

# **SRM VALLIAMMAI ENGINEERING COLLEGE**

**(An Autonomous Institution)**

SRM Nagar, Kattankulathur, Chengalpattu Dt.-603203, Tamil Nadu.

## **DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**ACADEMIC YEAR: 2025-2026**

**REGULATION 2019**

**REPORT**

**on**

**“1922709-INTERNSHIP”**

**at**

**JAGS CONSULTANCY LTD,  
BENGALURU, KARNATAKA.**



**SUBMITTED BY**

**142222243036 – RAAHULESHWAR A**

**SEVENTH SEMESTER**

**2022- 2023 (Odd Semester)**

**BATCH: 2022-2026**

# **SRM VALLIAMMAI ENGINEERING COLLEGE**

**(An Autonomous Institution)**

SRM Nagar, Kattankulathur – 603 203

## **DEPARTMENT OF INFORMATION TECHNOLOGY**

### **CERTIFICATE**

This is to certify that Mr. Raahuleshwar A(Reg. No.142222243036) VII Semester B. Tech (ARTIFICIAL INTELLIGENCE & DATA SCIENCE) has completed the Internship entitled '**TEXT SUMMARIZATION AND INSURANCE ANALYSIS USING LLMs**' during the 3rd June 2025 - 4<sup>th</sup> July 2025 at JAGS CONSULTANCY LTD, BENGALURU, KARNATAKA and the has been submitted to Anna University.

**HEAD OF THE DEPARTMENT**

**SIGNATURE**

**DEPT. OF ARTIFICIAL INTELLIGENCE**

**NAME :**

**& DATA SCIENCE**

**DESIGNATION :**

**SRM VALLIAMMAI ENGG. COLLEGE**

## **ACKNOWLEDGEMENT**

I am highly indebted to Director **Dr. B. Chidhambararajan, M.E., Ph.D.** and Principal **Dr. M. Murugan, M.E., Ph.D.** for the facilities provided to accomplish this internship.

I would like to thank my Head of the Department **Dr. B. Muthusenthil, B.E., M.E., Ph.D. Associate Professor** for his constructive criticism throughout my internship.

I would like to thank **R. Vaishnavi B.E., M.E., Assistant Professor (OG)** and internship coordinator Department of AI&DS for her support, advices to get and complete internship in the organization.

I am extremely grateful to my department staff members and friends who helped me in successful completion of this internship.

Also, I would like to thank Mr. **Abinandhanan Sripal Udhaykumar** (Managing Partner) of **JAGS CONSULTANCY LTD** for giving me the opportunity to do an internship within the organization.

I also would like to thank all the people that worked along me in **JAGS CONSULTANCY LTD** with their enthusiasm and assurance which created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

**RAAHULESHWAR A**

(142222243036)

# **SRM VALLIAMMAI ENGINEERING COLLEGE**

*(An Autonomous Institution)*

SRM Nagar, Kattankulathur, Chengalpattu Dt.-603203, Tamil Nadu.

## **DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

### **VISION OF THE DEPARTMENT**

To become a model for Artificial Intelligence with innovation and analysis for higher learning through various analytical knowledge, creative competent and dynamic technocrats; while remaining responsive to ethical, societal, and environmental issues.

### **MISSION OF THE DEPARTMENT**

**M1:** To develop the students as Artificial Intelligence designers and data analyst professionals to meet the global design challenges and entrepreneurs of international excellence as global leaders capable of contributing towards technological innovations, learning processes, participation citizenship in their neighbourhood and economic growth.

**M2:** To transform value-based data science education to the students and groom them as leaders in the field of Artificial Intelligence and Data Science for the empowerment of society.

## TABLE OF CONTENT

<b>Sl. No</b>	<b>Description</b>	<b>Number of Pages</b>
1.	Front Page & Sheet	1
2.	Acknowledgement	3
3.	Vision & Mission	4
4.	Table of Contents	5
5.	Introduction – about Internship	6
6.	Technology Learnt – topics covered	7
7.	Details of project work carried out	11
8.	Skills Learnt and Experience gained	21
9.	Conclusion	23
10.	Internship Certificate	25

### **ANNEXURE**

i	Permission Letter	26
ii	About the Industry / Company Profile/ address/ contact Person	27
iii	Schedule copy	28
iv	Sample Photos	29

## **INTRODUCTION - ABOUT INTERNSHIP**

The internship at JAGS Consultancy Ltd provided me with a valuable opportunity to experience the real-time working environment of a professional consultancy firm. JAGS Consultancy Ltd is a dynamic and evolving organization that offers specialized services in business consulting, IT solutions, data management, software development, and process automation.

