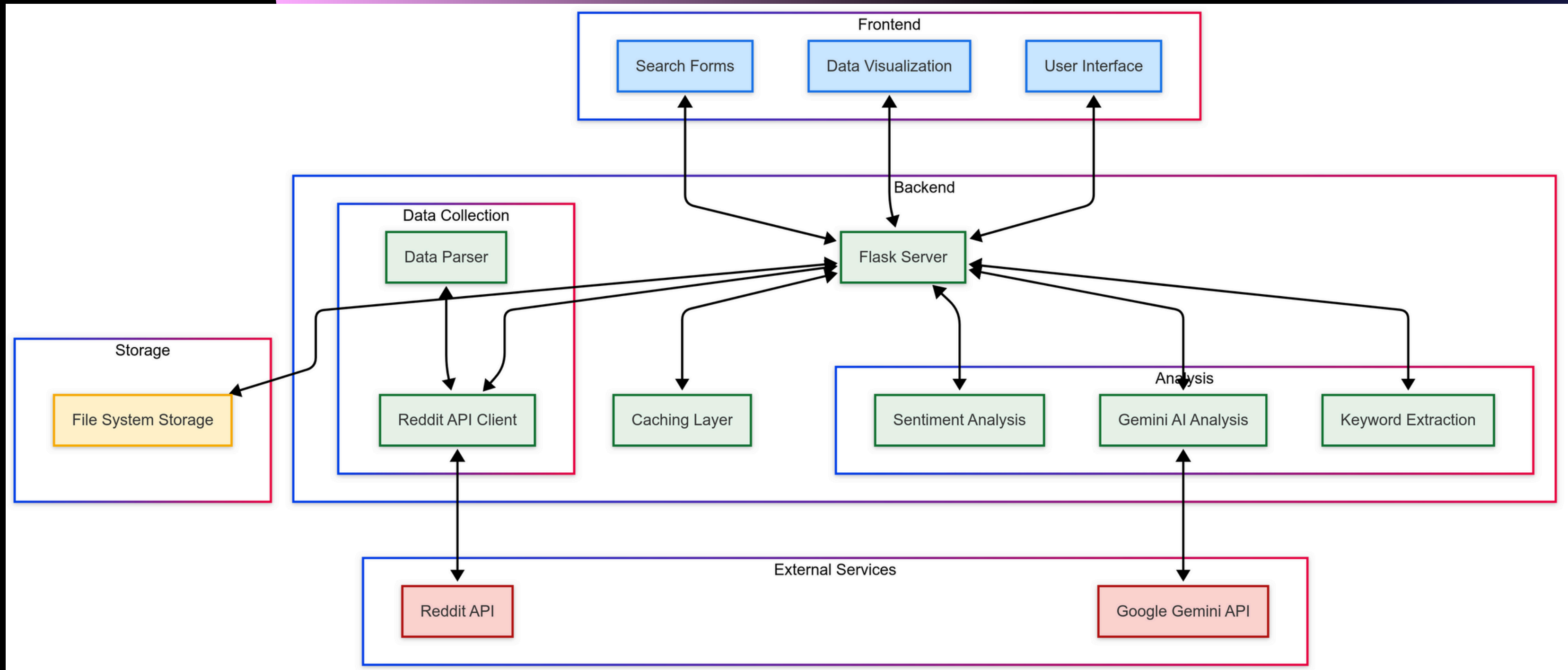


High Level Design of Therapeutic web scraper

System Design & Architecture



COMPONENT SPECIFICATIONS

A. Frontend Components

Component	Technology	Function
Dashoard	Bootstrap 5+ chart.js	Visualizes sentiment/keyword trends
Search Form	HTML5 + Fetch API	Submits scraping parameters
Analysis Modal	Bootstrap Modal	Displays therapeutic insights
Client-side Logic	DOM Manipulation	Dynamic content updates



