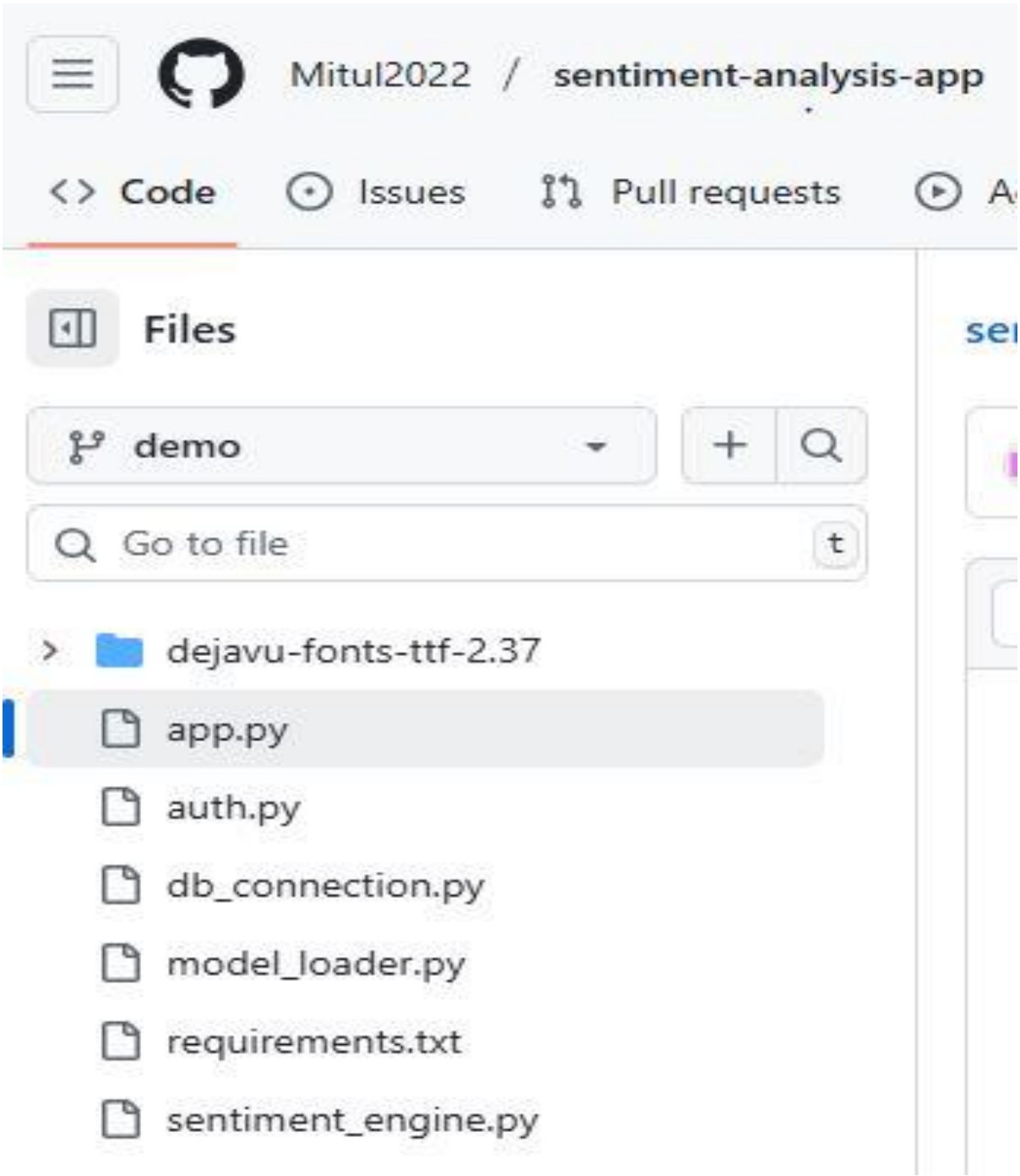
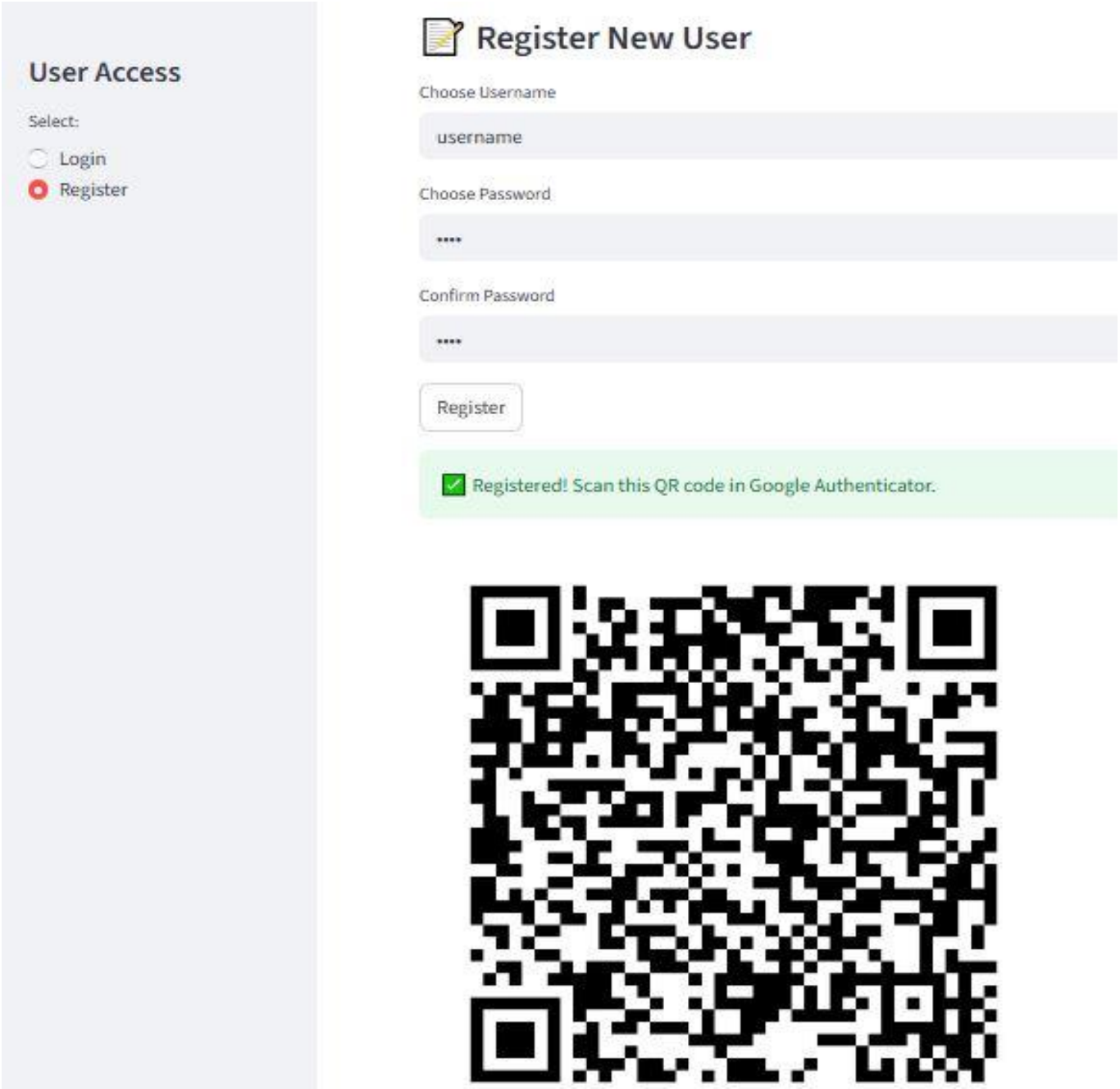
Github Link:- "https://github.com/Mitul2022/sentiment-analysis-app/blob/demo/app.py**"**

