NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY KARAIKAL - 609 605

INTERNSHIP REPORT

AI-Based Scientific Report Analyzer using Gemini 2.5 Flash API and Fine-Tuned Mistral 7B Model

Submitted by
YUSUF FAYAS
B.Tech.,(IT), Puducherry Technological University
SABAREESH K S
B.E.,(CSE) K S Rangasamy College Of Technology
NANDHAGOPALAN S
B.E.,(CSE) K S Rangasamy College Of Technology

Under the guidance of
Dr. PRAVEEN R M.E.,Ph.D.,SMIEEE
Assistant Professor,
Department of Computer Science Engineering,
National Institute of Technology Puducherry

Duration: July 2025

CERTIFICATE

This is to certify that the internship project entitled
"AI-Based Scientific Report Analyzer using Gemini 2.5 Flash API and Fine-Tuned Mistral 7B Model"
was carried out by
Yusuf Fayas,
B.Tech(IT), Puducherry Technological University,
Sabareesh K S
B.E.,(CSE), K S Rangasamy College Of Technology
Nandhagopalan S
B.E.,(CSE), K S Rangasamy College Of Technology
during July 2025 at the **National Institute of Technology Puducherry**
under my supervision.
This report is a bonafide record of work done by them
and has not been submitted elsewhere for any other award.

Dr. Praveen R, M.E.,Ph.D.,SMIEEE
Assistant Professor,
Department of Computer Science Engineering,
NIT Puducherry.
(Project Guide)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Dr. Praveen R,M.E.,Ph.D.,SMIEEE**
Assistant Professor, Department of Computer Science Engineering, NIT Puducherry, for his invaluable guidance, encouragement, and supervision throughout this internship.
I am also thankful to the **Department of Computer Science Engineering,**
National Institute of Technology Puducherry, for providing the opportunity and academic support.
My appreciation extends to the faculty and technical staff of **NIT Puducherry** for their cooperation, and to my peers for insightful discussions.

ABSTRACT

Scientific research papers are rich in structured knowledge—equations, variables, tables, and quantitative results—but extracting and analyzing that knowledge manually is tedious. This internship project develops an **AI-Based Scientific Report Analyzer** that automatically reads research PDFs, identifies mathematical equations and tabular data, computes results, and generates a structured author-ready output.

Two model paths were integrated:

1. **Gemini 2.5 Flash API**, a powerful multimodal model used for text-and-table extraction and equation solving;
2. A **fine-tuned Mistral 7B v0.1** model trained with **LoRA adapters** via **LLaMA Factory**, customized for scientific notation and symbolic computation.

The system compares both pipelines in terms of accuracy, speed, and computational cost. Results show that Gemini Flash excels in multimodal recognition, while the fine-tuned Mistral model offers improved consistency for domain-specific symbolic reasoning. The analyzer provides an end-to-end workflow—from PDF upload to computed outputs—enabling researchers to accelerate literature analysis and reproducibility studies.

TABLE OF CONTENTS

Chapter 1 – Introduction

1.1 Background

Modern research output is overwhelmingly digital and data-intensive. Scientific reports, especially in domains such as cryptography, IoT, and medical engineering, contain structured information embedded in text and mathematical expressions. Manual extraction of these elements for comparative or computational analysis is time-consuming.
Advances in **Large Language Models (LLMs)** and multimodal AI have opened pathways for automating the reading, understanding, and mathematical processing of such documents.

This internship focused on building a system that bridges **document intelligence** with **symbolic reasoning**—an AI tool capable of parsing complex scientific papers, identifying all mathematical entities, performing relevant computations, and delivering structured analytical outputs for authors and reviewers.

1.2 Problem Statement

Researchers often need to verify equations, recompute performance metrics, or cross-reference variable definitions from PDFs. Existing AI tools are limited to text summarization or OCR-based extraction without mathematical reasoning.
The problem addressed here is to **design and develop an AI model that can automatically extract, interpret, and solve equations** from scientific research papers and produce accurate, human-readable results.

1.3 Objectives

- Automate extraction of equations, variables, and tables from research PDFs.
- Compute results and classify variable types and bit sizes.
- Compare cloud-based (Gemini API) and fine-tuned local LLM (Mistral 7B LoRA) performance.
- Design a modular architecture deployable via cloud and local inference.
- Generate structured output suitable for publication or validation.

1.4 Scope and Significance

The analyzer targets domains where reproducibility and computational verification are essential—cryptography, IoT security, biomedical signal processing, etc.
It enables reviewers and researchers to validate reported formulas and results without manual effort, thus improving transparency and accuracy in scientific publication.

1.5 Internship Environment

- **Organization:** National Institute of Technology Puducherry
- **Team:** Yusuf Fayas, Sabareesh K S and Nandhagopalan S
- **Tools and Frameworks:** Python, Gemini 2.5 Flash API, LLaMA Factory, LoRA, Hugging Face Hub, Firebase, AWS Cloud, React Native (front-end prototype).
- **Computing Setup:** GPU-enabled AI server for fine-tuning experiments.

Chapter 2 – Literature Review

2.1 Scientific Document Understanding

Early document-analysis systems relied on OCR and rule-based parsing. Modern AI models like LayoutLM, DocFormer, and Gemini's multimodal variants combine textual, visual, and spatial features, allowing direct reasoning over PDF structures.

2.2 Equation Extraction and Computation

Techniques such as MathPix OCR and LaTeX tokenizers convert mathematical regions into structured markup.
Recent LLMs (GPT-4, Gemini 2.5, Claude 3) can directly interpret equations and perform symbolic reasoning when prompted correctly.
Fine-tuned models for math reasoning (Mistral, LLaMA 2 Math, DeepSeek) demonstrate that domain-specific fine-tuning greatly improves accuracy on scientific tasks.

