PYQ Analyzer – Pipeline

📘 Master Project Execution Plan

Evidence-Based Exam Preparation Analytics System

🔵 PHASE 0 — Scope Locking & Dataset Finalization

Status: Planning phase (do once, never revisit)

🎯 Goal

Freeze the problem context so development does not drift.

What you do

- Fix **subject**: Deep Learning
- Fix **exam type**: FAT
- Fix **dataset**:
    - 1 text-based PDF (model paper)
    - 3 scanned/image-based PDFs
- Fix **minimum papers rule**: ≥ 3
- Fix **syllabus**: official DL syllabus (module-wise)

Output of this phase

- A folder with:

/data

├— syllabus.pdf

├— DL_FAT_model.pdf

├— DL_FAT_1_scanned.pdf

├— DL_FAT_2_scanned.pdf

├— DL_FAT_3_scanned.pdf

Why this phase matters

- Prevents scope creep
- Makes project defensible in reviews
- Establishes dataset assumptions clearly

✅ **Do not move forward until this is frozen**

🔵 **PHASE 1 — PYQ Ingestion & Question Extraction**

Status: Foundation phase (most critical)

🎯 **Goal**

Convert unstructured question papers into **structured, machine-readable questions**.

What you do

1. Load each PDF
2. For each page:
    - If text exists → extract directly
    - Else → apply OCR → extract text
3. Normalize text (remove noise, fix spacing)
4. Detect:
    - Question numbers (Q1, Q2…)
    - Sub-questions (a, b, c)
    - Marks (if present)
    - Paper ID, year, exam type

Output of this phase (MANDATORY structure)

{

"paper_id": "DL_FAT_2023",

"exam_type": "FAT",

```
 "year": 2023,
 "questions": [
  {
   "question_id": "Q3_b",
   "text": "Explain backpropagation algorithm",
   "marks": 8,
   "page_number": 4
  }
 ]
}
```


Validation rule

- Manually verify at least **10–15 extracted questions**
- If extraction is wrong → fix here, not later

Why this phase matters

- Everything else depends on this
- Clean input = reliable analytics

⛔ **Never proceed with dirty question extraction**

🔵 PHASE 2 — Syllabus Parsing & Topic Universe Creation

Status: Constraint-definition phase

🎯 Goal

Make the system **syllabus-aware and hallucination-safe**.

What you do

1. Parse syllabus PDF
2. Extract:
    - Total number of modules

    - Topic list under each module
3. Store syllabus as a **fixed topic universe**

Output of this phase

```
{
  "Module 1": [
    "Introduction to Deep Learning",
    "Perceptron",
    "Activation Functions"
  ],
  "Module 2": [
    "Backpropagation",
    "Loss Functions",
    "Gradient Descent"
  ]
}
```

Why this phase matters

- Prevents topic invention
- Enables module-wise analysis
- Anchors all intelligence to official syllabus

🔵 PHASE 3 — Exam-Aware Module Range Control

Status: Context-awareness phase

🎯 Goal

Support real exam patterns (CAT/FAT style).

What you do

- Implement logic to select module range:

start_module → end_module

- Filter questions to only those modules
- Allow analysis to change dynamically based on range

Output of this phase

Selected modules: 1–5

Questions considered: 42 / 65

Why this phase matters

- Reflects real exam scope
- Makes system practical, not generic

🔵 PHASE 4 — Question → Topic Mapping

Status: Core intelligence phase

🎯 Goal

Map each question to syllabus topics **without explicit subject knowledge**.

What you do

1. Compare question text with syllabus topics
2. Perform **syllabus-constrained semantic matching**
3. Assign:
    - Module
    - Topic
    - Confidence score
4. Detect sub-concepts if present in question wording

Output of this phase

{

"question_id": "Q3_b",

"module": 2,

"topic": "Backpropagation",

"confidence": 0.86,

"detected_subconcept": "Chain Rule"

}

Why this phase matters

- Enables frequency analysis
- Enables repetition detection
- Resolves your earlier “subject knowledge” doubt cleanly

🔵 PHASE 5 — Analytics Engine

Status: Value-creation phase

This phase answers **most student-facing questions**.

🔹 Phase 5.1 — Topic Frequency & Importance (Module-wise)

Answers

What should I study in this module?

What you do

- Count topic occurrences
- Weight by marks
- Rank topics per module

Output

Module 2 Priority:

1. Backpropagation (6 times, avg 8 marks)

2. Gradient Descent (4 times)

3. Loss Functions (2 times)

🔹 Phase 5.2 — Canonical Question Detection

Answers

What questions repeat even if phrased differently?

What you do

- Compare semantic similarity of questions
- Cluster similar ones
- Select a canonical form
- Count repetitions
- Attach references

Output

Canonical Question:

Explain Backpropagation Algorithm

Repeated:

- 2019 Q5

- 2021 Q3(b)

- 2023 Q4(a)

🔹 Phase 5.3 — Module-wise Grading Analysis

Answers

Which modules carry more marks?

What you do

- Aggregate marks per module
- Average across papers

Output

- Pie chart

- Percentage contribution per module

🔹 Phase 5.4 — Question Paper Style Analysis

Answers

Descriptive or numerical?

What you do

- Classify questions using rules
- Compute probability distribution

Output

Descriptive: 82%

Analytical/Numerical: 18%

🔵 PHASE 6 — Dependency-Aware Preparation Sequencing

Status: Student guidance phase

🎯 Goal

Give **step-by-step clarity**, not just rankings.

What you do

- Detect prerequisite relationships using:
    - Topic naming patterns
    - Simple rules
    - Optional manual hints
- Generate learning sequence

Output

Recommended Study Order:

1. Perceptron

2. Activation Functions

3. Backpropagation

4. Gradient Descent

Why this phase matters

- Reduces confusion
- Converts analytics into a preparation plan

🔵 PHASE 7 — Evidence-Based Dashboard & UX

Status: Trust-building phase

🎯 Goal

Present insights with **complete transparency**.

What you do

- Build dashboard views:
  - Topic rankings
  - Repeated questions
  - Charts & graphs
- Every insight links to:
  - Question text
  - Paper ID
  - Year
  - Question number

Why this phase matters

- Prevents hallucination
- Builds student confidence
- Impresses evaluators

🔵 PHASE 8 — Validation, Documentation & Evaluation

Status: Academic safety phase

🎯 Goal

Make the project defensible.

What you do

- Test on unseen question paper
- Manually verify mappings
- Document:
    - Assumptions
    - Limitations
    - Confidence scores
    - Failure cases

Why this phase matters

- Shows maturity
- Aligns with research ethics
- Strengthens viva defense