JHU-AI Project 1: Complete Implementation Outline

Phase 0: Fix Current Issues - **START HERE**

Fix Path and API Setup

```python
# 1. Fix the email directory path
EMAIL_DIR = "/content/drive/MyDrive/email_assistant/mock_emails"  # Add /mock_emails

# 2. Add OpenAI API setup
import openai
from google.colab import userdata

# Set your API key (you'll need to add this to Colab secrets)
openai.api_key = userdata.get('OPENAI_API_KEY')  # Or use your preferred method

# 3. Re-run the email loading with correct path
eml_paths = glob.glob(os.path.join(EMAIL_DIR, "*.eml"))
print(f"Found {len(eml_paths)} email files.")  # Should show 30 files
```

Phase 1: Data Preparation & Email Classification (Foundation)

Step 1: Review and Organize Your Sample Emails ✅ COMPLETED

☑ Ensure you have a diverse set of sample emails representing different priorities

☑ Create a data structure (list/dictionary) to store emails with metadata

☑ Include fields: sender, subject, content, timestamp, original_category (if any)

☑ **30 mock emails generated and parsed into DataFrame**

☑ **DataFrame with columns: file, from, subject, body, date**

Step 2: Email Classification System - **NEXT PRIORITY**

□ **Write System Prompt** for email classification

- Define AI role as email classifier
- Specify 6 categories: Urgent & High-Priority, Deadline-Driven, Routine Updates, Non-Urgent/Informational, Personal/Social, Spam/Unimportant
- Include classification criteria for each category

□ **Write User Prompt** for classification

- Format: "Classify the following email into one of these categories..."

□ **Test Classification** on your sample emails

□ **Store Results** with assigned categories

□ **Add OpenAI API setup** (you'll need this for all AI tasks)

Phase 2: Task 1A - Executive Dashboard Creation

Step 3: Dashboard Summary System

□ **Write System Prompt** for dashboard creation

- Define AI role as executive email assistant
- Specify output format matching the sample (with emojis and counts)
- Emphasize clarity and actionable insights

□ **Write User Prompt** for dashboard

- Include classified email counts
- Request executive summary with AI conclusion

□ **Generate Dashboard** using your classified emails

□ **Format Output** to match sample structure exactly

Phase 3: Task 1B - Urgent & High-Priority Email Analysis

Step 4: Urgent Email Deep Dive

□ **Filter Emails** to only urgent & high-priority ones

□ **Write System Prompt** for urgent analysis

- Define AI role as business operations analyst
- Focus on immediate action requirements
- Emphasize impact on critical projects/client relationships

□ **Write User Prompt** for urgent analysis

- Request detailed analysis of each urgent email
- Ask for next steps recommendations

□ **Generate Analysis** for each urgent email

□ **Compile Results** in structured format

Phase 4: Task 1C - Deadline-Driven Email Analysis

Step 5: Time-Sensitive Email Review

□ **Filter Emails** to only deadline-driven ones

□ **Write System Prompt** for deadline analysis

- Define AI role as project timeline coordinator

- Focus on time-sensitive requirements
- Emphasize deadline adherence

□ **Write User Prompt** for deadline analysis

- Request analysis of time-sensitive emails
- Ask for priority ordering and next steps

□ **Generate Analysis** for each deadline-driven email
□ **Compile Results** with timeline recommendations

Phase 5: Task 2 - AI Response Draft Generation

Step 6: Response Drafting System

□ **Combine Critical Emails** (Urgent + Deadline-Driven)
□ **Write System Prompt** for response drafting

- Define AI role as professional email composer
- Specify email structure: Salutation → Greeting → Body → Next Steps → Closing
- Emphasize appropriate tone and professionalism
- Include context awareness requirements

□ **Write User Prompt** for each critical email

- Include original email content
- Request properly structured response draft

□ **Generate Response Drafts** for all critical emails
□ **Review Structure** (ensure all 5 components are present)

Phase 6: Task 3 - LLM-as-a-Judge Evaluation

Step 7: Evaluation System Setup

□ **Select Content to Evaluate** (dashboard, analyses, response drafts)
□ **Write System Prompt** for LLM judge