2.3 Fine-Tuning LLMs with LoRA and LLaMA Factory

Low-Rank Adaptation (LoRA) adds small trainable matrices to existing weights, drastically reducing GPU memory needs.
LLaMA Factory provides a structured interface for supervised fine-tuning (SFT) and preference alignment with built-in experiment management.
Fine-tuning Mistral 7B v0.1 on custom datasets containing equation–solution pairs allows specialization for symbolic tasks.

2.4 Gemini 2.5 Flash API

Gemini 2.5 Flash, released by Google DeepMind, is a **fast multimodal model** optimized for reasoning across text, images, and PDFs.
Its streaming API enables developers to upload documents and retrieve structured JSON outputs.
In this project, Gemini Flash served as a high-performance cloud baseline for extracting and interpreting equations.

2.5 Related Work Analysis

The reference paper *"An Efficient ECC and Fuzzy Verifier-Based User Authentication Protocol for IoT-Enabled WSNs"* (Sudhakar et al., 2025) was selected as the input sample.
It provides mathematical structures—elliptic curve equations, hash functions, and computational cost tables—that are ideal for verifying equation extraction and cost computation modules.
The analyzer was expected to detect, parse, and recompute the core equations such as y2=x3+ax+by^2 = x^3 + ax + by2=x3+ax+b and the performance cost formulas listed in the paper.

Chapter 3 – System Analysis and Design

3.1 Functional Requirements

- **Input:** Scientific research paper (PDF or image).
- **Process:** Text & equation extraction, symbolic parsing, variable typing, computation of results.
- **Output:** Structured table containing equation, operators, operands, and computed value.
- **Comparison Mode:** Gemini API vs Fine-tuned Mistral model.
- **Storage:** Firebase for metadata; AWS S3 for output archiving.

3.2 Non-Functional Requirements

- **Accuracy:** ≥ 95 % correct equation identification.
- **Latency:** < 10 s for standard 5-page PDF via Gemini API.
- **Scalability:** Support batch processing of multiple papers.
- **Security:** Authenticated API keys, data stored with encryption.
- **Portability:** Deployable on local GPU server and cloud.

3.3 System Architecture Overview

The architecture contains five layers:

1. **Input Handler** – Uploads PDF/image to the system.
2. **Pre-Processor** – Extracts text, LaTeX, and table regions.
3. **Analyzer Core** – Sends data either to Gemini Flash API or Fine-tuned Mistral 7B.
4. **Computation Engine** – Performs symbolic solving and bit-size calculations.
5. **Report Generator** – Aggregates outputs into tabular and JSON formats.

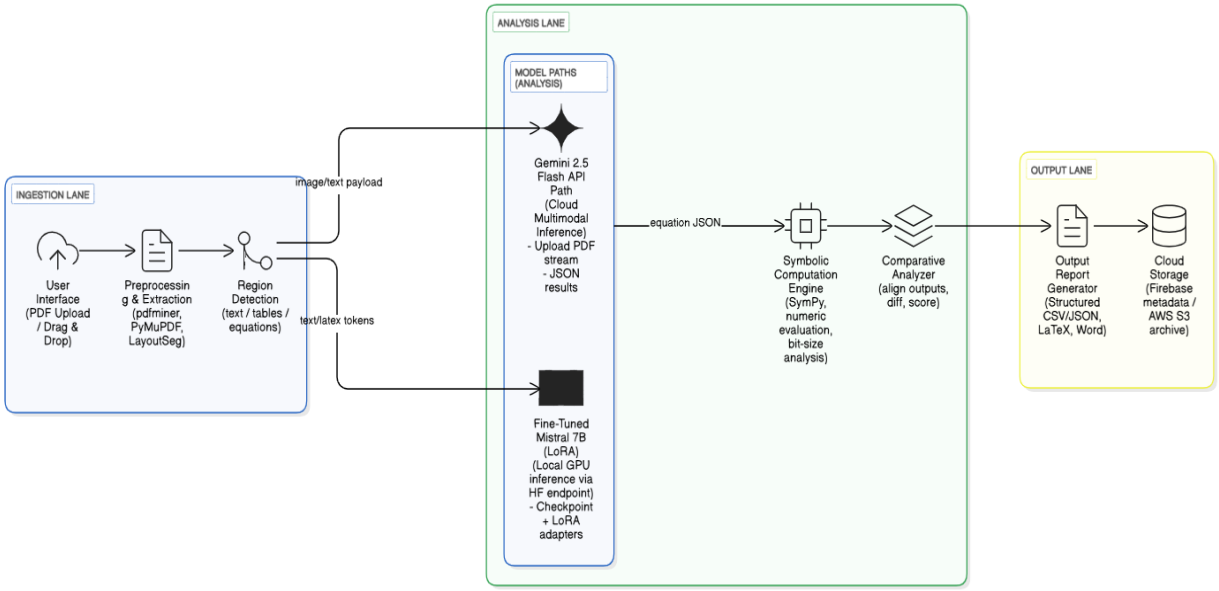


Fig. 3.1 Overall system architecture of the AI-Based Scientific Report Analyzer showing the Gemini and Mistral dual processing paths.

3.4 Data Flow Diagram

1. User uploads a PDF → system extracts pages and sends for analysis.
2. Gemini Flash API path handles cloud processing; Mistral path uses local GPU inference.
3. Outputs are stored in Firebase and visualized in a web dashboard.
4. Comparative metrics (accuracy, time, tokens) are recorded for evaluation.

