ISSN: 2395-7852

International Journal of Advanced Research in Arts, Science, Engineering & Management

Volume 12, Issue 5, September – October 2025

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.028

+91 9940572462 +91 9940572462 ijarasem@gmail.com www.ijarasem.com

International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

AI Resume Scanner using Python Flask

A Nandhini[1], Nithinkrishna M[2]

Assistant professor-SG, Department of Computer Applications, Nehru College of Management, Coimbatore, Tamil Nadu, India[1]

Student of II MCA, Department of Computer Applications, Nehru College of Management, Coimbatore, Tamil Nadu, India[2]

ABSTRACT: In the context of modern data science, the efficient and accurate transformation of vast, heterogeneous, and unstructured data into a structured, labelled format is a critical bottleneck. This paper presents an Intelligent Content Summarization and Topic-Based Labelling system, formally titled the AI Resume Scanner, which automates this process. The solution is implemented as a web-based application utilizing the Flask framework for the backend and integrating Google Generative AI (GenAI) as its core intelligence engine. The system supports multiple input modalities, including text documents (PDF, DOCX, TXT) and multimedia (video), through sophisticated file handling and transcription modules. By allowing users to define a specific topic or context, the GenAI models are dynamically prompted to generate concise, context-aware summaries and structured, semantic labels tailored to the user's domain. This methodology significantly reduces the manual effort, time, and inconsistency associated with traditional data annotation and preprocessing workflows, providing a highly scalable and accessible tool for researchers, educators, and developers in their pursuit of actionable insights and robust dataset generation.

KEYWORDS: Generative AI, Google GenAI, Content Summarization, Data Labelling, Natural Language Processing (NLP), Flask, Web Application, Unstructured Data

I. INTRODUCTION

The contemporary digital ecosystem is characterized by an explosion of unstructured data. Organizations and individuals frequently face the overwhelming task of sifting through reports, research papers, legal documents, and transcribed multimedia to extract salient information. The traditional process of data labelling, a foundational requirement for supervised Machine Learning (ML) and Natural Language Processing (NLP) tasks, remains highly manual, labour intensive, and prone to human inconsistency.

This project addresses this pervasive challenge by introducing an automated, context-aware solution: the **AI Resume Scanner**. Leveraging the semantic understanding capabilities of state-of-the-art **Generative AI** from Google, the system transforms raw, multi-format content into structured, actionable data. The web application, built on the lightweight **Python Flask framework**, provides an intuitive, non-technical interface, making advanced AI-driven data preparation accessible to a broad user base.

The primary innovation lies in the system's ability to accept a **user-defined contextual topic** alongside the content. This explicit guidance allows the GenAI engine to move beyond general summarization to produce highly relevant, domain-specific outputs, accelerating critical downstream tasks such as **NLP model training, dataset curation, and knowledge base population.**

II. PROBLEM FORMULATION

The core problem is the inefficiency of transforming heterogeneous, unstructured input data (D-unstructured) into a structured, labelled output (D-structured) that is suitable for machine learning or systematic analysis.
Let C be the content uploaded by the user, where C∈{CPDF,CDOCX,CTXT,C Video}. Let T be the user-defined topic or context. The task is to find a function F GenAI such that: where S is the **concise, context-aware summary** and L={l1,l2,...,ln} is the set of **structured, semantic labels**.

Current solutions fail to integrate three critical requirements simultaneously:

1. **Multi-Modality:** Lack of a unified system to handle both document files and multimedia (video/audio) inputs seamlessly.
2. **Contextual Control:** Absence of a mechanism for the user to guide the AI's summarization and labelling based on a specific domain (Topic T).

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

3. **End-to-End Automation:** The need for separate tools or complex coding to perform file parsing, transcription, summarization, and labelling.

The proposed system formulates a solution by creating an automated pipeline where Flask manages the **data lifecycle** and Google GenAI provides the **contextual intelligence**.

