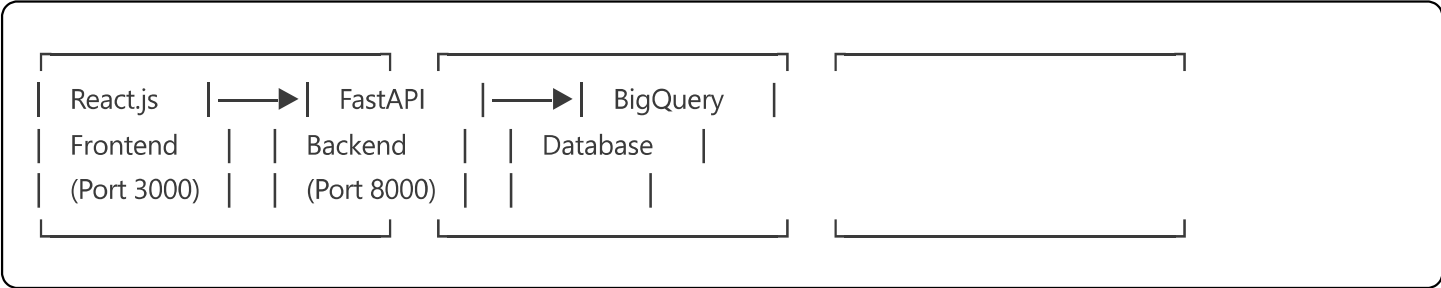


RAG Document Processing System - Setup Instructions

Overview

This system provides a complete RAG (Retrieval-Augmented Generation) solution for extracting business rules from BRD documents and stored procedures using React.js frontend, Python FastAPI backend, and Google BigQuery database.

Architecture



Prerequisites

System Requirements

- Node.js 18+ and npm
- Python 3.8+
- Google Cloud Platform account
- Git

Google Cloud Setup

1. Create a GCP project
2. Enable BigQuery API
3. Create a service account with BigQuery Admin role
4. Download service account key JSON file

Installation Steps

1. Clone and Setup Backend

```
bash
```

```
# Create project directory
mkdir rag-document-processor
cd rag-document-processor

# Create backend directory
mkdir backend
cd backend

# Create virtual environment
python -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt

# Create directory structure
mkdir services models temp_uploads
```

2. Configure Environment Variables

Create `.env` file in backend directory:

```
bash

# Copy the env_config content and update with your values
GCP_PROJECT_ID=your-actual-project-id
GOOGLE_APPLICATION_CREDENTIALS=path/to/your/service-account-key.json
BQ_DATASET_ID=rag_document_processing
# ... other configurations
```

3. Setup BigQuery Database

```
bash

# Install Google Cloud CLI
# https://cloud.google.com/sdk/docs/install

# Authenticate
gcloud auth login
gcloud config set project your-project-id

# Run the DDL script to create tables
bq query --use_legacy_sql=false < bigquery_schema.sql
```

4. Setup Frontend

```
bash
```

```
# From project root
```

```
cd ..
```

```
npx create-react-app frontend
```

```
cd frontend
```

```
# Install additional dependencies
```

```
npm install lucide-react
```

```
# Replace src/App.js with the React component
```

```
# Copy the RAG Interface component code to src
```