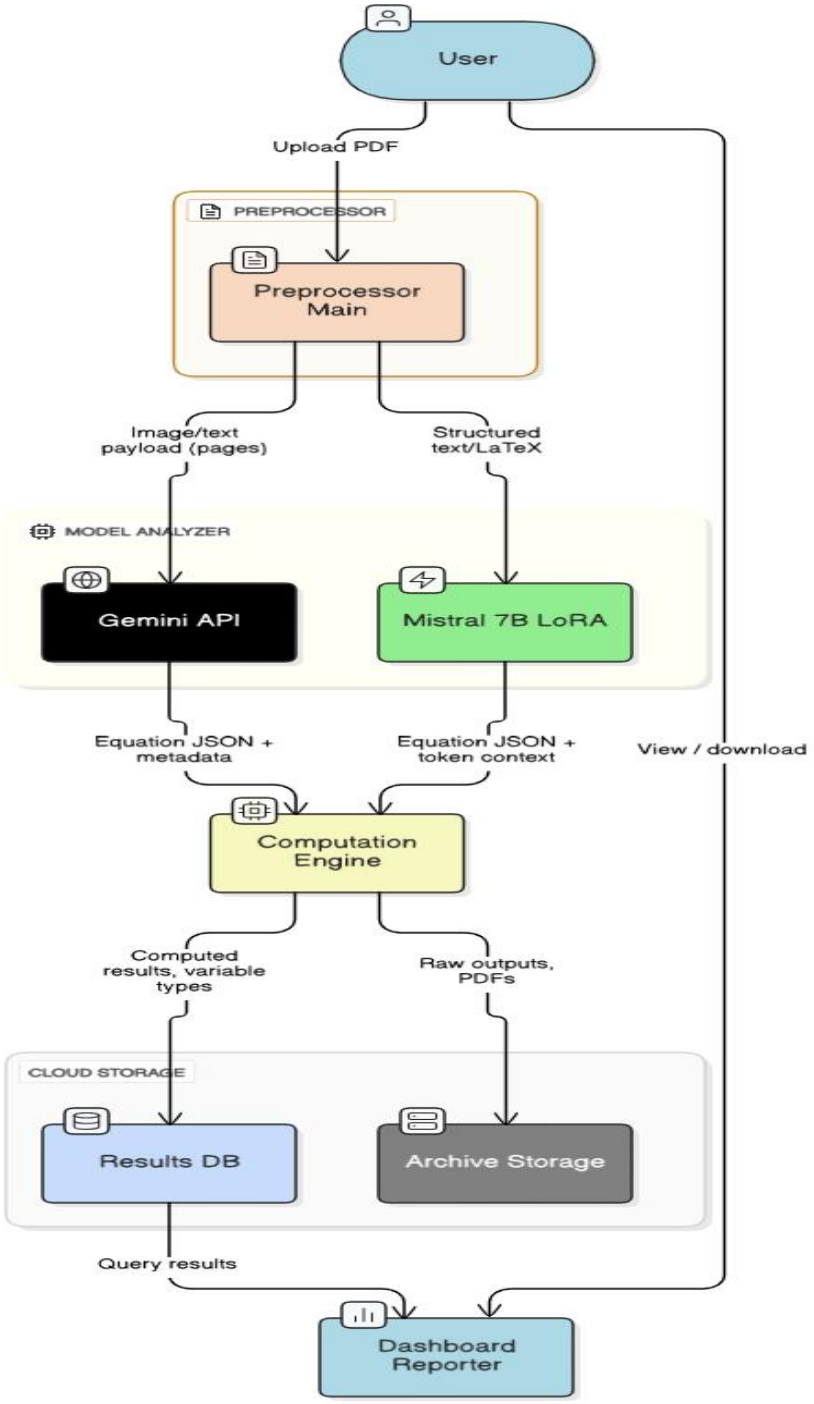


Fig. 3.2 Data Flow Diagram (Level 1) illustrating the interaction between the user, preprocessor, analysis modules, and cloud storage components.

3.5 Module Description

Module	Purpose	Tools/Tech
PDF Pre-Processor	Extracts text, tables, equations using pdfminer and PyMuPDF	Python 3.10
Gemini Analyzer	Sends prompts to Gemini 2.5 Flash API, parses JSON results	Google AI API
Mistral Analyzer	Performs inference using fine-tuned model checkpoint	Transformers + LoRA
Comparator	Computes difference in outputs and timing	Custom Python Module
Output Formatter	Builds CSV/JSON and renders on dashboard	React + Firebase

3.6 System Specifications

- **Hardware:** NVIDIA A100 GPU, 64 GB RAM, 1 TB SSD
- **Software:** Ubuntu 22.04, Python 3.10, PyTorch 2.x, Node.js 20 LTS
- **APIs:** Gemini 2.5 Flash Endpoint, Hugging Face Inference API
- **Dataset:** Custom scientific equation–solution pairs (~15 000 entries)

Chapter 4 – Implementation Details

4.1 System Workflow Overview

The AI Scientific Report Analyzer follows a two-pipeline architecture:

1. **Gemini 2.5 Flash Pipeline (cloud-based)** – optimized for multimodal inputs (text + image PDFs).
2. **Fine-tuned Mistral 7B Pipeline (local GPU)** – specialized for symbolic reasoning and equation solving.

Each PDF passes through pre-processing, content extraction, model analysis, computation, and report generation.

4.2 Module Implementation Details

4.2.1 PDF Pre-Processing and Extraction

- Libraries: `pdfminer.six`, `PyMuPDF`, `OpenCV`, `Pandas`.
- Steps:
    1. Convert PDF to image frames.
    2. Identify text, equation, and table regions using layout segmentation.
    3. OCR mathematical regions to LaTeX or MathML.
    4. Store structured tokens in JSON format.