The company is known for its commitment to delivering high-quality, client-focused solutions by integrating cutting-edge technologies with strategic planning. During the internship period, I was able to closely observe and participate in the daily operations of the organization, working alongside a team of skilled professionals who were supportive and collaborative. The work culture at JAGS Consultancy Ltd was highly professional yet approachable, creating an environment conducive to learning and growth. I was entrusted with tasks that allowed me to explore real-world business problems, understand client requirements, and support ongoing projects involving digital transformation and data-driven decision-making.

The hands-on experience enabled me to gain insights into how consultancy firms operate, how technology is applied to solve organizational challenges, and how effective communication and teamwork play a vital role in delivering successful outcomes. This internship not only enhanced my technical skills but also provided exposure to project management, industry expectations, and professional ethics. It was an enriching journey that contributed significantly to my personal and professional development.

## **TECHNOLOGY LEARNT**

### **NATURAL LANGUAGE PROCESSING (NLP)**

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and human languages. It combines techniques from linguistics, computer science, and machine learning to enable machines to understand, interpret, and generate human language in a meaningful way. NLP is used in various applications such as speech recognition, machine translation, sentiment analysis, chatbots, text summarization, and more.

Core tasks in NLP include tokenization (breaking text into words or sentences), part-of-speech tagging (identifying nouns, verbs, etc.), lemmatization (reducing words to their base form), and named entity recognition (identifying proper names, dates, etc.). More advanced NLP involves syntactic parsing, semantic analysis, and context-aware language modeling using deep learning.

Recent advancements in NLP have been driven by transformer-based models like BERT, GPT, and T5, which learn deep contextual representations of language from large datasets. These models have significantly improved the performance of tasks like question answering, language translation, and text generation.

NLP is essential in making unstructured text data understandable to machines. It enables systems to extract insights, automate decisions, and interact with users more naturally, making it a critical technology in modern AI solutions.

## OCR AND TEXT EXTRACTION

Optical Character Recognition (OCR) is a technology that converts printed, scanned, or handwritten text from documents and images into machine-readable text. It allows computers to recognize characters, words, and structures in static visual formats and transform them into editable digital content. OCR is widely used for digitizing printed documents, invoices, books, ID cards, and forms.



The OCR process involves image preprocessing (such as noise reduction, binarization, and skew correction), text detection, character segmentation, and character recognition. Popular OCR engines like Tesseract use machine learning techniques to identify character patterns and convert them into text. Some advanced systems also integrate Natural Language Processing to enhance the accuracy of recognition in complex layouts or noisy images.

Text extraction refers to retrieving meaningful text from digital documents like PDFs, HTML pages, or image files. In addition to OCR, text extraction may use tools like PyMuPDF, PDFPlumber, or BeautifulSoup to extract structured or unstructured data.

OCR and text extraction are essential for converting unstructured data into usable forms, enabling automation in document processing tasks. These technologies are especially useful when dealing with large volumes of static content where manual data entry would be time-consuming, error-prone, and inefficient.

## DATA PREPROCESSING

Data preprocessing is a fundamental step in any machine learning or natural language processing workflow. It involves cleaning, formatting, and transforming raw data into a structured and usable format that models and algorithms can understand. In the context of text-based tasks, preprocessing ensures consistency, removes noise, and enhances the overall quality of the input.

For textual data, preprocessing often starts with **tokenization**, which breaks down text into words or sentences. **Stop-word removal** eliminates common but uninformative words like “and,” “is,” and “the.” **Lemmatization** or **stemming** reduces words to their base or root form, helping generalize variations (e.g., “running” → “run”). Additional steps include removing special characters, punctuations, numbers, and converting text to lowercase.

In some workflows, **normalization** is used to standardize formats (e.g., date or currency), and **noise filtering** helps clean spelling errors or irrelevant symbols. Preprocessed text improves the performance and accuracy of models by reducing redundancy, handling inconsistencies, and ensuring that only meaningful data is analyzed.

Without proper preprocessing, models may misinterpret or overfit irrelevant patterns in the data. This step is critical for enhancing the reliability, efficiency, and accuracy of downstream tasks like classification, summarization, or information extraction.

