MattGPT Project Customization Guide

Overview

This guide covers backend customization options for the MattGPT research platform, focusing on four key areas: changing the system name, configuring AI models, setting up company data, and customizing news sources.

---

1. Changing the System Name

Frontend Changes

To rename "MattGPT" to your preferred name, update these files:

`index.html`

- **Line ~6**: Update the page title

```html
<title>YourName: Your Research Buddy</title>
```

- **Line ~17**: Update the main header

```html
<h1>YourName</h1>
```

- **Lines with welcome messages**: Search for "MattGPT" and replace with your chosen name

```javascript
// Find and replace in JavaScript sections
'👋 **Welcome to YourName Research Assistant**'
```

Logo File

- Replace `/static/MattGPT.png` with your custom logo
- Maintain 80x80 pixel dimensions for optimal display
- Update the CSS reference if changing the filename:

```css
background-image: url('/static/YourLogo.png');
```

`style.css`

- **Line ~60**: Update logo background image reference if filename changed

  ```css
  background-image: url('/static/YourLogo.png');
  ```

---

2. Model Configuration

Centralized Configuration System

The system uses **fully centralized model configuration** through `sources_config.json`. The model is loaded once and propagated throughout all system components.

Single Configuration Point: `data/sources_config.json`

Update this single file to change the model system-wide:

```json
{
  "model_settings": {
    "ollama_model": "your-preferred-model:tag",
    "temperature": 0.1,
    "max_tokens": 2000
  }
}
```

How Model Propagation Works:

1. **ContentCurator** loads model from `sources_config.json`
2. **System initialization** passes `curator.ollama_model` to all other components:
   - **InterestManager**: `InterestManager(ollama_model=curator.ollama_model)`
   - **Current Events**: `set_ollama_model(curator.ollama_model)`
   - **Chat Assistant**: `ChatAssistant(curator.ollama_model)`
   - **Customer Insights**: Receives model via parameter passing

Hardcoded Fallbacks:

Components have `granite3.2:8b` as hardcoded fallbacks, but these are **only used if the centralized configuration fails to load**. In normal operation, all components use the model specified in `sources_config.json`.

Supported Models

The system supports any Ollama-compatible model. Popular alternatives:

- `llama3.2:8b`
- `mistral:7b`
- `codellama:7b`
- `phi3:medium`
- `gemma2:9b`

Model Configuration Best Practices

1. **Performance Considerations**:
   - Larger models (>8B parameters) provide better analysis quality
   - Smaller models (<7B parameters) offer faster response times
   - Test with your hardware capabilities
2. **Temperature Settings**:
   - `0.1-0.3`: More deterministic, factual analysis
   - `0.5-0.7`: Balanced creativity and consistency
   - `0.8-1.0`: More creative, varied responses
3. **Token Limits**:
   - Current events: 2000 tokens (recommended)
   - Customer insights: 3000-4000 tokens for detailed analysis
   - Content curation: 2000 tokens sufficient

---

3. Company Data Setup (`my_company.JSON`)

File Location

`data/my_company.JSON`

Complete Configuration Structure

```json
```

```
{
  "company_info": {
    "name": "Your Company Name",
    "description": "Brief company description highlighting core value proposition"
  },
  "industries": [
    {
      "name": "Industry Name",
      "specialties": ["Specialty 1", "Specialty 2", "Specialty 3"],
      "key_products": ["Product A", "Product B"],
      "competitive_advantages": ["Advantage 1", "Advantage 2"],
      "information_links": ["https://company.com/industry-page"]
    }
  ],
  "products": [
    {
      "name": "Product Name",
      "category": "Software|Hardware|Service",
      "target_industries": ["Industry 1", "Industry 2"],
      "key_features": ["Feature 1", "Feature 2"],
      "competitive_advantages": ["Advantage 1", "Advantage 2"],
      "information_links": ["https://product-info.com"]
    }
  ],
  "specialties": [
    {
      "name": "Specialty Name",
      "description": "Detailed description of this specialty",
      "applicable_industries": ["Industry 1", "All"],
      "competitive_advantages": ["Advantage 1", "Advantage 2"],
      "information_links": ["https://specialty-info.com"]
    }
  ]
}
```