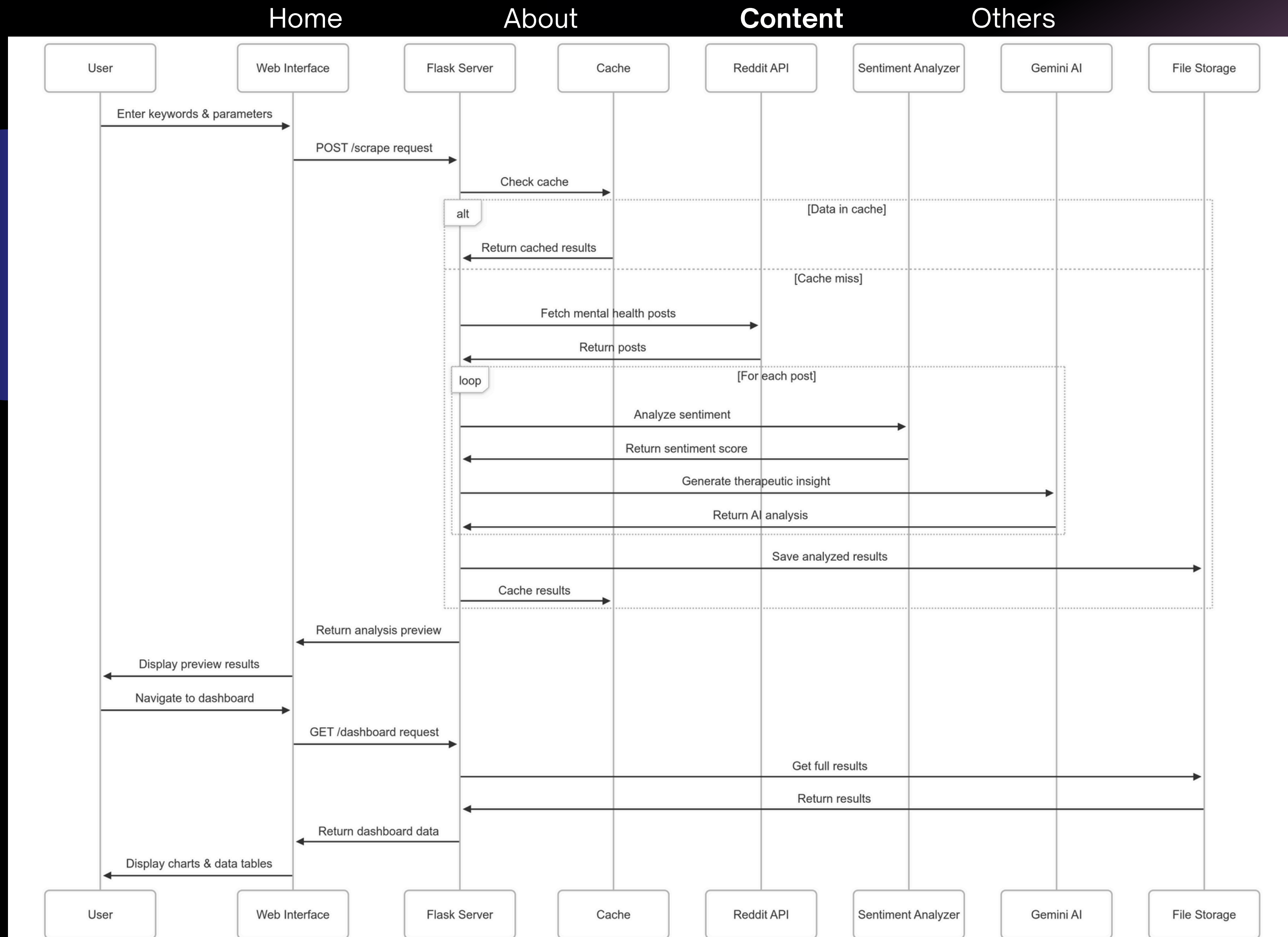
COMPONENT SPECIFICATIONS

B. Backend Components

Service	Language/Tool	Function
API Server	Flask (Python)	Routes requests between components
Scraper	PRAW Library	Fetches Reddit posts with rate limiting
NLP Engine	DistilBERT + Gemini API	Performs dual sentiment/therapeutic analysis
Cache	JSON files + MD5 hashing	Stores query results
Configuration	Python-dotenv	Environment variable management



Data flow



USER ROLES

- Researchers/Analysts: Primary users who analyze mental health discussions
- Mental Health Professionals: Use insights to understand patient concerns and trends
- System Administrators: Manage API keys and system configuration

KEY FUNCTIONS & PROCESS STEPS

1. Reddit Content Scraping

Function: `reddit_scraper.scrape_mentalhealth()`

Steps:

Connect to Reddit API using PRAW

Search across mental health subreddits

Filter by user-provided keywords

Extract post metadata (title, content, author, date, etc.)