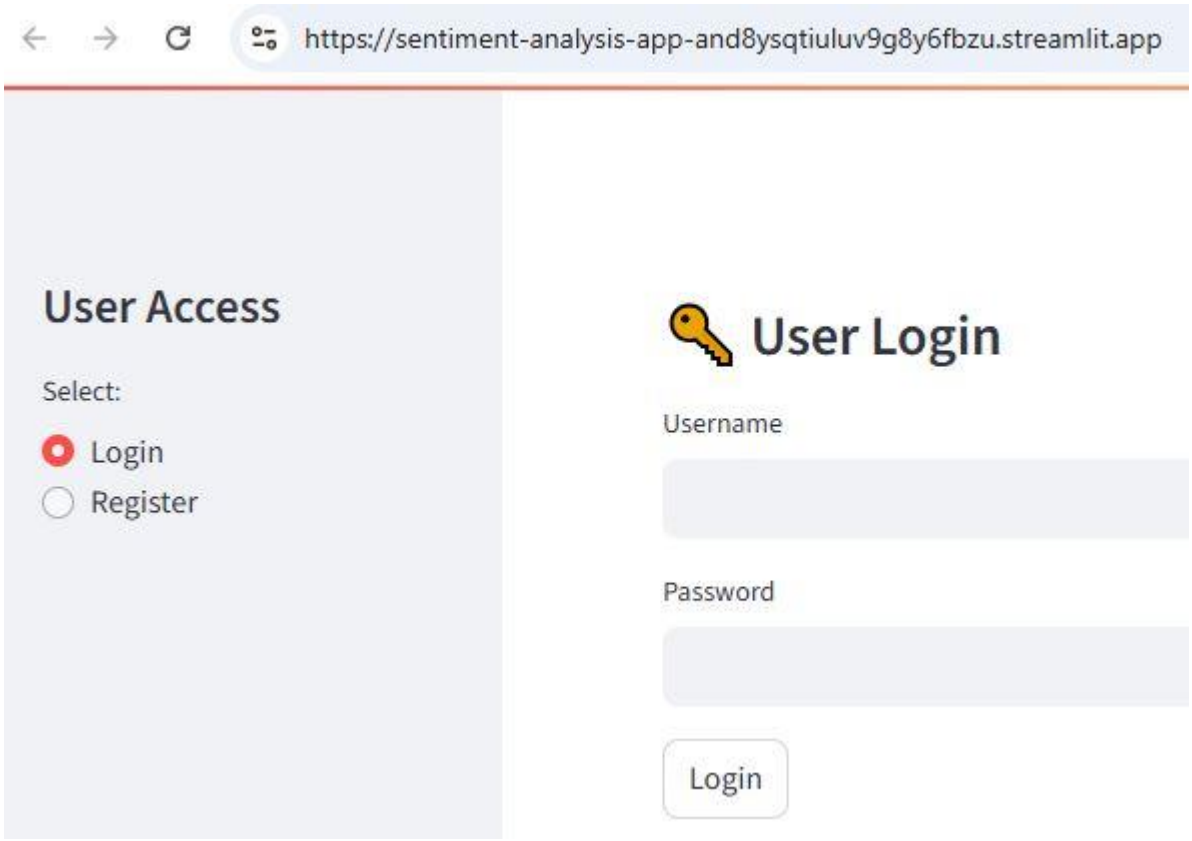


(Note:- Make Sure To select "sentiment-analysis-app" repository and inside that……Select "demo" branch. ("main" branch is containing time taking version…so select "demo" branch only). [both versions are available on streamlit])

Streamlit Link:- "https://sentiment-analysis-app-and8ysqtiuluv9g8y6fbzu.streamlit.app**/"**