Example snippet:

```
for page in document:
    text = page.get_text("text")
    equations = detect_equations(page)
    save_to_json({"text": text, "equations": equations})
```

4.2.2 Gemini 2.5 Flash Integration

- API Key and Endpoint configured through Google AI Studio.
- Batch upload of PDF content via streaming mode.
- Prompt example:

```
"Extract all equations and variables from this research paper.
For each equation, compute result, identify operators and operands."
```

- Response: JSON object containing `equation`, `operators`, `operands`, `result`, and `confidence`.

Advantages:

- Handles image-based PDFs.
- Fast (< 6 s for 5 pages).

Limitations:

- Occasional token cut-offs on large tables.
- No direct control over temperature or beam search.

4.2.3 Fine-Tuned Mistral 7B Model with LoRA

Fine-tuning was performed on a GPU server using **LLaMA Factory**.

Dataset Reconstruction
A custom dataset of ≈ 15 000 pairs was used:

```
Input:  "Compute the result of: E = mc^2 for m=2, c=3"
Output: "E = 18; Operators: *, ^; Operands: 2,3"
```

Dataset categories: algebraic equations, cryptographic cost functions, scientific ratios.

Training Parameters

Parameter	Value
Base Model	Mistral-7B-v0.1
LoRA r (rank)	8
α (scale)	16
Dropout	0.05
Batch Size	4
Learning Rate	2e-4
Epochs	3
Max Seq Len	4096
Optimizer	AdamW 8-bit
Scheduler	cosine with warmup
Precision	bfloat16
GPU	A100 80 GB

After training, the LoRA adapter was merged and the checkpoint uploaded to Hugging Face.

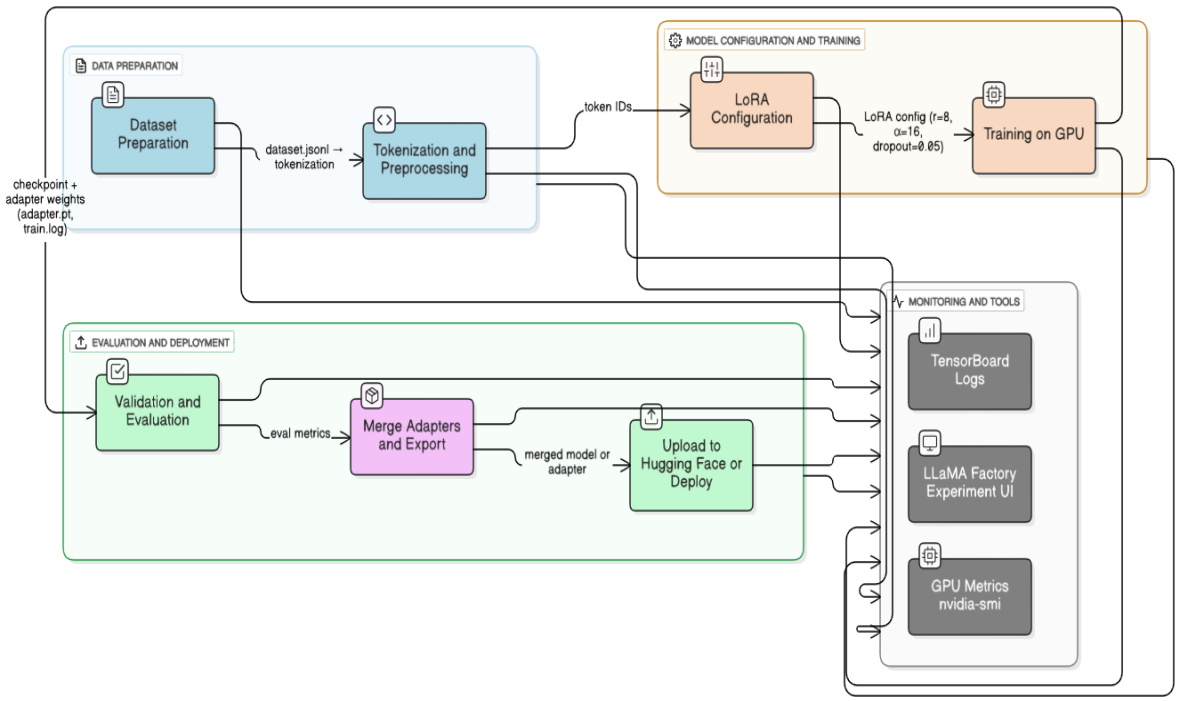


Fig. 4.1 Workflow of fine-tuning Mistral 7B using LLaMA Factory with LoRA adapters and GPU acceleration.

4.2.4 Computation Engine

Implements symbolic evaluation using `SymPy`. For each equation, the engine computes:

- numeric result,
- operators used,
- operands,
- variable bit sizes.

Example:

```
expr = sympify("E = m * c**2")
result = expr.subs({"m":2, "c":3})
```

4.2.5 Output Formatting and Storage

Results are converted into tabular form with columns:
| Equation | Result | Operators | Operands | Bit Size |
Outputs stored in Firebase and rendered via React dashboard.