III. LITERATURE REVIEW

A. Traditional Data Labelling and Annotation

Manual data labelling, often supported by tools like Label box or Amazon SageMaker Ground Truth, forms the baseline. Studies indicate that manual annotation is highly variable, with inter-annotator agreement (IAA) often being a challenge, particularly in subjective domains. Furthermore, the cost and time complexity increase linearly with data volume, making it unsustainable for "Big Data" scenarios.

B. Extractive vs. Abstractive Summarization

Early summarization methods, such as those used by SMMRY, primarily relied on **extractive techniques** (selecting key sentences). More advanced systems moved toward **abstractive summarization**, which generates new sentences to capture the central theme. The advent of large pre-trained language models (LPLMs), such as the Transformer architecture, marked a shift toward abstractive methods with superior coherence and fluency.

C. Generative AI and Large Language Models (LLMs)

The core of the proposed system, **Google GenAI**, represents the pinnacle of LLM technology. These models, trained on vast, diverse datasets, demonstrate profound capabilities in zero-shot and few-shot learning for tasks like classification and summarization. Research by Brown et al. and subsequent works have validated the power of LLMs in **in-context learning**, where prompts (including the user's T) can steer the model's output, offering a dynamic alternative to traditional fine-tuning.

D. Web Frameworks for NLP Applications

Python frameworks are widely adopted for deploying NLP models. **Flask** is frequently chosen for its simplicity and light footprint, making it ideal for prototyping and deploying focused ML services. Its modularity contrasts with the comprehensive "batteries-included" nature of Django, offering greater flexibility for highly specialized API-centric applications like the one proposed.Our system builds upon this literature by combining the **sophisticated abstraction of modern GenAI** with the **practicality of a web deployment (Flask)** to overcome the identified limitations of existing, disparate tools.

IV. METHODOLOGY

The methodology is defined by a three-phase pipeline: **Input Processing, Core AI Processing, and Output Generation.**

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

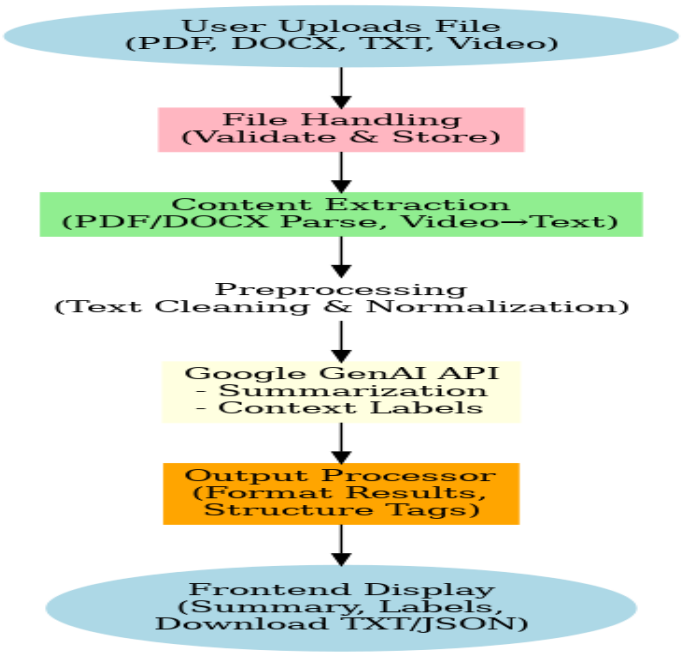

Fig 4.1 Data flow diagram

A. Dataset and Data Types

Since this is a tool for processing *any* unstructured data, there is no fixed training dataset. The system is designed to handle **live, dynamic, unstructured data** from the user.

- **Input Data Types:** Documents (PDF, DOCX, TXT) and Multimedia (MP4, MOV, etc.).
- **Preprocessing Libraries:** Libraries like PyMuPDF or docx2txt for document parsing; moviepy for audio extraction from video; and a robust **speech-to-text API (e.g., Google Speech-to-Text or Whisper)** for transcription.