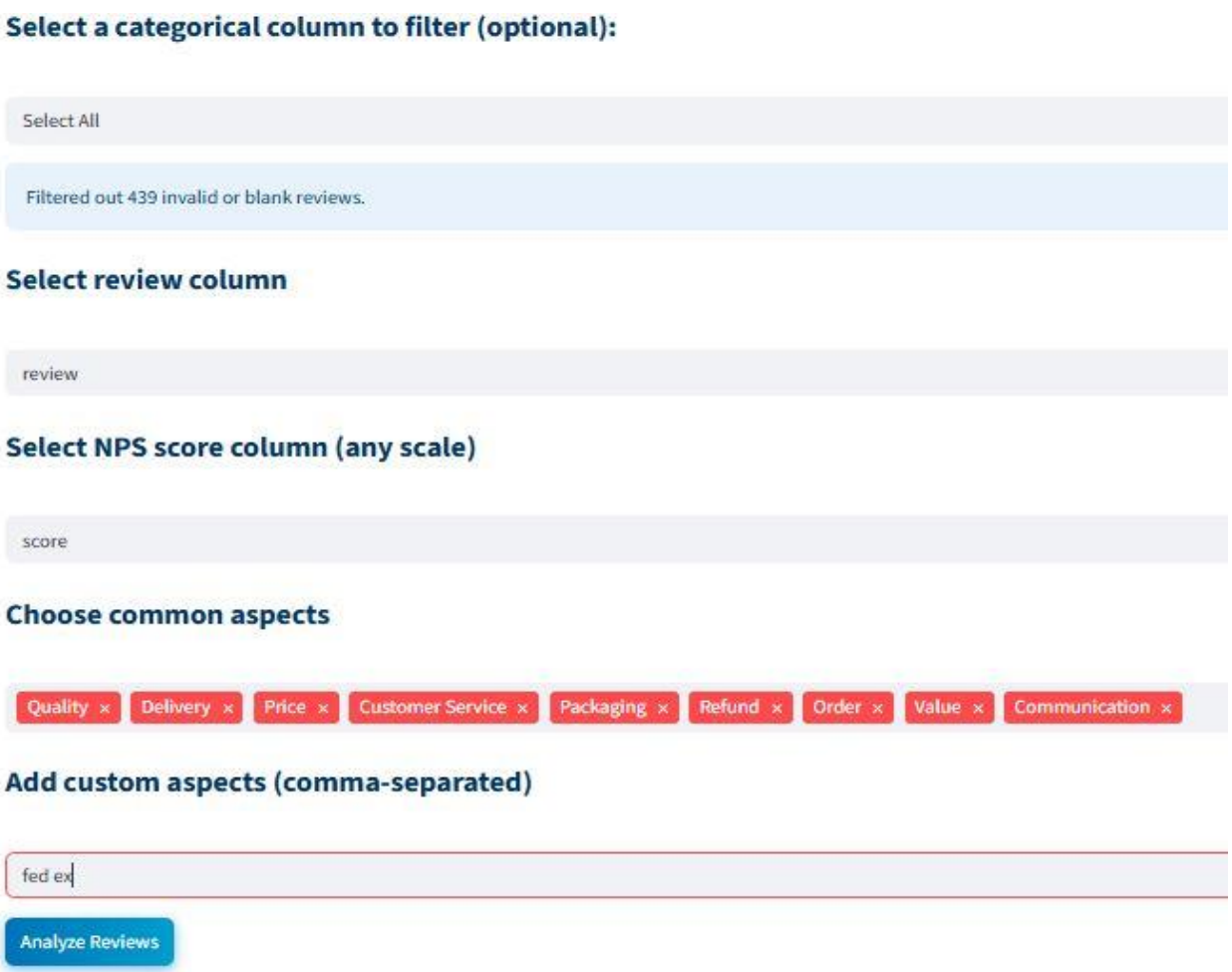


Register Yourself with appropriate Username and password. And then Scan the generated QR code from Google Authenticator App