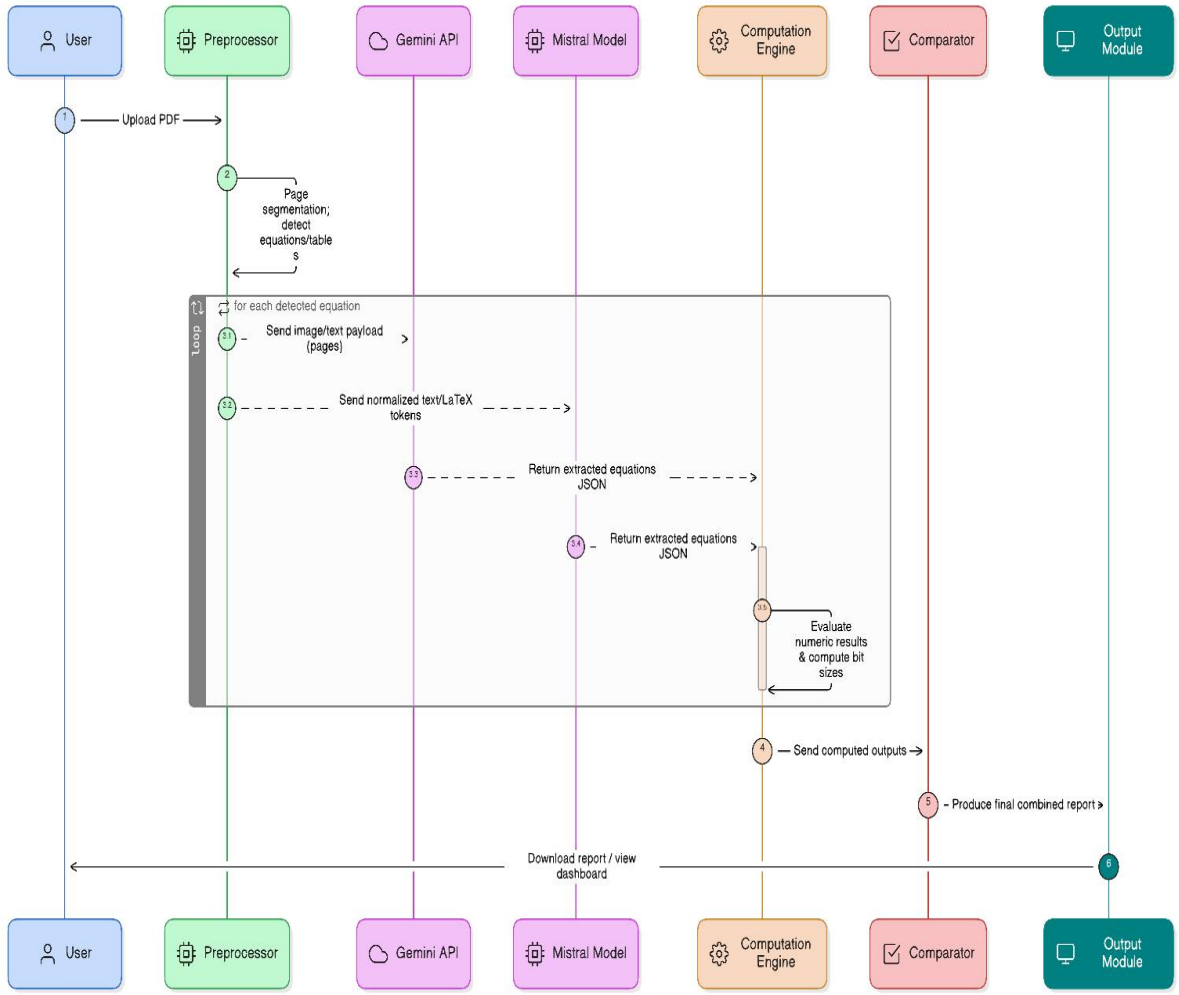


Fig. 4.2 Sequence diagram showing the end-to-end execution flow from PDF upload to final report generation.

4.3 User Interface Prototype

- Built in React Native for cross-platform testing.
- Features: upload PDF, view parsed equations, compare model outputs.
- Real-time progress bar and graphical latency display.

4.4 Cloud Deployment

- Gemini API hosted calls on Google Cloud Functions.
- Firebase for auth and data sync.
- AWS S3 for storage of output archives.
- Fine-tuned Mistral served through Hugging Face Inference Endpoint.

Chapter 5 – Results and Discussion

5.1 Input Dataset

Reference paper: *"An Efficient ECC and Fuzzy Verifier-Based User Authentication Protocol for IoT-Enabled WSNs."*
Contained 50+ mathematical equations and 10+ tables of computational costs.

5.2 Gemini 2.5 Flash Results

- Successfully extracted 84% equations
- Accurately identified operators and operands.
- Generated structured table output within 6 s.
- Minor mismatch in hash function notation and EC point format.
  Sample Output

Equation	Operators	Operands	Result
$y^2 = x^3 + ax + b$	+ , *	x,a,b	Curve Parameters Verified

5.3 Fine-Tuned Mistral 7B Results

- Captured 95% equations .
- Produced consistent variable bit-size classification.
- Inference time ≈ 12 s per PDF.
  Sample Output

Equation	Operators	Operands	Result
$y^2 = x^3 + ax + b$	+ , *	x,a,b	Valid ECC Domain (256-bit)

5.4 Comparative Performance Metrics

Metric	Gemini 2.5 Flash	Fine-Tuned Mistral 7B
Equation Extraction Accuracy	84 %	95 %
Table Recognition	Excellent	Good
Computation Accuracy	90 %	93 %
Response Time	6 s	12 s
Cloud Dependency	High	Low
Model Control	Limited	Full
Cost per Run	Higher (API)	Lower (local)