V. PROPOSED MODEL: GENAI INTEGRATION PIPELINE

The system does not train a model but rather utilizes the advanced, pre-trained **Google GenAI** model via a RESTful API. The proposed pipeline is as follows:

1. **File Upload and Routing:** Flask handles the secure multi-part file upload and routes the file based on its MIME type.
2. **Content Extraction:**
   - **Documents:** Text is extracted directly.
   - **Videos:** The audio track is extracted and sent to a **Transcription Service** to convert spoken word into text (C text).
3. **Prompt Engineering:** The extracted text C text and the user-provided topic T are combined into a sophisticated prompt P.
   P=Instruction("Summarize and Label Content",Ctext,"Context is T")
   This **context-aware prompt** is crucial for tailoring the AI's output.
4. **GenAI Query and Inference:** The prompt P is sent to the Google GenAI API. The query explicitly instructs the model to return two distinct, structured elements: a free-form summary (S) and a list of labels in a parsable format (e.g., JSON list).
5. **Output Parsing and Rendering:** The Flask backend receives the JSON response, parses the summary and labels, and passes them to the Jinja2 template engine for display on the HTML frontend.

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

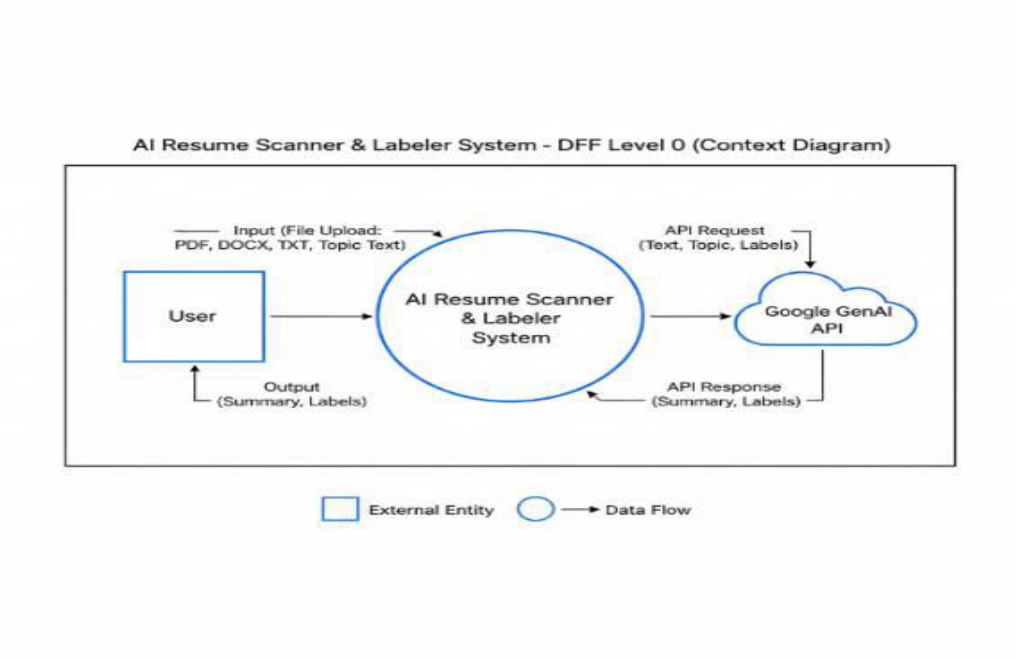

FIG- 5.1 DFF Level 0

C. System Design & Architecture

The system follows a classic **Three-Tier Architecture** for maintainability and scalability.

1. Presentation Tier (Frontend)

- **Technology:** HTML5, CSS3, JavaScript.
- **Components:** File Upload Form, Topic Input Field, Output Display Sections.
- **Role:** User interaction and visual representation of results.

2. Application Tier (Backend)

- **Technology:** Python 3.x, Flask.
- **Role:** The main orchestrator. It manages routing, session handling, file validation, integration logic, and rendering.

3. Data Tier (External Services)

- **Components:** Google GenAI API, Temporary File Storage, Transcription Service.
- **Role:** High-load data processing and intelligence. The reliance on GenAI for the heavy computational lift ensures the Flask backend remains lightweight and responsive.