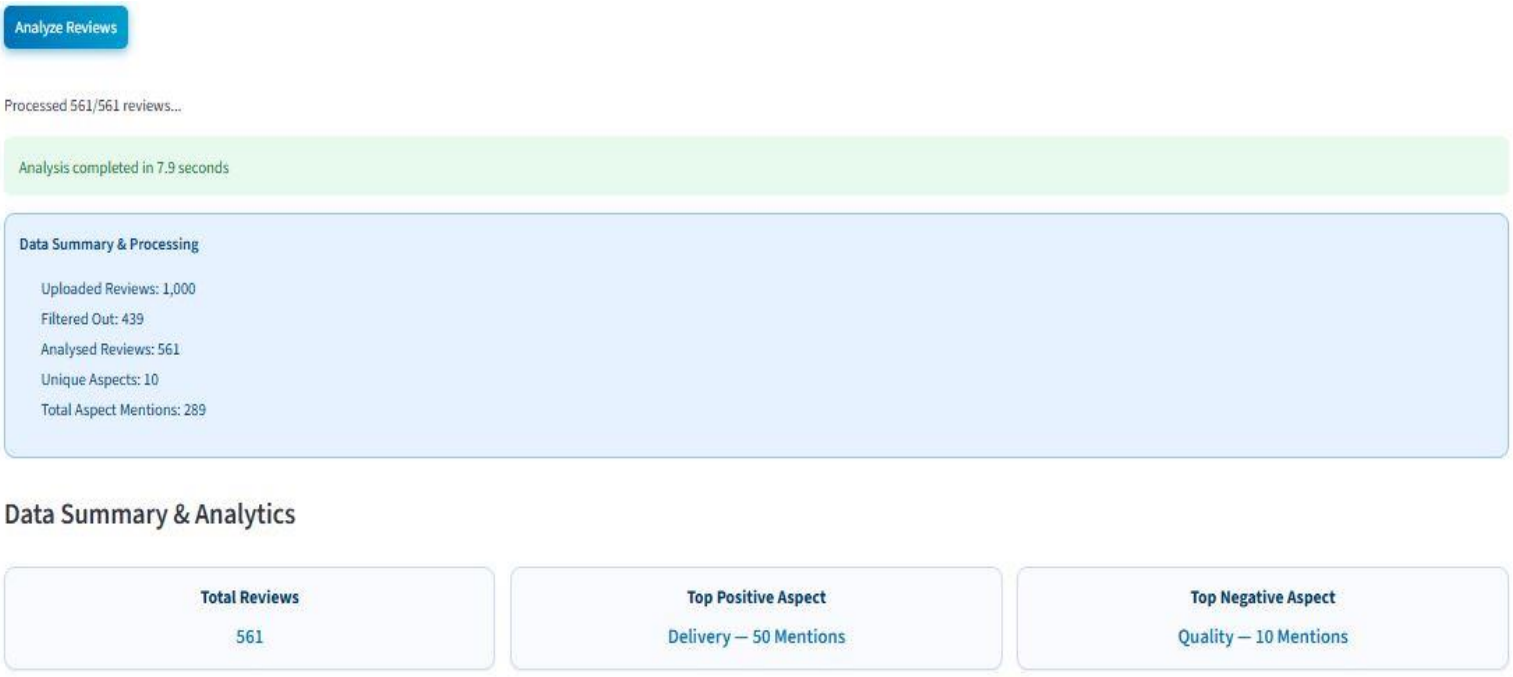


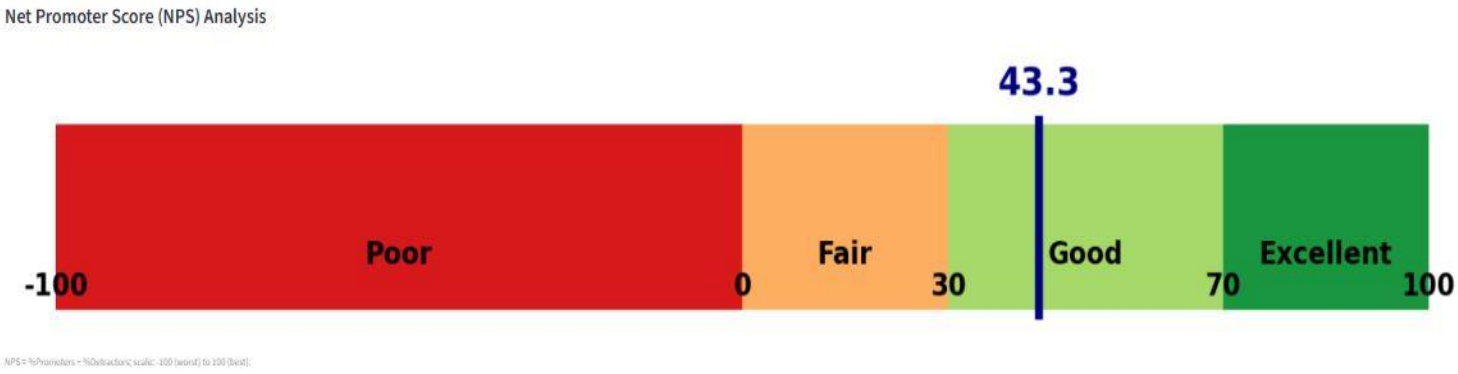
Fill Registered Credentials and Provide OTP generated on google authenticator App to login securely.

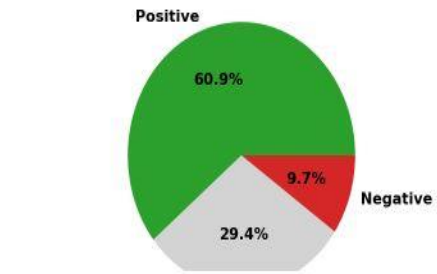
Upload review column containing “CSV or Excel File”



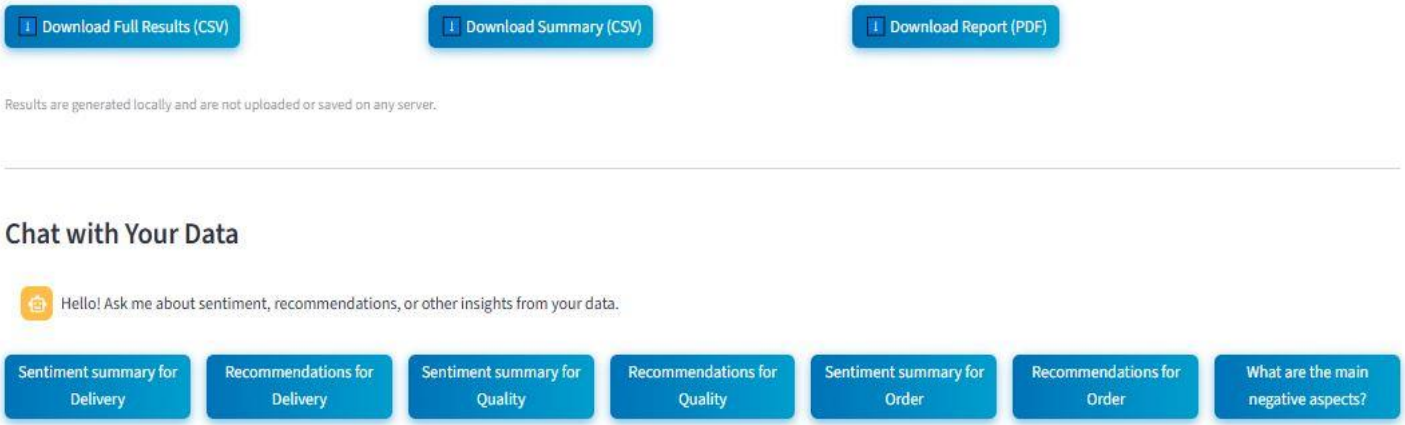
Follow Required Steps as per your Requirement. And Click “Analyze Reviews” to getting it start.







Likewise there will be all Analytical Visuals on the front end with the insights of Review Data.



And at the end of all 3 Download Buttons and the Data Generated Quick Insights Button Will be there for providing End To End detailed Descriptive Insights.

Review_ID	Review	Aspect	Aspect_Sentiment	Aspect_Context
1	Great flowers; good value	Value	Positive	Great flowers; good value
5	Beautiful flowers but problems with delivery. Flowers left at door, stayed there all day even th	Delivery	Neutral	Beautiful flowers but proble
10	The flowers were gorgeous. The price was reasonable, they arrived as promised and they wer	Price	Positive	The price was reasonable,
16	Loved the packaging. A few of the blooms came in brown/wilted The vase was a little flimsy.	Packaging	Positive	Loved the packaging.
26	Good service and the flowers are of exceptional quality	Quality	Positive	Good service and the flowe
28	Great flowers! Delivery and ordering was very easy.	Delivery	Positive	Delivery and ordering was v
30	I'm always really happy with the quality of the blooms when they arrive.	Quality	Positive	I'm always really happy wit
38	The price and quality. My mother inlaw told me the flowers are beautiful and the price was rig	Quality	Neutral	Most other flower delivery s
38	The price and quality. My mother inlaw told me the flowers are beautiful and the price was rig	Price	Neutral	My mother inlaw told me th
38	The price and quality. My mother inlaw told me the flowers are beautiful and the price was rig	Delivery	Neutral	Most other flower delivery s
39	Easy to order, price was excellent, love the "farm to delivery" aspect, flowers were beautiful a	Delivery	Positive	Easy to order, price was exc
39	Easy to order, price was excellent, love the "farm to delivery" aspect, flowers were beautiful a	Price	Positive	Easy to order, price was exc
39	Easy to order, price was excellent, love the "farm to delivery" aspect, flowers were beautiful a	Order	Positive	Easy to order, price was exc
44	Fresh flowers, wrapping, delivery on time	Delivery	Positive	Fresh flowers, wrapping, de