## PYTHON AND MACHINE LEARNING TOOLS

Python is one of the most widely used programming languages in artificial intelligence and machine learning due to its simplicity, readability, and vast ecosystem of libraries. It provides a flexible environment for developing and deploying machine learning models, data pipelines, and intelligent applications.

In the context of machine learning, Python supports numerous libraries that simplify model development and data handling. **NumPy** and **Pandas** are used for numerical computations and data manipulation, while **Matplotlib** and **Seaborn** help with data visualization. **Scikit-learn** is a core library for traditional machine learning algorithms such as classification, regression, and clustering. It offers tools for model training, evaluation, and validation.

For deep learning tasks, Python integrates with powerful frameworks like **TensorFlow**, **PyTorch**, and **Keras**, which allow developers to build neural networks, handle large datasets, and fine-tune pre-trained models. In natural language processing, libraries such as **spaCy**, **NLTK**, and **Transformers (by Hugging Face)** offer tools to work with text data and language models.

Python's large community support and rich documentation make it ideal for both beginners and professionals. Its versatility across data science, automation, and web development also makes it a preferred choice for building end-to-end machine learning solutions efficiently and effectively.

## **DETAILS OF PROJECT WORK CARRIED OUT**

**TOPIC :** Text Summarizer and Insurance Policy Analyzer

### **TEXT SUMMARIZER :**

The **Text Summarizer** is a key component of the project designed to automatically condense lengthy insurance policy documents into shorter, meaningful summaries. Insurance policies are typically long and complex, often filled with technical and legal terms that make it difficult for clients to quickly understand the important details. The text summarizer addresses this challenge by using Natural Language Processing (NLP) to generate clear, concise versions of these documents.

#### **Functionality:**

The summarizer processes the extracted and cleaned text from insurance documents and reduces it to a few coherent sentences that highlight the most critical information. These include policy coverage, conditions, premium amount, and exclusions.

There are two main types of summarization techniques:

- **Extractive Summarization:** Selects and combines key sentences directly from the original document.
- **Abstractive Summarization:** Generates new sentences that paraphrase the original content using deep learning models.

#### **Implementation Approach:**

- **Text Preprocessing:** The input text was cleaned using techniques like tokenization, stop-word removal, and lemmatization.

- **Model Selection:** Transformer-based models such as T5 or BART were used for abstractive summarization due to their strong performance in generating human-like summaries.
- **Summarization Output:** The model produced 3-5 sentence summaries tailored for user readability while preserving the original context.

## **Outcomes and Benefits:**

The summarizer helped reduce information overload and made insurance policies more accessible to users. It enabled quick comprehension and better decision-making by presenting only the most essential content. The summarization module also laid the foundation for further enhancements, such as multilingual support or personalized summary generation.

## **INSURANCE POLICY ANALYZER:**

The **Insurance Policy Analyzer** is a system designed to intelligently extract and interpret key information from insurance documents. Insurance policies often contain dense legal text, technical terms, and extensive conditions that are not easily understood by most users. The goal of the analyzer is to automatically read such documents, identify important fields, and organize them in a clear, structured format for easy reference.

### **Core Functionalities:**

#### **1. Field Identification:**

The analyzer extracts essential details such as:

- Policy number
- Policyholder name
- Coverage start and end dates

- Premium amount
- Sum insured
- Policy exclusions and conditions
- Claim process instructions

## 2. Pattern Recognition and Tagging:

Rule-based matching and regular expressions were used to locate consistent patterns (e.g., dates, monetary values, percentage rates). For more complex fields like exclusions, keyword-based identification and linguistic patterns were used.

## 3. Text Structuring:

After extraction, the information was reorganized and labeled clearly using JSON or tabular formats, making it easier to feed into applications, dashboards, or client reports.

## 4. Validation:

Checks were implemented to ensure the correctness of extracted fields, especially in identifying duplicate sections or noise caused by OCR errors.

## Approach and Tools Used:

- **Text Extraction:** Used PyMuPDF and Tesseract OCR depending on the format of the policy document.
- **Regex & Rule-Based Extraction:** Custom scripts were written to capture specific data points from raw text using known patterns.
- **NLP Techniques:** Lemmatization, named entity recognition (NER), and POS tagging were used to extract semi-structured content.
- **Data Cleaning:** Removed formatting issues, unwanted characters, headers/footers, and unstructured noise.