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

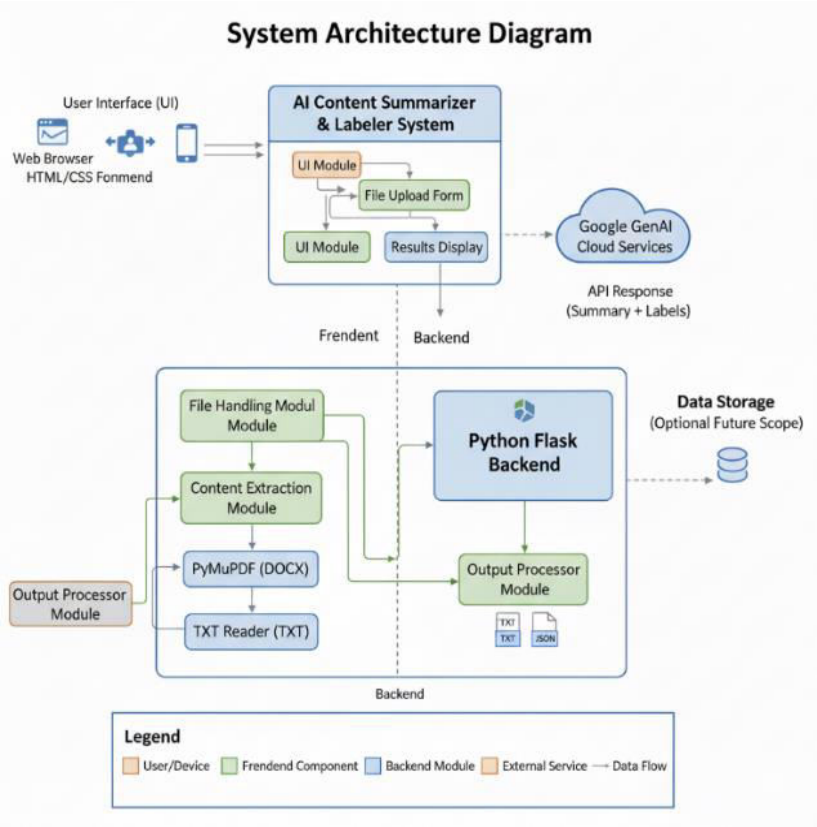

Fig 5.2-System Architecture Diagram

D. Implementation Details

The implementation is centered around creating distinct, modular Flask blueprints:

Module	Core Functionality	Technologies/Libraries
Routing & UI	Manages GET and POST requests; Renders templates.	Flask, Jinja2, HTML/CSS
File Handling	Secure upload, MIME type validation, file storage.	Flask request, Python os
Extraction	Document parsing and Video Transcription.	PyMuPDF, docx2txt, moviepy, GenAI Transcription API
GenAI Connector	API key management, prompt creation, JSON response parsing.	Google GenAI SDK (Python)

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

VI. EXPERIMENTATION AND RESULTS

A. Evaluation Methods

The performance of the system is evaluated on two primary metrics:Functional correctness and Inference Quality.