Configuration Guidelines

Company Info Section

- **Name**: Used in battle cards and customer analysis
- **Description**: Should be 1-2 sentences capturing core value proposition

Industries Section

- **Purpose**: Defines markets you serve
- **Specialties**: Specific capabilities within each industry
- **Key Products**: Industry-specific product offerings
- **Competitive Advantages**: What differentiates you in this industry
- **Information Links**: Resources for more details

Products Section

- **Category**: Classify as Software, Hardware, or Service
- **Target Industries**: Which industries can use this product
- **Key Features**: Technical/functional capabilities
- **Competitive Advantages**: Why choose this over alternatives

Specialties Section

- **Applicable Industries**: Use "All" for cross-industry specialties
- **Description**: Detailed capability explanation
- **Competitive Advantages**: Unique value propositions

Impact on System Features

This configuration affects:

- **Customer Battle Cards**: Product recommendations and positioning strategies
- **Industry Analysis**: Relevant insights based on your served markets
- **Conversation Starters**: Business development suggestions
- **Competitive Positioning**: How your company is presented to prospects

---

4. News Sources Configuration

Configuration Files

The system uses two different configuration files for news sources:

Current Events: `data/current_events_sources.json`

```json
```

```
{
 "sources": [
  {
   "name": "Source Display Name",
   "url": "https://source.com/rss",
   "category": "technology|business|world"
  }
 ],
 "tech_business_keywords": [
  "keyword1",
  "keyword2",
  "industry-specific-term"
 ]
}
```

Content Curator: `data/sources_config.json`

```
json
```

```
{
 "model_settings": {
  "ollama_model": "your-model:tag",
  "temperature": 0.1,
  "max_tokens": 2000
 },
 "rss_feeds": [
  {
   "url": "https://source.com/rss",
   "name": "Source Name",
   "category": "technology"
  }
 ],
 "arxiv_queries": [
  {
   "query": "machine learning",
   "max_results": 5
  }
 ],
 "collection_settings": {
  "default_max_results_per_source": 5,
  "rate_limit_delay_seconds": 2,
  "rss_max_entries_per_feed": 20
 }
}
```

Recommended News Sources

Technology Sources

```
json
```

```
{
  "name": "TechCrunch",
  "url": "https://techcrunch.com/feed/",
  "category": "technology"
},
{
  "name": "Ars Technica",
  "url": "https://feeds.arstechnica.com/arstechnica/index",
  "category": "technology"
},
{
  "name": "The Verge",
  "url": "https://www.theverge.com/rss/index.xml",
  "category": "technology"
},
{
  "name": "Wired",
  "url": "https://www.wired.com/feed/",
  "category": "technology"
}
```

Business Sources

json

```
{
  "name": "Reuters Business",
  "url": "https://feeds.reuters.com/reuters/businessNews",
  "category": "business"
},
{
  "name": "Fast Company",
  "url": "https://www.fastcompany.com/latest/rss",
  "category": "business"
},
{
  "name": "VentureBeat",
  "url": "https://venturebeat.com/feed/",
  "category": "business"
}
```

Research Sources

json

```json
{
  "name": "MIT Technology Review",
  "url": "https://www.technologyreview.com/feed/",
  "category": "research"
},
{
  "name": "Google Research Blog",
  "url": "https://blog.research.google/feeds/posts/default",
  "category": "research"
}
```

Industry-Specific Sources

Financial Services

- **American Banker**: `https://www.americanbanker.com/feed`
- **Financial Planning**: `https://www.financial-planning.com/feed`
- **Banking Dive**: `https://www.bankingdive.com/feeds/news/`

Healthcare

- **Healthcare IT News**: `https://www.healthcareitnews.com/rss.xml`
- **Modern Healthcare**: `https://www.modernhealthcare.com/rss.xml`
- **HIMSS**: `https://www.himss.org/rss.xml`

Manufacturing

- **Manufacturing.net**: `https://www.manufacturing.net/rss.xml`
- **Industry Week**: `https://www.industryweek.com/rss.xml`
- **Plant Engineering**: `https://www.plantengineering.com/rss.xml`