## **Why This Matters:**

This analyzer component plays a critical role in **automating insurance workflows**. It reduces the need for humans to manually read and interpret long policies. It can be used to quickly generate customer summaries, feed CRM systems, or trigger alerts if certain clauses are present (e.g., high deductibles or missing coverage).

## **Learning Highlights:**

Through building the policy analyzer, I:

- Improved my skills in text pattern recognition and entity extraction
- Gained experience in handling real-world unstructured documents
- Learned to balance rule-based logic with NLP for higher accuracy
- Understood the challenges of document variability and OCR limitations
- Strengthened my confidence in designing modular AI systems

# PROJECT REQUIREMENTS

## SOFTWARE REQUIREMENTS

- **Operating System:** Windows 10 / Linux / macOS
- **Python Version:** 3.8 or above
- **Required Libraries:**
  - numpy, pandas
  - nltk, spacy
  - pytesseract
  - transformers
  - fitz (PyMuPDF)
  - regex
- **OCR Engine:** Tesseract OCR (open-source)

## HARDWARE REQUIREMENTS

- **Processor:** Intel Core i5 or equivalent
- **RAM:** Minimum 8 GB
- **Storage:** At least 2 GB of free disk space
- **GPU (Optional):** For faster model inference (recommended for abstractive summarization)

## CODE SNIPPET

```
#app.py
import streamlit as st
import requests
# FastAPI endpoint
API_URL = "http://127.0.0.1:8000/summarize/"

st.set_page_config(page_title="Smart Document Summarizer", layout="centered")
st.title("📄 Smart Document Summarizer with Gemini + Word2Vec")

# File upload
uploaded_file = st.file_uploader("Upload a PDF, DOCX, or Image", type=["pdf", "docx",
"png", "jpg", "jpeg"])

# Query input
query = st.text_input("Enter your question or focus point for the summary")

# Style selection
style = st.selectbox("Choose Summary Style", ["Detailed", "Bullet Points", "Short"])

# Submit button
if st.button("Summarize"):
    if uploaded_file is None or query.strip() == "":
        st.warning("Please upload a file and enter a query.")
    else:
        with st.spinner("Summarizing... 🕒 "):
            try:
                files = {"file": (uploaded_file.name, uploaded_file.read())}
                data = {"query": query, "style": style}
                response = requests.post(API_URL, files=files, data=data)

                if response.status_code == 200:
                    summary = response.json()["summary"]
                    st.subheader("📝 Summary")
                    st.write(summary)
                else:
                    st.error(f"Error: {response.json().get('error', 'Unknown error')} ")
            except Exception as e:
                st.error(f"Something went wrong: {e} ")

```

## ANALYSIS

The Smart Document Summarizer is an intelligent system designed to extract, understand, and summarize documents based on user-defined queries. It provides a seamless interface through Streamlit for users to upload various document formats such as PDFs, DOCX files, and images (JPEG, PNG). The system intelligently handles both digital text and scanned documents using OCR (Optical Character Recognition) to ensure maximum text retrieval from any format. Once the document is uploaded, it is processed in the backend using a FastAPI server. The system reads the file and extracts the raw content using different techniques based on the file type. For DOCX files, it uses python-docx to read the paragraphs. PDFs are handled using PyMuPDF, and any embedded or scanned images are processed using Tesseract OCR. For standalone image files, the text is extracted using OCR through the PIL and pytesseract libraries.