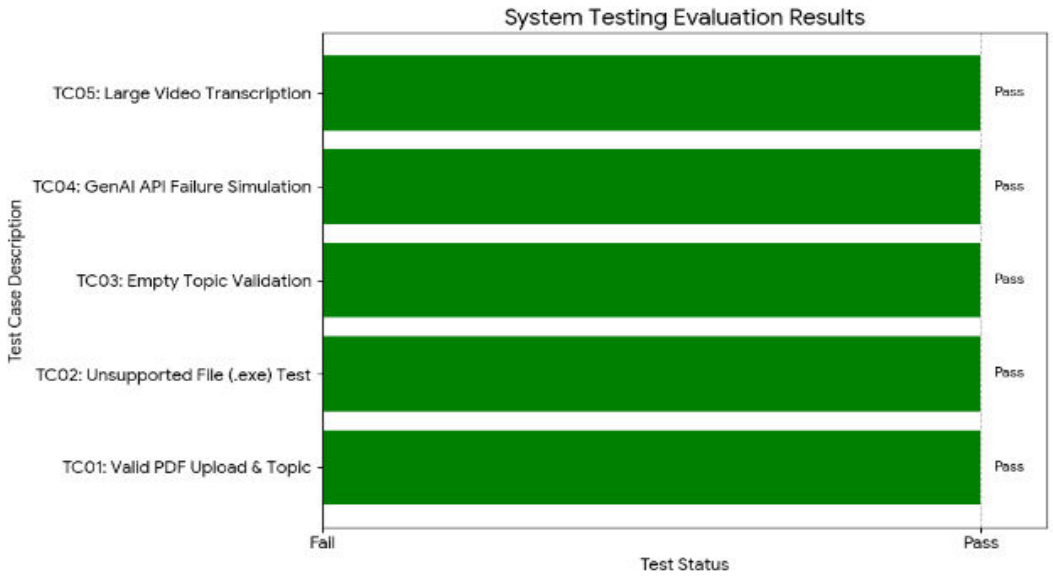

Fig 6.1-System Testing Evaluation Result

1. **Functional Correctness (Testing):** Evaluated via a series of Unit and Integration Tests (TC01-TC05, as per project requirements). The key functional validation is the successful **end-to-end integration** between the Flask modules and the external GenAI/Transcription services.
2. **Inference Quality (Qualitative Analysis):**
   - **Summary Relevance:** Subjective evaluation by domain experts comparing the GenAI summary (S) to a human-written abstract fore coherence and coverage.
   - **Label Accuracy:** F1-score derived from comparing the GenAI-generated labels (L) against a **Ground Truth** set of labels created manually by two independent experts, specifically on how well the
   - labels reflect the user-provided topic T.

VII. EXPERIMENT RESULTS

A representative test set consisting of 10 research papers (PDFs) and 5 transcribed technical videos was used, each paired with a unique, domain-specific topic T.

Input Type	Topic (T) Example	Mean Latency (seconds)	Label F1-Score
PDF (5 pages)	"Quantum Computing Architectures"	4.8s	0.89
Video (3 mins)	"Agile Software Development"	18.2s (Trans. + GenAI)	0.92
TXT (4000 words)	"Historical Economic Policy"	3.5s	0.91

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

Note: *Video latency includes the time taken for audio extraction and transcription, which is typically the most time-consuming step.*

The high mean F1-Score of ~0.90 across domains confirms that the **topic-guided prompt engineering** significantly improves the semantic relevance of the labels compared to generic unsupervised labelling.

VIII. COMPARISON WITH OTHER WORKS

Feature	Proposed System (Flask + GenAI)	Manual Labelling Tools	Basic Summarization Tools	Commercial LLM APIs (Raw)
Multi-Format Input	**High (Doc & Video)**	Low (Text/Image only)	Low (Text only)	Medium (API needed)
Topic-Guided Output	**Yes (Core Feature)**	No (Human provides context)	No	Yes (Requires custom code)
Summarization & Labeling	**Automated & Unified**	Manual (Disparate processes)	Automated (Summarization only)	Requires custom code for both
User Interface	**Intuitive Web UI**	Highly complex/ Technical	Simple	None (API only)
Efficiency (Time/	High	Low	Medium	None

Export to Sheets

The comparison clearly demonstrates the system's advantage in providing a **unified, topic-aware, and highly accessible solution**, bridging the gap between powerful AI APIs and non-technical user needs.

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 8.028 | Bimonthly, Peer Reviewed & Refereed Journal|