Keyword Configuration

Tech Business Keywords

Customize the keyword list to match your industry focus:

json

```
"tech_business_keywords": [
  "ai", "artificial intelligence", "machine learning",
  "digital transformation", "cloud computing", "cybersecurity",
  "fintech", "healthtech", "edtech",
  "your-industry-specific-terms",
  "competitor-names", "market-trends"
]
```

Keyword Strategy

- **Include competitor names** for competitive intelligence
- **Add industry jargon** relevant to your customers
- **Include technology trends** affecting your markets
- **Add geographic terms** if you serve specific regions

ArXiv Research Queries

For academic and research content:

```json
"arxiv_queries": [
  {"query": "artificial intelligence business", "max_results": 5},
  {"query": "machine learning applications", "max_results": 5},
  {"query": "your-research-area", "max_results": 3}
]
```

Source Performance Optimization

RSS Feed Reliability

- **Test feeds** before adding to ensure they're active
- **Monitor feed performance** - remove consistently failing sources
- **Balance source types** between news, analysis, and research

Rate Limiting Settings

- **Delay between requests**: 2-3 seconds recommended
- **Max entries per feed**: 20 for active feeds, 10 for slower sources
- **Total source limit**: 15-25 sources for optimal performance

Category Distribution

Recommended distribution:

- **40%** Technology/Industry-specific
- **30%** Business/Market news
- **20%** Research/Academic
- **10%** General news/World events

Implementation Checklist

Pre-Customization

□ Backup original configuration files
□ Document current model performance benchmarks
□ Test current functionality before changes

Name Change Implementation

□ Update `index.html` title and headers
□ Replace logo file in `/static/` directory
□ Update CSS logo references
□ Test frontend display after changes

Model Configuration

□ Verify chosen model is installed in Ollama
□ Update model references in all Python files
□ Update `sources_config.json` model settings
□ Test model performance with sample queries
□ Monitor response times and quality

Company Data Setup

□ Create `data/my_company.JSON` from template
□ Populate all required sections with accurate data
□ Validate JSON syntax
□ Test customer analysis with company data
□ Verify battle card generation quality

News Sources Configuration

- □ Create/update `data/current_events_sources.json`
- □ Create/update `data/sources_config.json`
- □ Test RSS feed connectivity
- □ Verify keyword relevance to your industry
- □ Monitor news analysis quality

Post-Implementation Testing

- □ Run "What's Going On?" analysis
- □ Test customer battle card generation
- □ Verify news source processing
- □ Check system performance with new model
- □ Validate all customizations work together

Troubleshooting

Common Issues

Model Not Found

- Verify model is installed: `ollama list`
- Install model: `ollama pull model-name:tag`
- Check model name spelling in configuration files

RSS Feed Failures

- Test feed URLs manually in browser
- Check for authentication requirements
- Verify feed format is RSS/XML
- Review rate limiting settings

Company Data Not Loading

- Validate JSON syntax with online validator
- Check file path: `data/my_company.JSON`
- Verify file permissions
- Review application logs for error messages

Performance Issues

- Monitor system resources during analysis
- Consider smaller model for faster processing
- Reduce number of RSS sources
- Increase rate limiting delays

Validation Commands

Test Model Availability

```bash
ollama list | grep your-model-name
```

Validate JSON Files

```bash
python -m json.tool data/my_company.JSON
python -m json.tool data/current_events_sources.json
python -m json.tool data/sources_config.json
```

Test RSS Feeds

```bash
curl -I "https://your-rss-feed-url"
```

Advanced Customization

Custom Analysis Prompts

The system uses specific prompts for different analysis types. Advanced users can modify these in:

- `current_events.py`: News analysis prompts
- `customer_insights.py`: Customer analysis prompts
- `content_curator.py`: Content analysis prompts

Database Customization

Customer data storage can be extended by modifying:

- `customers.csv` structure

- Database schema in various modules
- Cache management settings

Integration Points

The system provides hooks for:

- Custom authentication
- External data sources
- API integrations
- Workflow automation

This guide provides comprehensive backend customization options while maintaining system functionality. Always test changes in a development environment before implementing in production.