Aspect	Total Mentions	Positive	Neutral	Negative	Positive (%)	Neutral (%)	Negative (%)	Dominant Sentiment	Avg NPS	Promoters	Passives	Detractors
Delivery	77	50	23	4	64.94	29.87	5.19	Positive	8.3	56	14	7
Quality	65	41	14	10	63.08	21.54	15.38	Positive	7.55	43	9	13
Order	51	24	21	6	47.06	41.18	11.76	Positive	6.12	25	10	16
Price	36	25	10	1	69.44	27.78	2.78	Positive	8.78	28	6	2
Customer Service	29	19	5	5	65.52	17.24	17.24	Positive	6.41	16	5	8
Value	11	7	4	0	63.64	36.36	0	Positive	6.82	7	1	3
Packaging	8	7	1	0	87.5	12.5	0	Positive	8.62	6	1	1
Refund	6	1	5	0	16.67	83.33	0	Neutral	0.17	1	0	5
Communication	4	1	1	2	25	25	50	Negative	4.25	2	0	2
Fed Ex	2	1	1	0	50	50	0	Neutral	7.5	1	0	1

1st Download Button Will Provide Detailed analytics of Review Data. &

2nd Download Button Will Provide Summary of Analysis.

CUSTOMER REVIEW SENTIMENT ANALYSIS

EXECUTIVE SUMMARY & KEY INSIGHTS

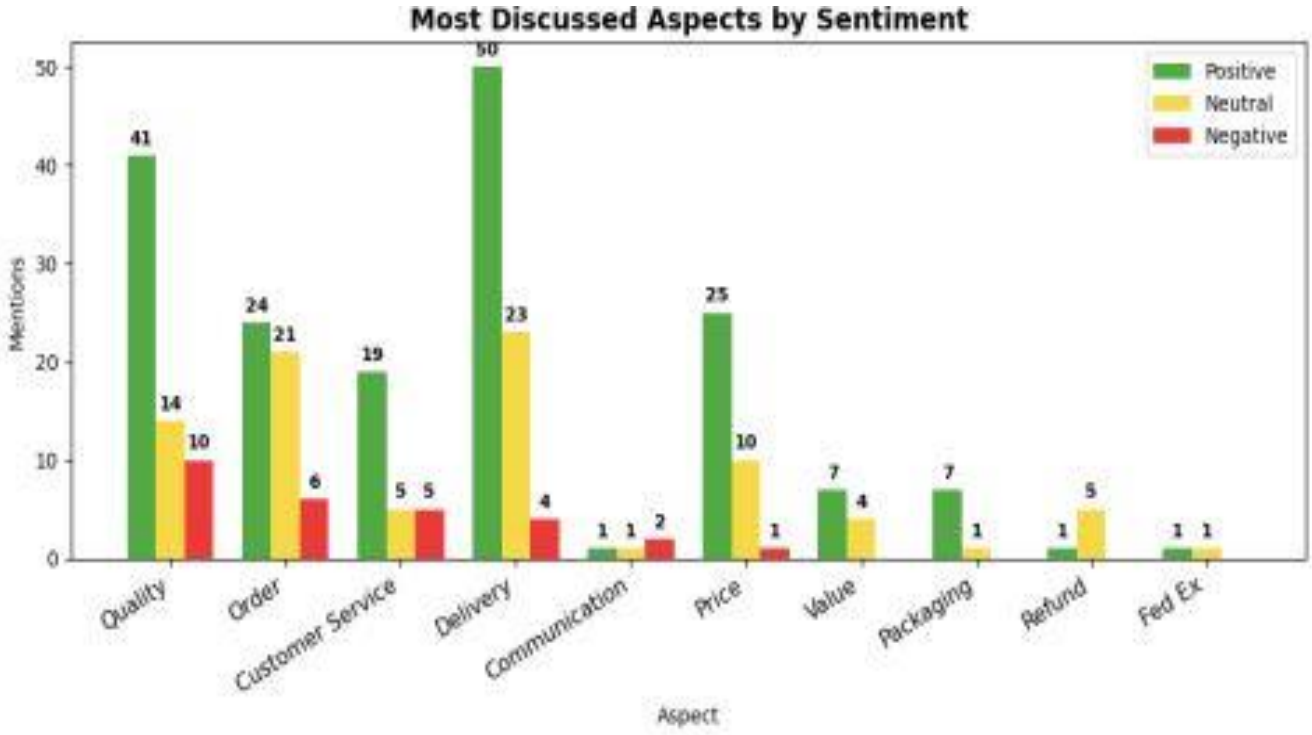
Data processing summary:
- Uploaded: 1,000 reviews
- Filtered out: 439 (blanks, undefined, N/A)
- Analysed: 561 valid reviews
- Unique aspects: 10
- Total mentions: 289