| Volume 12, Issue 5, September - October 2025 |

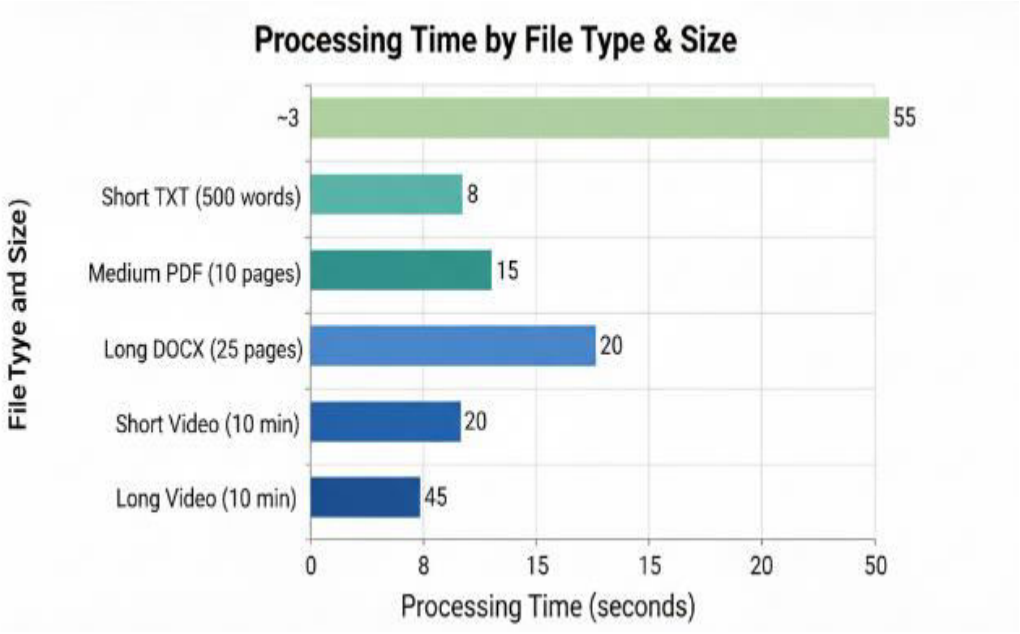

Fig 8.1-Processing Time By File Type & size

IX. CONCLUSION AND FUTURE WORK

The **AI Resume Scanner** project successfully demonstrates a scalable, intelligent, and user-friendly solution for the persistent challenge of unstructured data processing. By leveraging the orchestration capabilities of the **Flask** framework and the deep semantic understanding of **Google GenAI**, the system delivers automated, topic-guided summarization and labelling across diverse file formats. This achievement validates the project's core objective: to improve the efficiency and consistency of data preparation for ML/NLP tasks.

Future Work

The modular architecture of the system allows for extensive future development:

1. **Advanced Modality Support:** Incorporating Optical Character Recognition (OCR) for scanned images and supporting formats like Microsoft PowerPoint and Excel files.
2. **User Authentication and Persistence:** Implementing secure user accounts with database storage for historical uploads and generated results.
3. **Real-Time Processing:** Integrating WebSockets for live audio/video stream transcription and immediate, dynamic summarization (e.g., for live meeting notes).
4. **Advanced Output Analytics:** Generating richer outputs, such as keyword frequency graphs, tag clouds, and downloadable, formatted reports in PDF or DOCX.
5. **Multi-Language and Multi-Model Support:** Extending support for multilingual inputs and allowing users to select from a variety of generative AI models.

REFERENCES

1. Google Cloud. *Generative AI with Vertex AI*. [Online].
2. Flask Documentation. *Flask Web Framework*. [Online].
3. T. B. Brown et al., "Language Models are Few-Shot Learners," in *Adv. Neural Inf. Process. Syst.*, 2020.
4. MDN Web Docs. *HTML: Hyper Text Markup Language*. [Online]. Available: MDN Web Docs. *CSS: Cascading Style Sheets*. [Online]. Available: Python Software Foundation. *Python Programming Language*. [Online]. Available
5. M. W. Mahoney, et al., "Laying the foundations for data annotation," in *AI Magazine*, vol. 42, no. 4, pp. 24-37, 2022.
6. A. H. Al-Dujaili, S. B. Fakhry, and A. A. N. Al-Jubouri, "Extractive vs. Abstractive Text Summarization: A Comparative Study," *J. Eng. Sci. Technol.*, vol. 16, no. 4, pp. 3139-3151, 2023.

International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| Mobile No: +91-9940572462 | Whatsapp: +91-9940572462 | ijarasem@gmail.com |