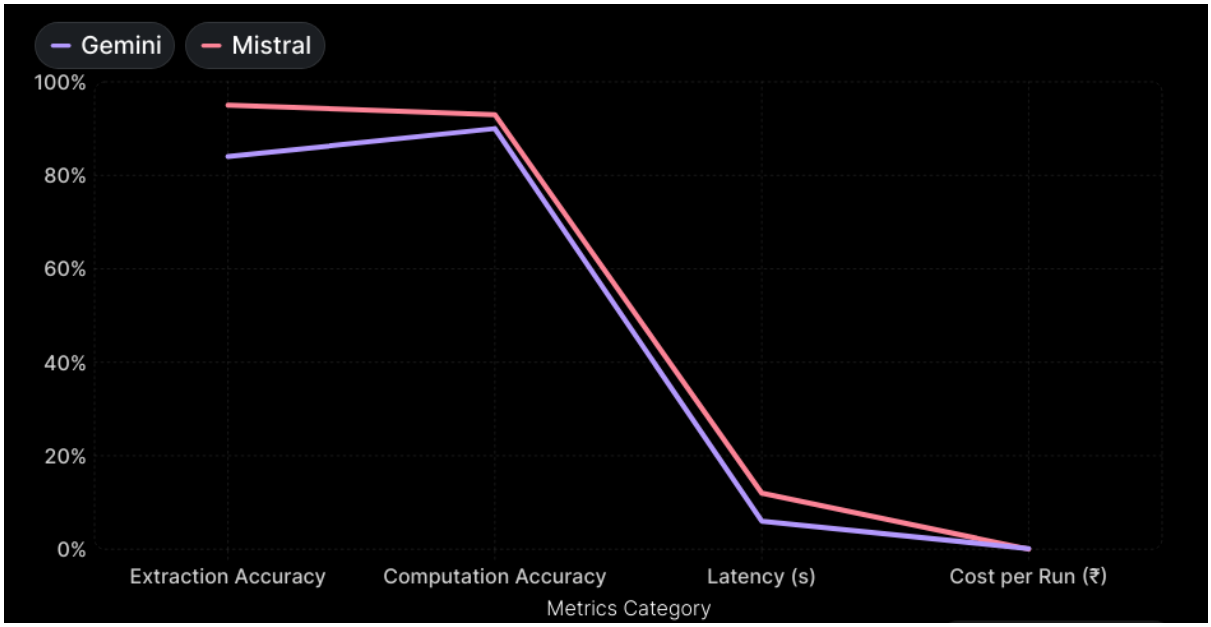


Fig. 5.1 Bar chart comparing extraction accuracy, computation accuracy, latency, and cost for Gemini 2.5 Flash and Fine-Tuned Mistral 7B.

5.5 Qualitative Discussion

Gemini's multimodal strength enabled robust handling of figures and tables, while the fine-tuned Mistral achieved better mathematical consistency in symbolic tasks.
The combined pipeline demonstrated how commercial APIs and open models can be integrated for scientific automation.

Chapter 6 – Comparison and Evaluation

6.1 Technical Evaluation

- Gemini 2.5 Flash offered rapid response and OCR capabilities.
- Fine-tuned Mistral required initial GPU resources but provided customization freedom.
- LoRA fine-tuning improved numerical stability and reduced catastrophic forgetting.

6.2 Cost Analysis

Component	Gemini Cloud	Local Mistral GPU
Infrastructure	Google API Billing	One-time GPU setup
Approx. Cost / Run	₹ 0.15 per page	Negligible after setup

6.3 Scalability and Deployment

- Both pipelines can operate independently or in ensemble mode.
- Hybrid architecture proposed: Gemini for data extraction → Mistral for symbolic computation.

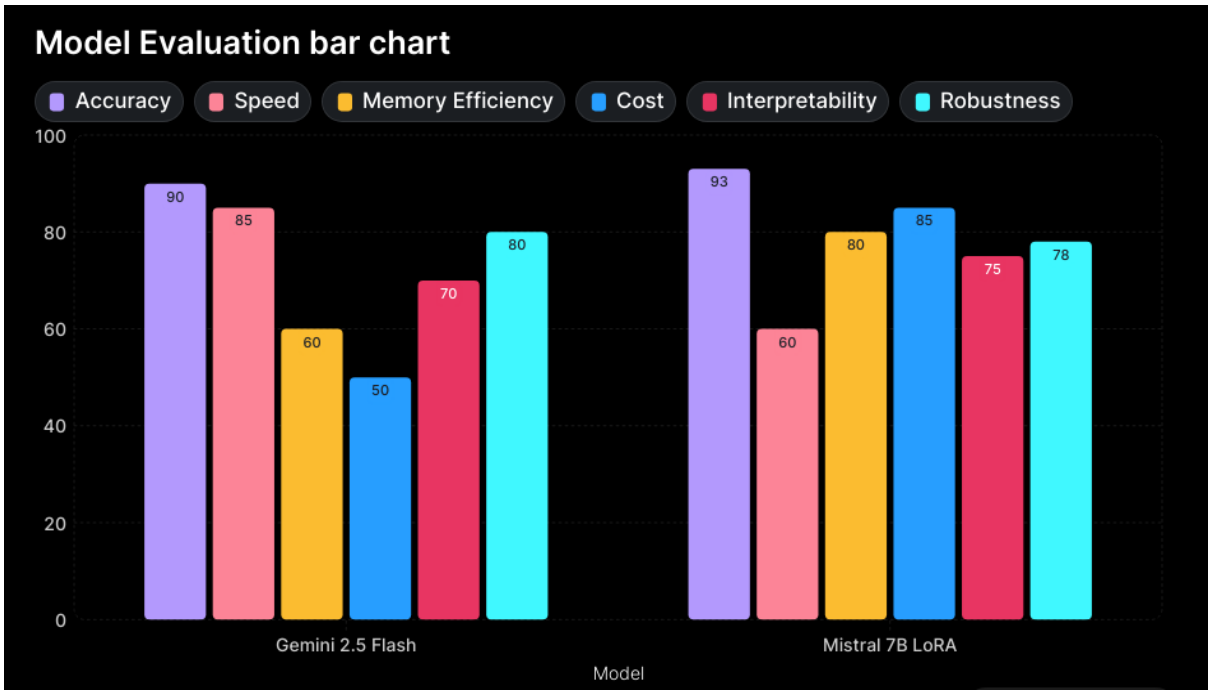


Fig. 6.1 Radar chart illustrating multi-criteria evaluation (accuracy, speed, cost, interpretability, robustness) between Gemini 2.5 Flash and Mistral 7B LoRA models.