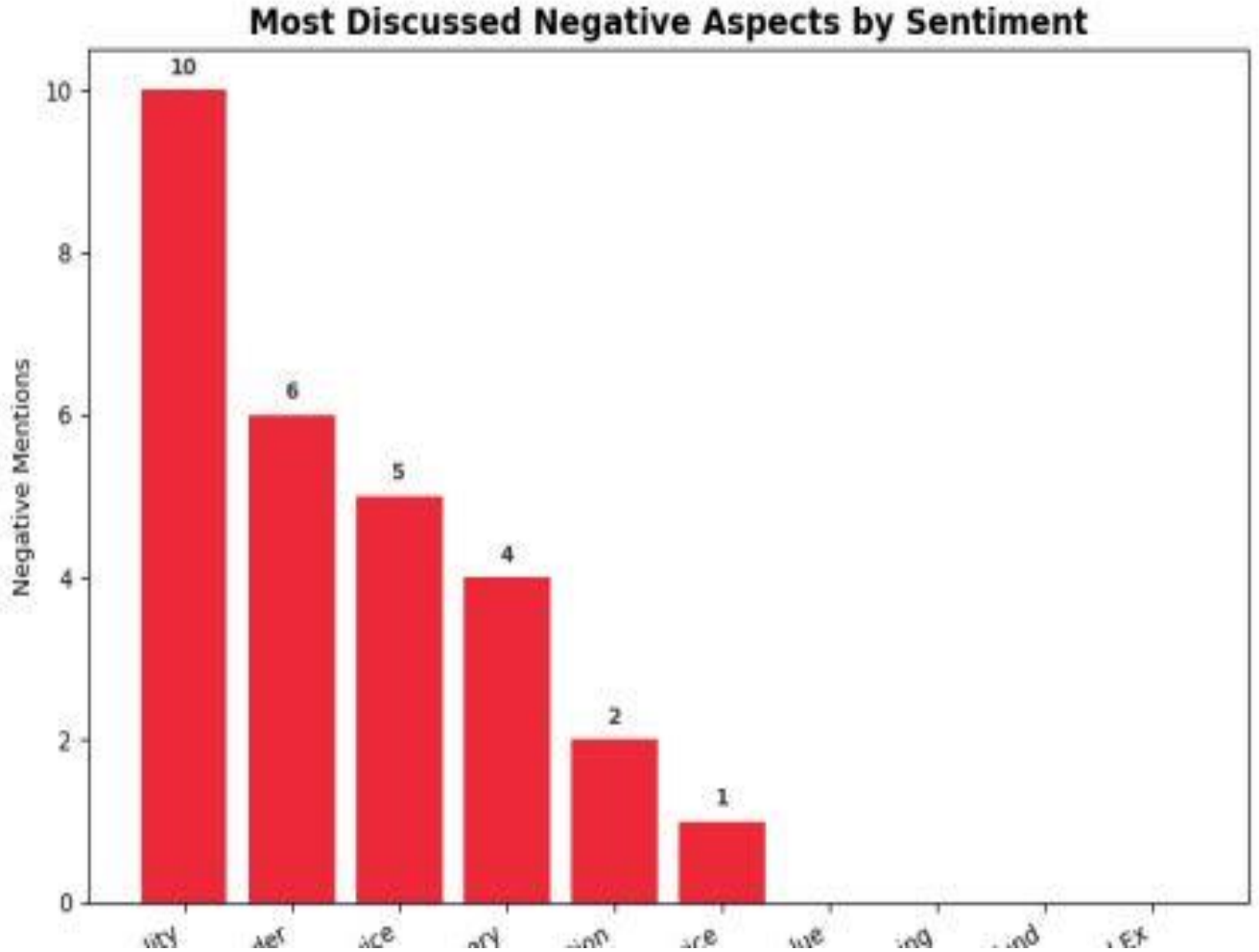
Top mentioned aspect: Delivery (77 mentions).

KEY METRICS & KPI OVERVIEW

Sentiment & NPS Score KPIs

KPI	Value
Total Mentions	289
Positive Mentions (%)	60.90%
Neutral Mentions (%)	29.41%
Negative Mentions (%)	9.69%
Average NPS Score	7.34
NPS Score (%)	43.3
Promoters	183
Detractors	58


Most Discussed Aspects by Sentiment
Positive
Neutral
Negative
Mentions
Aspect
Quality
Order
Customer Service
Delivery
Communication
Price
Value
Packaging
Refund
Fed Ex
41
14
10
24
21
6
19
5
5
50
23
4
1
1
2
25
10
1
7
4
7
1
1
5
1
1


Most Discussed Negative Aspects by Sentiment
Negative Mentions
10
6
5
4
2
1

RECENT NEGATIVE REVIEWS BY ASPECT

Delivery - Negative Reviews

1. No card. Delivery screwed up.
2. Poor delivery communication
3. I sent the sparrow (deluxe) to my grandma after her kitten unexpectedly passed away. She was so excited about the flowers and absolutely adored them. She said everyone at work and people who came into the store were in awe of her bouquet and taking photos of it themselves. I've ordered flowers on...
4. Delivery lost by shipping carrier. Will definitely recommend using local florists instead to avoid this.

Quality - Negative Reviews

1. The flowers I sent didn't look like the bouquet I chose. And the quality didn't look super great.
2. I spent $10 extra on a vase. The vase was of such poor quality, it leaked water constantly.
3. The worst quality flowers I have ever purchased. Completely different than what was pictured on the website. Truly an embarrassment to give on Valentine's Day. I had to scramble last minute to go buy flowers from a real florist.
4. The quality was so poor. Looked nothing like the pictures of flowers on your web page. Im ashamed I paid over 5.00.
5. Flower quality was very poor. Over half of flowers delivered dead. Only offered store credit in return.

ACTIONABLE RECOMMENDATIONS

Delivery - Recommendations

1. Customers frequently mention 'online flower company', 'ordered flowers online', 'taking photos', 'shipping carrier', 'said everyone', 'poor communication', 'never use', 'many companies', 'always disappoint', 'absolutely adored', 'flowers', 'work', 'way', 'thank', 'store', 'sparrow', 'shows', 'sent', 'screwed', 'quality', 'put', 'people', 'lost', 'happier', 'grandma', 'excited', 'display', 'deluxe', 'care', 'card', 'came', 'bouquet', 'awe', 'avoid' related to Delivery. Investigate root causes immediately.
2. Prioritize resolving 'online flower company' to improve Delivery experience.

Quality - Recommendations

1. Customers frequently mention 'scramble last minute', 'offered store credit', 'leaked water constantly', 'looked nothing like', 'look super great', 'go buy flowers', 'flowers delivered dead', 'look like', 'worst flowers', 'web page', 'real florist', 'im ashamed', 'ever purchased', 'completely different', '10 extra', 'flowers', 'website', 'vase', 'valentine', 'truly', 'spent', 'sent', 'return', 'poor', 'pictures', 'pictured', 'paid', 'half', 'give', 'flower', 'embarrassment', 'day', 'chose', 'bouquet', '5', '00' related to Quality. Investigate root causes immediately.
2. Prioritize resolving 'scramble last minute' to improve Quality experience.

Order - Recommendations

1. Customers frequently mention 'fedex got delayed', 'new bouquet monday', 'looked nothing like', 'new ones', 'looked droupey', 'complimentary bouquet', 'past sarurday', 'original message', 'money spent', 'mil loved', 'half dead', 'good thing', 'customer service', 'bouqs website', 'bouqs offered', 'also doubt', 'ordered flowers', 'flowers leaked', 'flowers choices', 'flowers arrived', 'never received', 'like', 'flowers', 'never', 'received', 'worth', 'wife', 'way', 'vase', 'upset', 'told', 'submit', 'starting', 'spoke', 'soon', 'smashed', 'sister', 'send', 'scratch', 'roses', 'respectful', 'replaced', 'reciepient', 'product', 'problem', 'pictured' related to Order. Investigate root causes immediately.
2. Prioritize resolving 'fedex got delayed' to improve Order experience.

