR** (using **Tesseract** or **PaddleOCR**) will be applied to **extract text** from images or scanned pages.
   * **Metadata**: Metadata (e.g., document title, tags, and keywords) will be captured and stored to improve searchability.
2. **Bilingual Support (Azerbaijani & English):**
   * The system supports **bilingual queries** (both **Azerbaijani** and **English**) for searching, summarizing, and extracting insights.
   * **Semantic Search**: Implement **semantic search** using embeddings (e.g., **OpenAI** embeddings or **BGE**) to handle **meaning-based** queries in either language.
   * **Text Summarization**: Use models like **T5** or **BART** for **summarizing documents** in both languages.
3. **Search & Retrieval:**
   * Users can search for documents by **name** (e.g., “Give me the report of Week 5”) or by **querying content** (e.g., "What changed in the AML policy in 2024?").
   * **Semantic Search**: The system uses **vector embeddings** (generated by models like **OpenAI embeddings**) to retrieve documents based on the **meaning** of the query, not just keyword matching.
4. **Summarization and Insights Extraction:**
   * **Summarization**: Documents will be automatically summarized using **Hugging Face models** (e.g., **T5**, **BART**) to generate concise summaries of long reports or policy documents.
   * **Insights Extraction**: The system will extract key **data points** from documents (e.g., **financial data**, **policy updates**) and generate **actionable insights**.
   * **Visualization**: If the document contains **financial data** or **tables**, the system will automatically generate **visualizations** (e.g., **bar charts**, **line graphs**, **pie charts**) using **Plotly** or **Matplotlib**.
5. **User Interface (UI):**
   * **Streamlit UI** will be developed for users to interact with the system.
     + **Search bar** for querying documents in either **Azerbaijani or English**.
     + **File upload** for uploading new documents.
     + **Display** for showing **summaries**, **insights**, and **visualizations**.
     + **Bilingual support** in UI for switching between Azerbaijani and English.
6. **Security & Compliance:**
   * **Encryption**: All documents and embeddings will be **encrypted** both in **transit** and **at rest** to ensure data security.
   * **Role-Based Access Control (RBAC)**: Implement RBAC to control user access based on roles within the organization.
   * **Audit Logs**: Keep track of document uploads, updates, and searches for compliance and traceability.

**Technology Stack:**

* **Document Parsing**:
  + **pdfplumber** (for PDFs), **python-docx** (for Word), **openpyxl** (for Excel).
  + **Tesseract** or **PaddleOCR** for OCR (for scanned or image-based documents).
* **Embeddings**:
  + **OpenAI embeddings**, **BGE**, or **Sentence-BERT** for semantic search.
* **Text Summarization**:
  + **Hugging Face models**: **T5**, **BART**, or **GPT-4o-mini** for summarization and RAG-based question answering.
* **Vector Database**:
  + **FAISS** or **Milvus** for storing and querying document embeddings.
* **Visualization**:
  + **Matplotlib**, **Plotly** for generating charts from financial data.
* **Web Framework**:
  + **Streamlit** for building the user interface.
* **Security**:
  + **Encryption** for document storage and **RBAC** for access control.

**Project Workflow:**

1. **Document Upload**: Users upload documents (PDF, Word, Excel) or scanned images (PDF with images) through the **Streamlit UI**.
2. **Text Extraction**:
   * If the document is machine-readable (text-based), **extract text** directly using libraries like **pdfplumber**.
   * If the document is a **scanned image** (e.g., PDF with embedded images), **apply OCR** to extract the text.
3. **Vectorization**:
   * Convert the extracted text into **embeddings** using **OpenAI embeddings**, **BGE**, or **Sentence-BERT**. Store these embeddings in the **vector database** (**FAISS** or **Milvus**).
4. **Search & Query**:
   * Users can **search** for documents based on **content** (e.g., a specific policy update) or by document **name** (e.g., a specific report).
   * **Semantic Search** will retrieve relevant documents using **vector search** (based on meaning, not just keywords).
5. **Summarization & Insights**:
   * Once the document is retrieved, it will be **summarized** using **T5** or **BART**.
   * **Insights** (e.g., key statistics, financial data trends) will be extracted from the document.
6. **Visualization**:
   * If the document contains **tables** with financial data, the system will **generate graphs** (bar, line, pie charts) to visualize key metrics.
7. **Results Display**:
   * The results (summaries, insights, and visualizations) are displayed in the **Streamlit UI** for the user to view and interact with.

**Project Timeline (10-12 Days):**

* **Day 1-2**: Setup and environment configuration (VS Code, virtual environment, dependencies).
* **Day 3-5**: Document parsing, extraction, and chunking.
* **Day 6-7**: Implementing **vector search** and semantic retrieval with embeddings.
* **Day 8-9**: Integrating **RAG-based summarization** and **question answering** models.
* **Day 10-11**: **Streamlit UI** development and integration with backend.
* **Day 12**: Final testing, optimization, and deployment.

**Expected Outcomes:**

1. **Enhanced Efficiency**: ABB employees can quickly retrieve and analyze important documents.
2. **Improved Decision-Making**: Automated **summaries** and **insights** allow employees to make data-driven decisions more quickly.
3. **User-Friendly Interface**: A **bilingual** UI (supporting **Azerbaijani and English**) makes it easy for employees to interact with the system.
4. **Automated Insights**: **Financial data extraction** and **visualization** for easy understanding of trends and figures.
5. **Secure and Compliant**: **Role-based access control** and **encryption** ensure that only authorized personnel can access sensitive documents.

**Conclusion:**

The **GenAI-based Document Navigator** will be a **powerful tool** for ABB, improving productivity and decision-making by providing employees with quick access to important documents, automatic summarization, and **data-driven insights**. The system will be **bilingual**, secure, and easy to use, enabling ABB to efficiently handle internal documents and reports in both **Azerbaijani and English**.