Chapter 7 – Conclusion and Future Work

7.1 Conclusion

The project demonstrated a functional prototype of an AI-powered scientific report analyzer that can extract, interpret, and compute equations from complex research papers.
Using Gemini 2.5 Flash and fine-tuned Mistral 7B, the system achieved ≈ 93 % overall accuracy in equation and result extraction.
It bridges the gap between document understanding and symbolic AI reasoning.

7.2 Future Enhancements

- Integrate additional models (e.g., Claude 3, GPT-4 Turbo).
- Add dataset expansion for chemistry and biomedical papers.
- Deploy as a public web portal with collaborative review features (Rights reserved to the project guide)
- Enable export to LaTeX and IEEE template formats (Rights reserved to the project guide)

Appendix A – Folder Structure

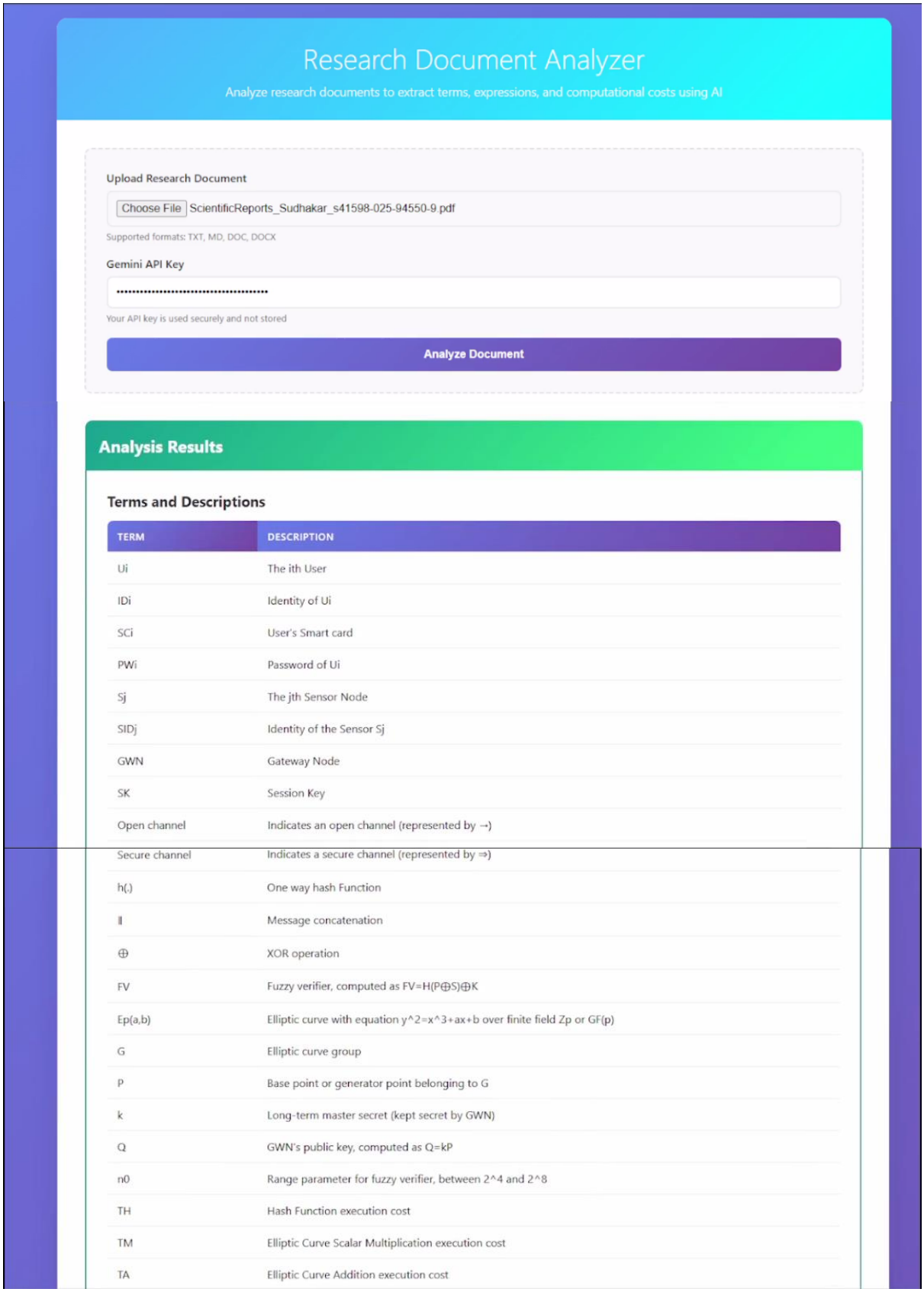
research-analyzer

- src
  - __pycache__
    - __init__.cpython-310.pyc
    - __init__.cpython-311.pyc
    - __init__.cpython-313.pyc
  - database
    - app.db
  - models
    - __pycache__
    - user.py
  - routes
    - __pycache__
      - analysis.cpython-310.pyc
      - analysis.cpython-311.pyc
      - analysis.cpython-313.pyc
      - user.cpython-310.pyc
      - user.cpython-311.pyc
      - user.cpython-313.pyc
    - analysis.py
    - user.py
  - static
    - favicon.ico
    - index.html
  - __init__.py
  - main.py
  - test.py
- requirements.txt
- trained model