3rd download button will provide full analytics pdf report with recommendations.

SentimentIQ System Analysis Report

Overview

SentimentIQ is a comprehensive customer review sentiment analysis platform that transforms unstructured customer feedback into actionable business insights. The system combines Natural Language Processing (NLP), aspect-based sentiment analysis, and interactive data visualization to help businesses understand customer opinions across different product/service dimensions.

Core Capabilities:

- Automated aspect extraction and sentiment classification
- Net Promoter Score (NPS) analysis with dynamic scaling
- Interactive chat interface for data exploration
- Comprehensive PDF reporting with visualizations
- Multi-format file support (CSV, XLSX)
- Real-time data processing with progress tracking

Technology Stack: Streamlit frontend, PostgreSQL database, spaCy NLP, NLTK sentiment analysis, scikit-learn, matplotlib visualization

Frontend Code Analysis

1. Main Application (app.py)

Authentication Layer The application implements a secure authentication wrapper that protects the entire analysis interface. Users must authenticate before accessing any functionality.

File Upload & Data Processing

- Supports CSV and XLSX files with automatic encoding detection
- Implements smart column auto-detection for review text and NPS scores
- Provides data filtering capabilities by categorical columns
- Removes invalid reviews (blank, "N/A", "unidentified", etc.)
- Handles large datasets with performance warnings

Interactive Column Selection

Auto-detect review column based on common naming patterns

auto_review_col = auto_detect_review_column(df)

Auto-detect NPS column with heuristics for numeric rating columns

auto_nps_col = auto_detect_nps_column(df)

Aspect Configuration Interface Users can select from 10 predefined common aspects or add custom aspects:

- Common aspects: Quality, Delivery, Price, Customer Service, Packaging, Refund, Order, Website, Value, Communication
- Custom aspects: Comma-separated text input
- Maximum limit: 10 aspects per analysis

Real-time Analysis Progress The system provides live feedback during processing with spinner indicators and progress counters showing "Processed X/Y reviews..."

Visualization Components

NPS Gauge Visualization Implements a dynamic NPS scoring system that works with any rating scale:

Dynamic threshold calculation

promoters = (scores >= max_score - range_size * 0.2).sum()

detractors = (scores < min_score + range_size * 0.4).sum()

nps_score = pct_promoters - pct_detractors

Chart Generation Functions

- plot_overall_sentiment(): Pie chart for overall sentiment distribution
- plot_aspect_popularity(): Grouped bar chart comparing positive/neutral/negative mentions
- plot_top_negative_aspects(): Focused chart on most problematic areas
- plot_review_length(): Histogram of review character lengths
- plot_wordcloud(): Visual word frequency representation
- plot_top_ngrams(): N-gram analysis for phrase patterns

Interactive Chat Interface

Dynamic Suggestion System Generates contextual chat suggestions based on analysis results:

```
def generate_suggestions(summ_df):
  suggestions = []
  top_aspects = summ_df["Aspect"].head(3).tolist()
  for aspect in top_aspects:
    suggestions.append(f"Sentiment summary for {aspect}")
    suggestions.append(f"Recommendations for {aspect}")
```

Chat Response Logic

- Pattern matching for common queries (sentiment, recommendations, negative aspects)
- Context-aware responses using analyzed data
- Fallback to suggestion prompts for unclear queries

2. Backend Code Analysis

Sentiment Engine (sentiment_engine.py)

Core NLP Pipeline The system uses a multi-stage approach for aspect-based sentiment analysis:

1. **Text Preprocessing**: Removes encoding artifacts, normalizes Unicode characters
2. **Sentence Segmentation**: Uses spaCy for intelligent sentence boundary detection
3. **Aspect Matching**: Regex-based keyword matching within sentence contexts
4. **Sentiment Classification**: VADER sentiment analyzer for polarity scoring

Key Functions:

extract_dynamic_aspects_user()

```
def extract_dynamic_aspects_user(review_text, user_aspects):
  doc = nlp(review_text)
  extracted = defaultdict(list)
  for sent in doc.sents:
    sent_score = sia.polarity_scores(sent_text)["compound"]
    # Classify as Positive (≥0.3), Negative (≤-0.3), or Neutral
    for aspect in user_aspects:
      if re.search(r'\b' + re.escape(aspect) + r'\b', sent_text_lc):
        extracted[aspect.title()].append((sent_text, label, aspect))
```

analyze_review_structured() Main orchestration function that:

- Processes each review for all specified aspects
- Extracts sentiment-bearing sentences containing aspect keywords
- Aggregates results into structured DataFrame
- Includes progress tracking for user feedback

Data Aggregation Functions

- generate_sentiment_summary(): Creates aspect-level summary statistics

- benchmark_kpis(): Calculates overall performance metrics
- groupby_supplier_product(): Optional vendor-specific analysis

3. Authentication System (auth.py)

Security Architecture Implements multi-factor authentication with:

- bcrypt password hashing with salts
- TOTP (Time-based One-Time Password) using PyOTP
- QR code generation for authenticator app setup
- Session state management

Database Integration

```
def get_user(username):
  rows = run_query(
    "SELECT user_id, username, password_hash, totp_secret FROM nlp.users WHERE username = %s;",
    (username,), fetch=True
  )
```

Registration Flow

1. Username/password validation
2. Password confirmation check
3. Duplicate username prevention
4. bcrypt hash generation
5. TOTP secret creation
6. QR code display for app setup

Login Process

1. Username validation
2. Password verification (bcrypt)
3. OTP prompt and verification
4. Session state establishment

4. Model Loader

Sentiment Model Integration (model_loader.py)

Model Architecture Uses Cardiff NLP's RoBERTa-based sentiment classifier:

- Model: cardiffnlp/twitter-roberta-base-sentiment
- Framework: HuggingFace Transformers
- Device: Automatic CUDA/CPU detection

Key Functions:

```
def load_sentiment_model(device=None):
  tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
  model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME)
  model.to(device)
  model.eval()
  return tokenizer, model, device
```

Inference Pipeline

- Text tokenization with padding/truncation
- Batch processing capability
- Softmax probability conversion
- Label mapping to human-readable sentiment

Note: The current implementation shows the model loader but it's not actively used in the main application, which relies on VADER sentiment analysis instead.