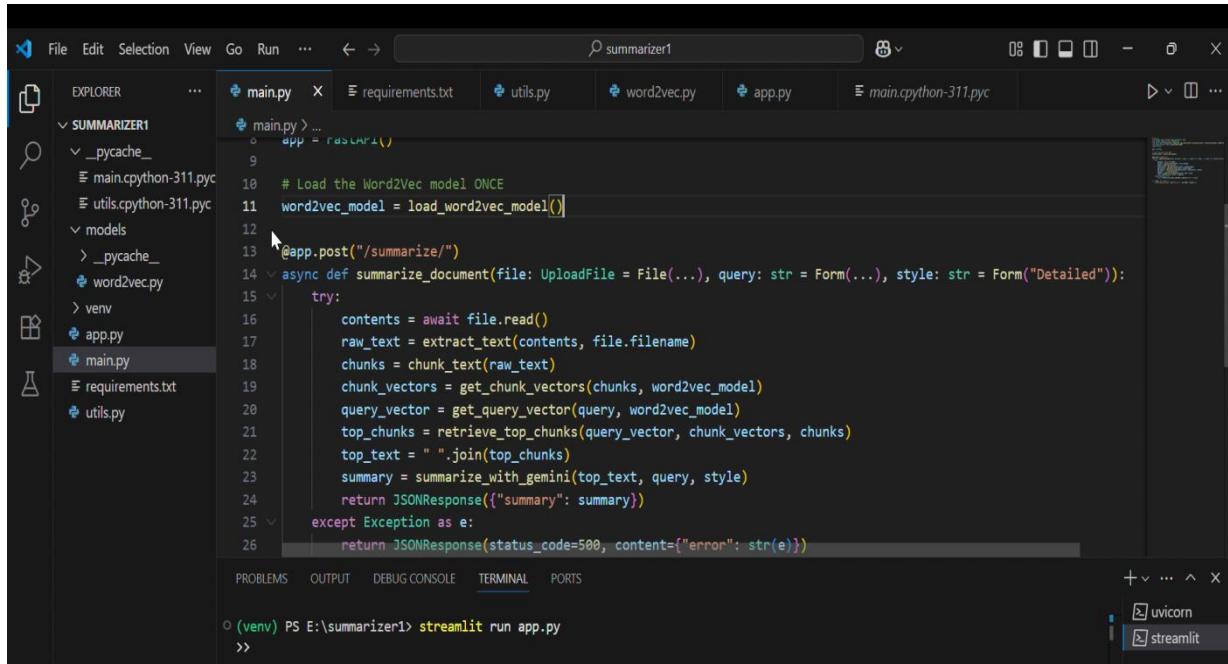
The extracted text is then preprocessed and split into manageable chunks of about 100 words each. These chunks are converted into vector representations using the pretrained Word2Vec model (Google News 300-dimension). The user's query is also vectorized in the same way. The system then calculates the cosine similarity between the query vector and each text chunk to identify the top 10 chunks that are semantically most relevant to the user's focus.

These top-ranked chunks are then passed into a prompt engineered for Google's Gemini language model. Depending on the user's selected style — whether "Detailed," "Bullet Points," or "Short" — a tailored prompt is constructed. Gemini uses these prompts and the relevant text to generate a coherent, context-aware summary. This approach ensures that the summary directly answers the user's specific query, rather than giving a generic document overview.

The output is then returned to the frontend, where it is displayed in a readable format. Users benefit from a customized summary that aligns with their interest or intent, without needing to read the entire document. This system is particularly powerful for navigating complex or lengthy documents, making it ideal for domains like insurance, law, education, or research.

The integration of traditional NLP (via Word2Vec) and modern generative AI (via Gemini) provides both relevance and fluency in summarization. The project showcases effective coordination between multiple technologies — OCR, semantic search, and generative models — tied together in a scalable and user-friendly pipeline. It also demonstrates the real-world application of AI in making unstructured data actionable and accessible.

## RESULTS AND FINDINGS:



A screenshot of a code editor (VS Code) showing a Python file named `main.py`. The code implements a document summarization service using FastAPI, Word2Vec, and Gemini. The editor interface includes a sidebar with project files like `requirements.txt`, `utils.py`, and `word2vec.py`, and a terminal at the bottom showing the command `streamlit run app.py` being run.

```
File Edit Selection View Go Run ... ⏪ summarizer1 ⏹ 🔍 main.py requirements.txt utils.py word2vec.py app.py main.cpython-311.pyc ⏵ ... 🔍 EXPLORER 🔍 SUMMARIZER1 🔍 _pycache_ 🔍 main.cpython-311.pyc 🔍 utils.cpython-311.pyc 🔍 models 🔍 _pycache_ 🔍 word2vec.py 🔍 venv 🔍 app.py 🔍 main.py 🔍 requirements.txt 🔍 utils.py main.py > ... app = FastAPI() 9 10 # Load the Word2Vec model ONCE 11 word2vec_model = load_word2vec_model() 12 13 @app.post("/summarize/") 14 async def summarize_document(file: UploadFile = File(...), query: str = Form(...), style: str = Form("Detailed")): 15 try: 16     contents = await file.read() 17     raw_text = extract_text(contents, file.filename) 18     chunks = chunk_text(raw_text) 19     chunk_vectors = get_chunk_vectors(chunks, word2vec_model) 20     query_vector = get_query_vector(query, word2vec_model) 21     top_chunks = retrieve_top_chunks(query_vector, chunk_vectors, chunks) 22     top_text = " ".join(top_chunks) 23     summary = summarize_with_gemini(top_text, query, style) 24     return JSONResponse({"summary": summary}) 25 except Exception as e: 26     return JSONResponse(status_code=500, content={"error": str(e)}) PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS o (venv) PS E:\summarizer1> streamlit run app.py >>
```