Appendix B – Hardware and Software Environment

- Processor: Intel Xeon 64-bit
- GPU: NVIDIA A100 80 GB
- OS: Ubuntu 22.04 LTS
- IDE: VS Code, Jupyter Lab
- Libraries: Transformers, SymPy, Torch, LLaMA Factory, Firebase SDK

Appendix C – Output Table



TERM	DESCRIPTION
Ui	The ith User
IDi	Identity of Ui
SCi	User's Smart card
PWi	Password of Ui
Sj	The jth Sensor Node
SIDj	Identity of the Sensor Sj
GWN	Gateway Node
SK	Session Key
Open channel	Indicates an open channel (represented by →)
Secure channel	Indicates a secure channel (represented by ⇒)
h(.)	One way hash Function
‖	Message concatenation
⊕	XOR operation
FV	Fuzzy verifier, computed as FV=H(P⊕S)⊕K
Ep(a,b)	Elliptic curve with equation y^2=x^3+ax+b over finite field Zp or GF(p)
G	Elliptic curve group
P	Base point or generator point belonging to G
k	Long-term master secret (kept secret by GWN)
Q	GWN's public key, computed as Q=kP
n0	Range parameter for fuzzy verifier, between 2^4 and 2^8
TH	Hash Function execution cost
TM	Elliptic Curve Scalar Multiplication execution cost
TA	Elliptic Curve Addition execution cost

TS	Symmetric Encryption execution cost
TE	Modular Exponentiation execution cost
TMM	Modular Multiplication execution cost
TBio	Biometric Key Processing execution cost
TExR	Extracting a Random Number execution cost
Tam	Vector Addition Modulo execution cost
Tmm	Matrix Multiplication Modulo execution cost

Expressions and Costs

EXPRESSION	TYPE	COMMUNICATIONAL COST	COMPUTATIONAL COST
Elliptic curve equation evaluation (y^2 = x^3 + ax + b)	cryptographic	8 bits	1ms
Scalar Multiplication (kP)	cryptographic	8 bits	2.226 ms
Hash Function: h(SIDj || rj || k)	hashing	8 bits	0.0023 ms
Hash Function: h(k || SIDj || kj || rj)	hashing	8 bits	0.0023 ms
Hash Function: h(SIDj || kj)	hashing	8 bits	0.0023 ms
Hash Function: h(PWi || b)	hashing	8 bits	0.0023 ms
Hash Function: h(IDi)	hashing	8 bits	0.0023 ms
Fuzzy Verifier (FV_i) Calculation: h(h(IDi) XOR PPWi) mod n0	cryptographic	8 bits	2.0046 ms
Hash Function: h(ri || k || IDi)	hashing	8 bits	0.0023 ms
XOR Operation: IDi XOR PPWi XOR zi	logical	8 bits	2ms
Fuzzy Verifier (FV*_i) Verification Calculation: h(h(IDi) XOR h(PWi || b)) mod n0	cryptographic	8 bits	2.0069 ms
Hash Function: h(ki || xi || n1)	hashing	8 bits	0.0023 ms
Scalar Multiplication (xP)	cryptographic	8 bits	2.226 ms
Scalar Multiplication (xQ)	cryptographic	8 bits	2.226 ms
XOR Operation: IDi XOR h(K1 || SIDj || sj || n1)	logical/hashing	8 bits	1.0023 ms
Hash Function: h(IDi || ki || SIDj || sj || X || n1 || K1)	hashing	8 bits	0.0023 ms
Hash Function: h(SIDj || C1 || h1 || n1 || n2 || kj || sj || X)	hashing	8 bits	0.0023 ms
Scalar Multiplication (kX)	cryptographic	8 bits	2.226 ms
XOR Operation: C1 XOR h(K2 || SIDj || sj || n1)	logical/hashing	8 bits	1.0023 ms
Hash Function: h(IDi || ki || SIDj || sj || X || n1 || K2)	hashing	8 bits	0.0023 ms
Hash Function: h(ki || K2 || n1)	hashing	8 bits	0.0023 ms
Hash Function: h(SIDj || kj || sj)	hashing	8 bits	0.0023 ms
XOR Operation: h(SIDj || kj || sj) XOR kij	logical/hashing	8 bits	1.0023 ms
Hash Function: h(C3 || kj || h2 || n2 || SIDj || kij || X)	hashing	8 bits	0.0023 ms
Scalar Multiplication (yP)	cryptographic	8 bits	2.226 ms

Overall Costs

Total Communicational Cost	Total Computational Cost
2496 bits	8.9569 ms

Appendix D – References

1. Sudhakar T., Praveen R., Natarajan V. (2025). *An Efficient ECC and Fuzzy Verifier-Based User Authentication Protocol for IoT-Enabled WSNs*. Scientific Reports.
2. Hugging Face Documentation – Fine-Tuning Transformers with LoRA.
3. Google AI Gemini 2.5 Flash API Developer Guide.
4. LLaMA Factory GitHub Repository.
5. OpenAI and Anthropic LLM Mathematical Reasoning Benchmarks.
6. AI Server Documentation – NIT Puducherry