Return structured data for analysis

2. Sentiment Analysis

Function: `sentiment_analyzer.analyze_text()`

Steps:

Initialize HuggingFace sentiment analysis pipeline

Process text content with character limits

Return sentiment label (POSITIVE, NEGATIVE, NEUTRAL) and confidence score

3. AI-Powered Therapeutic Insights

Function: `gemini_analyzer.generate_insight()`

Steps:

Process post content to fit API limits

Generate prompt for Gemini AI model

Call Google Gemini API

Process response and extract insights

Return structured therapeutic analysis

4. Data Caching & Persistence

Functions: `get_cached_results()`, `cache_results()`, `save_results()`

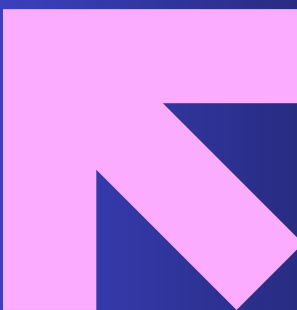
Steps:

Generate hash key based on search parameters

Check for existing cached results

Store new results in JSON format

Make results available for dashboard visualization



KEY FUNCTIONS & PROCESS STEPS

5. Dashboard Visualization

Pages: [/dashboard endpoint](#)

Components:

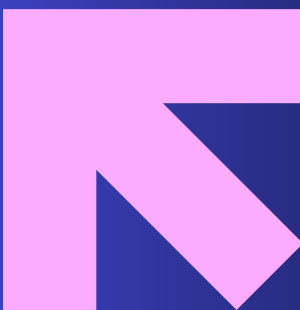
Sentiment distribution chart

Keyword frequency analysis

Interactive post filtering

Detailed post analysis functionality

CSV export capabilities



Development Phases

Phase 1: Core Setup & Infrastructure

- Setup Flask application structure
- Create project directory layout
- Configure environment variables and .env support
- Implement basic error handling and logging
- Setup basic templates with Bootstrap

Phase 2: Reddit Integration

- Implement RedditScraper class
- Add Reddit API authentication
- Create search functionality by keywords
- Implement data extraction and structure
- Add basic error handling for API failures

Phase 3: Analysis Components

- Implement SentimentAnalyzer class with HuggingFace
- Create GeminiAnalyzer class for AI insights
- Connect to Google Gemini API
- Develop prompt engineering for mental health analysis
- Implement analysis caching mechanism

Phase 4: Frontend Development

- Create base templates and layouts
- Implement index/search page
- Develop dashboard with Chart.js visualizations
- Add interactive features (filtering, sorting)
- Implement post analysis modal functionality(CSV export)

Development Phases

Phase 5: Data Management & Export

- Implement file-based storage system
- Create caching mechanism
- Add CSV export functionality
- Implement dashboard filters
- Add keyword extraction feature

Phase 6: Testing & Refinement

- Unit testing for core components
- Integration testing for API connections
- UI/UX testing and refinements
- Performance optimization
- Security review

Phase 7: Documentation & Deployment

- Create user documentation
- Add code documentation
- Prepare for deployment
- Implement final error handling
- Create administrator guide

Key Data Models

```
{
  "redditPost": {
    "source": "reddit",
    "subreddit": "mentalhealth",
    "title": "Struggling with anxiety after starting new medication",
    "content": "I started a new medication last week and my anxiety
has been through the roof...",
    "author": "username123",
    "date": "2023-04-15T14:32:18",
    "url": "https://reddit.com/r/mentalhealth/comments/abc123",
    "upvotes": 24,
    "comments": 12
  },
  {
    "analyzedPost": {
      "source": "reddit",
      "subreddit": "mentalhealth",
      "title": "Struggling with anxiety after starting new medication",
      "content": "I started a new medication last week and my anxiety has
been through the roof...",
      "author": "username123",
      "date": "2023-04-15T14:32:18",
```

```
    "url": "https://reddit.com/r/mentalhealth/comments/abc123",
    "upvotes": 24,
    "comments": 12,
    "sentiment": "NEGATIVE",
    "sentiment_score": 0.87,
    "insight": "This post reflects medication adjustment anxiety, a
common experience when starting new psychiatric medications.
The intense physical symptoms described are typical side effects
that often subside after 1-2 weeks of treatment. The author would
benefit from medical reassurance about the temporary nature of
these symptoms and possibly short-term coping strategies."
  },
```

Key Data Models

```
"sentimentAnalysis": {  
  "label": "NEGATIVE",  
  "score": 0.87  
},
```

```
"keywordFrequency": {  
  "anxiety": 28,  
  "therapy": 19,  
  "depression": 15,  
  "medication": 12,  
  "coping": 8  
},
```

```
"sentimentCounts": {  
  "positive": 12,  
  "negative": 25,  
  "neutral": 3  
},
```

```
"cacheEntry": {  
  "key": "anxiety,depression,therapy-10",  
  "timestamp": "2023-04-15T14:35:12",  
  "expires": "2023-04-16T14:35:12",  
  "results": [  
    /* Array of analyzedPost objects */  
  ]  
}
```

