

# **Phase 6 Production ready**

## **Overview**

Phase 6's objective is to transform the AI System's working prototype into a finished, usable product. While Phase 6 focusses on creating a system that is robust, scalable, reliable, and maintainable for real-world application, the prior phases were primarily concerned with data ingestion, model development, and system integration. This phase focusses on performance optimisation, edge case handling, evaluation, testing, maintenance, and real-world deployment of that system.

The goal of all of these components is to develop a system that will enable the AI workflow to function efficiently in the actual world while maintaining a respectable degree of accuracy, stability, and usability over time.

## **Subtask 1 – Performance Optimizing and caching**

### **Objective**

To improve system efficiency, reduce latency, and minimize unnecessary computational overhead.

### **Key optimizations Implemented**

#### **a) Model Optimization**

1. Selection of optimized model variants suitable for production (e.g., lightweight or distilled models).
2. Reduction of inference time by tuning hyperparameters.
3. Avoidance of redundant model loading by keeping models in memory during runtime.

#### **b) Data Processing Optimization**

1. Batch processing of inputs instead of single-item inference where applicable.
2. Streamlined preprocessing pipelines to remove unnecessary transformations.
3. Efficient memory handling for large documents and datasets.

#### **c) Caching Mechanisms**

Caching is used to prevent repeated computation for frequently processed inputs.

##### **Caching strategies include:**

1. **Input-level caching:** Previously processed documents return cached results.
2. **Embedding caching:** Vector embeddings are stored and reused for similarity searches.
3. **Response caching:** Frequently requested outputs are cached temporarily.

##### **Impact**

1. Reduced response time
2. Lower computational cost
3. Improved user experience in real time systems

## **Subtask 2 – Edge case Handling**

### **Objective**

To ensure system reliability when handling non-ideal or unexpected inputs.

## Identified Edge Cases

### a) Scanned PDFs

1. Scanned documents contain images instead of selectable text.
2. Optical Character Recognition (OCR) is applied to extract text.
3. OCR confidence thresholds are used to detect poor-quality scans.

#### Handling Strategy:

1. Automatically route scanned PDFs through an OCR pipeline.
2. Flag documents with low OCR confidence for manual review.

### b) Multilingual Documents

1. Documents may contain multiple languages or non-English text.
2. Language detection is performed before processing.
3. Translation or multilingual models are used when required.

#### Handling Strategy:

1. Detect language using lightweight language identification models.
2. Route content to language-specific pipelines or translators.

### c) Data Updates and Version Changes

1. Documents may be updated or replaced over time.
2. Model outputs may become outdated.

#### Handling Strategy:

1. Version control for documents and embeddings.
2. Reprocessing triggers when source data changes.
3. Metadata tracking for timestamps and versions.

```
[venv] C:\Users\raven\OneDrive\Desktop\Intern_Project\Intern Project>python test_rag.py
⚠️ Milvus not available: <MilvusException: (code=2, message='Fail connecting to server on localhost:19530, illegal connection params or server unavailable')>
Final Answer:
Air traffic controllers face several challenges, including:
1. Managing high volumes of air traffic: With increasing air travel demand, controllers must efficiently manage multiple flights simultaneously, ensuring safe distances and efficient flight paths.
2. Maintaining situational awareness: Controllers must stay alert and focused on multiple aircraft, weather conditions, and other factors that can impact flight safety.
3. Dealing with unexpected events: Controllers must respond quickly and effectively to unexpected events such as weather-related delays, mechanical issues, or air traffic conflicts.
4. Managing fatigue: Controllers often work long hours, including night shifts and overtime, which can lead to fatigue and decreased performance.
5. Adapting to new technologies: Air traffic control systems are constantly evolving, and controllers must learn to use new technologies and procedures to maintain efficiency and safety.
6. Prioritizing safety: Controllers must prioritize safety above all else, making decisions that balance the needs of multiple aircraft and passengers.
7. Managing stress: The high-stress environment of air traffic control can take a toll on controllers' mental health and well-being.
8. Coordinating with other agencies: Controllers must work with other agencies, such as military and law enforcement, to ensure safe and efficient air traffic management.
9. Managing air traffic during special events: Controllers must manage air traffic during special events, such as air shows, festivals, or natural disasters, which can create unique challenges.
10. Staying up-to-date with regulations: Controllers must stay current with changing regulations and procedures to ensure compliance and maintain safety standards.
```

*Figure 1: Subtask 2*

## **Phase 3 – Evaluation Metrics and Testing Frameworks**

### **Objective**

To validate the system performance, accuracy and reliability before deployment

### **Evaluation Metrics**

1. Accuracy Metrics
  - a. Precision
  - b. Recall
  - c. F1-Score
  - d. Confidence scores for extracted or generated outputs
2. Performance Metrics
  - a. Inference latency
  - b. Throughput (requests per second)
  - c. Memory usage
  - d. CPU/GPU utilization
3. Robustness Metrics
  - a. Error rate on noisy inputs
  - b. OCR accuracy for scanned documents
  - c. Multilingual processing success rate