- Define AI role as content evaluator
- Specify evaluation criteria: Relevance, Clarity, Actionability
- Request structured scoring format
- Include strengths, improvements, and justification requirements

□ **Write User Prompt** for evaluation

- Include original task description
- Include AI-generated content to evaluate
- Request scoring on 1-10 scale for each criterion

Step 8: Conduct Evaluations

□ **Evaluate Dashboard Summary** (Task 1A output)
□ **Evaluate Urgent Email Analysis** (Task 1B output)
□ **Evaluate Deadline Email Analysis** (Task 1C output)
□ **Evaluate Response Drafts** (Task 2 outputs)
□ **Compile Evaluation Results** in structured format

Phase 7: Task 4 - Analysis & Recommendations

Step 9: Project Reflection

□ **Analyze Your Results**

- Review evaluation scores
- Identify patterns in strengths/weaknesses
- Note which tasks performed best/worst

□ **Write Observations Section**

- Summarize overall project strengths
- Identify key weaknesses or limitations
- Comment on AI effectiveness for email management

□ **Write Improvement Suggestions**

- Recommend specific enhancements
- Suggest additional features for email efficiency
- Propose system optimization ideas

Phase 8: Final Integration & Documentation

Step 10: Notebook Organization

□ **Clean Up Code**

- Add clear function names
- Include inline documentation
- Organize cells logically

□ **Add Section Headers**

- Clear task delineation
- Professional formatting

□ **Include All Outputs**

- Dashboard summary

- Email analyses
- Response drafts
- Evaluation results
- Recommendations

Step 11: Quality Assurance

□ **Verify All Requirements Met**

- Check each grading criterion
- Ensure all tasks completed
- Confirm proper prompt placement

□ **Test All Code Cells**

- Run notebook from top to bottom
- Fix any errors
- Verify outputs display correctly

□ **Convert to HTML**

- Use Jupyter's File → Download as → HTML
- Review HTML output for formatting
- Ensure all content is visible

Current Status Analysis

✅ What You've Completed:

1. Environment setup (OpenAI, pandas, mail-parser, faker)
2. Google Drive integration
3. Mock email generation (30 emails)
4. Email parsing into structured DataFrame
5. Data loaded and sorted by date

🚨 Issues to Fix First:

1. **Path Issue**: Your final glob search is looking in `/content/drive/MyDrive/email_assistant` but emails are in `/content/drive/MyDrive/email_assistant/mock_emails`
2. **OpenAI API Key**: You need to add your API key setup
3. **Missing Prompts**: Need to add system/user prompt placeholders

📋 Immediate Next Steps:

System Prompt Best Practices

- Always start with clear AI role definition
- Include specific output format requirements
- Provide context about the business scenario
- Emphasize professionalism and accuracy

User Prompt Best Practices

- Include all necessary context/data
- Be specific about desired output format
- Reference the business context (Alex Carter, Orion Tech Solutions)
- Ask for actionable insights

Evaluation Criteria Alignment

- **Classification (1 pt)**: Ensure 6 categories are used correctly
- **Dashboard (3 pts)**: Focus on AI identity, goal clarity, readability
- **Urgent Analysis (3 pts)**: Include next steps recommendations
- **Deadline Analysis (3 pts)**: Include next steps recommendations
- **Response Drafts (5 pts)**: Ensure all 5 structural elements + proper tone
- **LLM Judge (3 pts)**: Use specified evaluation format
- **Recommendations (2 pts)**: Include observations and improvement suggestions

Estimated Timeline

- **Phase 1-2**: 2-3 hours (foundation + dashboard)
- **Phase 3-4**: 2-3 hours (email analyses)
- **Phase 5**: 2-3 hours (response drafting)
- **Phase 6**: 1-2 hours (evaluations)
- **Phase 7-8**: 1-2 hours (recommendations + cleanup)
- **Total**: 8-13 hours

Next Steps

1. Review your current notebook against this outline
2. Identify which phases you've completed

3. Focus on the next incomplete phase

4. Ask for specific help with prompts or implementation details