Error Handling Strategy

API Error Handling

Reddit API Failures:

- Graceful degradation with informative messages
- Retry mechanism for transient failures
- Clear error messages on API quota limits

Gemini API Failures:

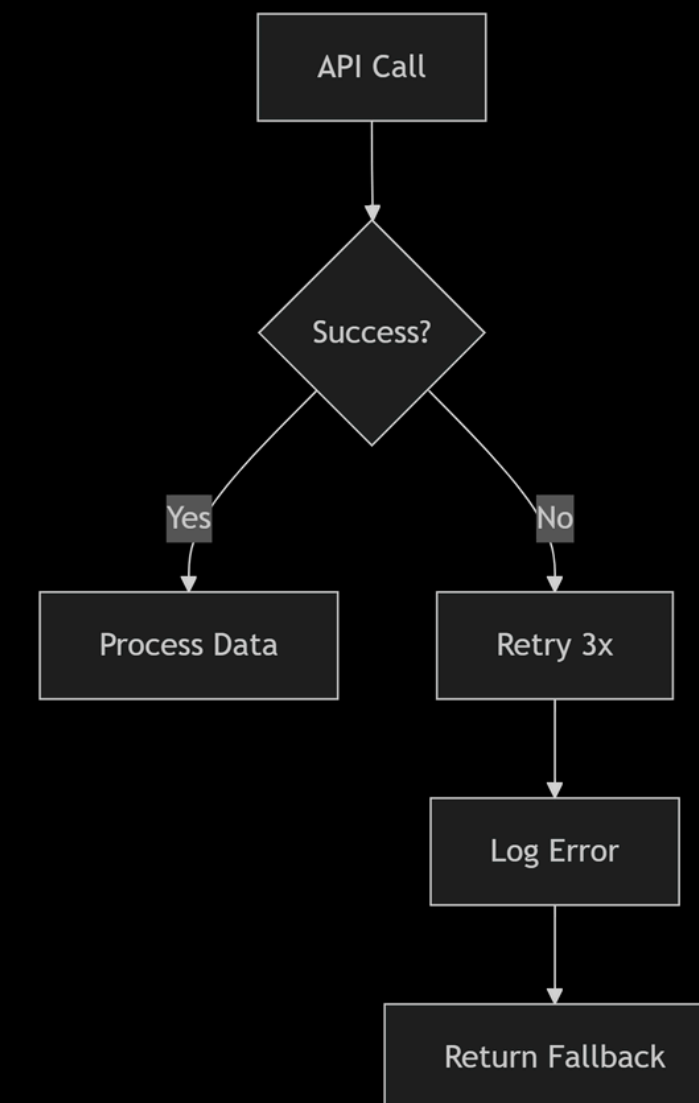
- Fallback to basic analysis when AI insights unavailable
- Proper logging of API error responses
- User-friendly error messages without technical details

General Error Handling:

- Structured JSON error responses
- HTTP status codes matched to error types
- Detailed server-side logging with minimal client exposure

User-Facing Error Reporting

- Bootstrap alert components for error visualization
- Clear, actionable error messages
- API key configuration guidance when relevant
- Spinner/loading indicators during processing



SECURITY CONSIDERATIONS

API Key Management

- Environment variable storage of sensitive keys
- No API keys in client-side code
- Validation of API key presence on startup

Input Validation

- Server-side validation of all user inputs
- Sanitization of HTML/JS content from Reddit
- Proper escaping in templates to prevent XSS

Data Protection

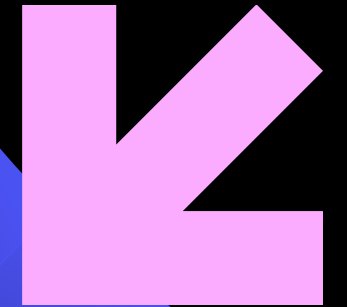
- Local file storage with proper permissions
- No personally identifiable information stored
- Optional data retention policies

External API Security

- Rate limiting considerations
- Adherence to API terms of service
- Proper error handling for API authentication failures

Frontend Security

- Cross-Site Scripting (XSS) prevention
- Content Security Policy considerations
- Proper escaping of user-generated content



Thank You

Presented by
Rima Chaieb
BA/IT