### **Testing Frameworks Used**

1. **Unit testing**
  - a. Individual modules (preprocessing, inference, post processing) tested independently
  - b. Ensures correctness of each component
2. **Integration Testing**
  - a. Tests full AI workflow from input ingestion to output generation
  - b. Validates interaction between subsystems.
3. **Stress and Load Testing**
  - a. Stimulates high user traffic and large data volumes
  - b. Identifies performance bottlenecks and failure points
4. **Regression testing**
  - a. Ensures new updates do not degrade existing functionality.
  - b. Automated test cases run after each major change

```
Command Prompt + ▾
7. Managing fatigue: Air traffic controllers often work long hours, which can lead to fatigue, and they must be able to manage their workload and take regular breaks to stay alert and focused.

8. Adapting to new technologies: Air traffic controllers must be able to adapt to new technologies and systems, such as automated air traffic control systems, to stay up-to-date with the latest developments in air traffic management.

9. Managing multiple tasks: Air traffic controllers must be able to manage multiple tasks simultaneously, such as handling multiple aircraft and responding to emergencies, which requires strong multitasking skills.

10. Maintaining a high level of concentration: Air traffic controllers must be able to maintain a high level of concentration over long periods, which can be mentally demanding.
Expected: Air traffic controllers face challenges such as workload, stress, fatigue, maintaining situational awareness, and handling emergencies.
Semantic Similarity: 0.779
⚠ Milvus not available: <MilvusException: (code=2, message=Fail connecting to server on localhost:19530, illegal connection params or server unavailable)>

Question: Why is situational awareness important in air traffic control?
Predicted: Situational awareness is crucial in air traffic control because it enables controllers to have a comprehensive understanding of the current situation, including the location, speed, altitude, and intentions of all aircraft in their airspace. This awareness allows them to anticipate potential conflicts, make informed decisions, and take proactive measures to prevent accidents or delays.
Expected: Situational awareness is important to ensure safe separation of aircraft and prevent collisions.
Semantic Similarity: 0.744

===== Evaluation Summary =====
Exact Match Accuracy: 1.0
Average Semantic Similarity: 0.7615024447441101

(venv) C:\Users\raven\OneDrive\Desktop\Intern_Project\Intern Project>
```

Figure 2: Subtask 3

## Phase 4 – Monitoring and Development

### Objective

To continuously observe system behaviour and maintain log term reliability

### Monitoring Components

1. System Monitoring
  - a. CPU, memory and storage usage
  - b. Inference latency and throughput
  - c. Error rates and failure logs
2. Model Monitoring
  - a. Prediction confidence trends
  - b. Data drift detection
  - c. Performance degradation over time
3. Logging and Alerting
  - a. Centralized logging for debugging and audits
  - b. Alerts triggered for anomalies or failures
  - c. Error classification for faster troubleshooting

### Continuous Development

1. Automated pipelines for retraining and redevelopment
2. Model versioning and rollback mechanisms
3. CI/CD integration for safe and repeatable releases
4. Documentation updates aligned with system changes

```
C:\Users\raven\OneDrive\Desktop\Intern_Project>uvicorn api:app --reload
INFO:     Will watch for changes in these directories: ['C:\\\\Users\\\\raven\\\\OneDrive\\\\Desktop\\\\Intern_Project\\\\Intern Project']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [19276] using StatReload
INFO:     Started server process [20724]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO: 127.0.0.1:49975 - "GET /docs HTTP/1.1" 200 OK
INFO: 127.0.0.1:49975 - "GET /openapi.json HTTP/1.1" 200 OK
INFO: 127.0.0.1:55182 - "GET /health HTTP/1.1" 200 OK
INFO: 127.0.0.1:60505 - "GET /health HTTP/1.1" 200 OK
```

Figure 3: Subtask 4

## Phase 5 – Final Capstone Project

### Objective

To deliver a complete, production ready AI system that demonstrates real-world applicability.

### Capstone Deliverables

1. Fully functional AI-driven workflow
2. Optimized and scalable architecture
3. Robust handling of edge cases
4. Comprehensive evaluation and testing results
5. Monitoring and maintenance strategy
6. Detailed technical documentation

### Deployment Readiness

The system is packaged for deployment using:

1. Configurable environment variables
2. Secure API endpoints
3. Modular architecture for extensibility
4. Clear setup and execution instructions

### Outcome

By the end of Phase 6, the project transitions from a development-stage prototype to a deployable AI solution, capable of operating reliably in real-world environments while meeting performance, accuracy, and maintainability standards.

### Conclusion

Phase 6 represents the final and most critical stage of the AI workflow lifecycle. It ensures that the system is not only functional but also production-grade, capable of handling real-world complexity, scale, and unpredictability. Through performance optimization, robust edge case handling, systematic evaluation, continuous monitoring, and a finalized capstone deployment, the AI solution is prepared for real-world use and future enhancements.