5. Requirements & Dependencies

Core Dependencies

Web Framework

streamlit==1.47.1

NLP & ML Libraries

transformers==4.49.0

torch==2.4.1

spacy==3.8.7

nltk==3.9.1

scikit-learn==1.4.2

rake-nltk==1.0.6

langdetect==1.0.9

Data Processing

pandas==2.3.1

numpy (implicit)

openpyxl==3.1.5

Visualization

matplotlib==3.9.0

seaborn==0.13.2

wordcloud==1.9.3

plotly==5.23.0

PDF Generation

fpdf2==2.8.4

Security & Authentication

bcrypt

pyotp

qrcode

Database

psycopg2-binary

Alternative UI (unused in main app)

gradio==4.44.1

System Requirements

- Python 3.8+
- Internet connection for model downloads
- PostgreSQL database server
- Memory: 2GB+ recommended for large datasets

Model Dependencies

- spaCy English model: en_core_web_sm
- NLTK data: stopwords, vader_lexicon, punkt tokenizers

6. Database & Connections

Database Schema (db_connection.py)

Connection Management

```
@st.cache_resource
def get_db_params():
    return {
        "host": st.secrets["postgres"]["host"],
        "dbname": st.secrets["postgres"]["dbname"],
        "user": st.secrets["postgres"]["user"],
        "password": st.secrets["postgres"]["password"],
        "port": st.secrets["postgres"]["port"],
        "cursor_factory": RealDictCursor
    }
```

User Table Structure

```
-- Inferred schema from code
CREATE TABLE nlp.users (
    user_id SERIAL PRIMARY KEY,
    username VARCHAR UNIQUE NOT NULL,
    password_hash VARCHAR NOT NULL,
    totp_secret VARCHAR NOT NULL
);
```

Connection Features

- Uses RealDictCursor for dictionary-style result access
- Streamlit secrets integration for secure credential management
- Connection pooling through @st.cache_resource

Query Utilities

```
def run_query(query, params=None, fetch=False, many=False):
```

Handles parameterized queries with automatic connection management

Supports both single and batch operations

Includes proper connection cleanup

Privacy & Security Structure

Authentication Security

- **Password Protection**: bcrypt hashing with salts (industry standard)
- **Multi-Factor Authentication**: TOTP implementation compatible with Google Authenticator
- **Session Management**: Streamlit session state for authenticated sessions
- **SQL Injection Prevention**: Parameterized queries throughout

Data Privacy

- **Local Processing**: Reviews analyzed in browser/server memory
- **No Data Persistence**: Analysis results not stored in database
- **Temporary Files**: PDF reports use temporary files with automatic cleanup
- **User Isolation**: Each session operates independently

Security Best Practices

- Input validation for file uploads
- Error handling without information disclosure
- Secure secret management through Streamlit secrets
- Connection resource management with proper cleanup

Alignment with Goals

Business Intelligence Focus

The system directly supports business decision-making through:

- Actionable recommendation generation
- Aspect-specific performance metrics
- Comparative analysis capabilities
- Executive-ready PDF reports

Scalability Considerations

- Handles datasets up to 20,000+ reviews
- Progress tracking for user experience
- Memory-efficient processing
- Modular architecture for feature expansion

User Experience Design

- Intuitive step-by-step workflow
- Auto-detection reduces configuration burden
- Interactive chat for data exploration
- Multiple export formats (CSV, PDF)

Technical Architecture

- Clear separation between authentication, analysis, and presentation layers
- Extensible aspect detection system
- Comprehensive error handling and user feedback
- Modern web application stack

Learning Flow for New Team Members

Phase 1: System Understanding

Prerequisites Setup:

1. Install Python 3.8+ and create virtual environment
2. Install PostgreSQL and create development database
3. Clone repository and install requirements: pip install -r requirements.txt
4. Download spaCy model: python -m spacy download en_core_web_sm

Architecture Comprehension:

1. Study the main application flow in app.py line by line
2. Understand authentication system in auth.py
3. Examine database connection patterns in db_connection.py
4. Review sentiment analysis pipeline in sentiment_engine.py

Phase 2: Hands-On Experience

Local Development:

1. Set up Streamlit secrets for database connection
2. Create test user account and verify authentication flow
3. Upload sample CSV file and run complete analysis
4. Test each visualization component
5. Experiment with chat interface functionality

Code Tracing Exercises:

1. Follow data flow from file upload to final report
2. Trace sentiment analysis for a single review
3. Understand NPS calculation with different rating scales
4. Examine PDF generation process

Phase 3: Development Contributions

Initial Tasks:

1. Fix minor UI/UX issues or add small features

2. Improve error handling in specific functions
3. Add new visualization types
4. Enhance chat response patterns

Advanced Development Areas:

1. **Performance Optimization**: Implement caching for repeated analyses
2. **Feature Enhancement**: Add new aspect detection algorithms
3. **Integration**: Connect additional data sources or export formats
4. **Security**: Enhance authentication or add audit logging

Key Development Principles

Code Organization:

- Follow existing module structure
- Maintain separation between UI and business logic
- Use consistent error handling patterns
- Document complex NLP operations

Testing Approach:

- Test with various file formats and edge cases
- Verify sentiment accuracy with known examples
- Validate authentication flows thoroughly
- Performance test with large datasets

Feature Development:

- Consider user experience in all UI changes
- Maintain backward compatibility for existing analyses
- Ensure new features integrate with PDF reporting
- Add appropriate progress indicators for long operations

Common Development Patterns

Adding New Visualizations:

1. Create plot function following existing naming convention
2. Add matplotlib figure generation with consistent styling

3. Include in main analysis results section
4. Update PDF report generation if needed

Extending Aspect Detection:

1. Modify extract_dynamic_aspects_user() function
2. Consider impact on aggregation functions
3. Update recommendation generation logic
4. Test with diverse review datasets

Database Schema Changes:

1. Update connection and query functions
2. Modify authentication flows if needed
3. Ensure backward compatibility
4. Update user management interfaces

This learning flow ensures new team members can contribute effectively while maintaining system quality and security standards.