## Smart Document Summarizer with Gemini + Word2Vec

Upload a PDF, DOCX, or Image



Drag and drop file here

Limit 200MB per file • PDF, DOCX, PNG, JPG, JPEG

Browse files



Applications of Digital Image Processing in Medical Imaging.pdf 432.7KB



Enter your question or focus point for the summary

summarize about Mammography

Choose Summary Style

Bullet Points

Summarize

Summarizing... 📐

Summarize about Mammography

Choose Summary Style

Bullet Points

Summarize

## Summary

Digital image processing significantly enhances various medical imaging modalities. Here's a summary focusing on Mammography:

**Mammography:**

- **Contrast Enhancement:** Improves visibility of microcalcifications and tumors.
- **Computer-Aided Detection (CAD):** AI-based systems assist radiologists in identifying potential abnormalities.
- **Image Segmentation:** Isolates breast tissues and lesions for detailed analysis.
- **Noise Reduction:** Filters remove grainy textures while preserving crucial details.

The provided text also details the applications of digital image processing in other medical imaging

Summarize about Mammography

Choose Summary Style

Bullet Points

Summarize

## Summary

Digital image processing significantly enhances various medical imaging modalities. Here's a summary focusing on Mammography:

**Mammography:**

- **Contrast Enhancement:** Improves visibility of microcalcifications and tumors.
- **Computer-Aided Detection (CAD):** AI-based systems assist radiologists in identifying potential abnormalities.
- **Image Segmentation:** Isolates breast tissues and lesions for detailed analysis.
- **Noise Reduction:** Filters remove grainy textures while preserving crucial details.

The provided text also details the applications of digital image processing in other medical imaging

## **SKILLS LEARNT :**

### **1. Optical Character Recognition (OCR)**

Learned to use Tesseract OCR and OpenCV to extract text from scanned documents and images. Gained knowledge of image preprocessing techniques such as noise removal, grayscale conversion, and thresholding for improved accuracy.

### **2. Natural Language Processing (NLP)**

Understood the fundamentals of text preprocessing, tokenization, stemming, and stop-word removal. Applied techniques like keyword extraction, entity recognition, and summarization using transformer-based models.

### **3. Text Vectorization & Similarity Matching**

Learned how to convert text into numerical vectors using TF-IDF and Word2Vec. Implemented cosine similarity to compare text segments and rank them based on semantic closeness to user input.

### **4. Prompt Engineering**

Understood how to craft effective prompts for large language models like Gemini to generate relevant and concise responses. Gained hands-on practice with prompt optimization for various summarization scenarios.

### **5. API Development with FastAPI**

Built RESTful APIs for the backend of the application using FastAPI. Learned how to handle POST requests, manage input/output formats (JSON), and integrate ML models into API endpoints.

## **6. Interactive UI with Streamlit**

Developed a user-friendly frontend using Streamlit. Implemented file upload components, chatbot-style UI, and dynamic display of extracted summaries and policy insights.

## **7. Project Architecture Design**

Learned how to structure a modular AI project with clear separation between frontend, backend, and processing logic. Designed data flow and component interactions in real-world use cases.

## **8. Real-time Inference and Integration**

Gained practical exposure in integrating OCR output with NLP modules to generate real-time responses. Learned to handle response delays, exceptions, and user inputs dynamically.

## **9. Debugging and Problem Solving**

Faced and resolved challenges related to inaccurate OCR output, irrelevant summarization, and API errors. Improved debugging techniques and gained confidence in solving integration issues.

## **10. Documentation & Deployment Readiness**

Developed the habit of writing clean, modular code with proper documentation. Understood the basics of deploying AI projects in a scalable and testable format.

## **EXPERIENCE GAINED**

During the course of this project, I gained hands-on experience in developing real-world AI applications. I learned how to extract meaningful information from unstructured data using OCR techniques and transform it into structured text for further processing.

Working with NLP enhanced my understanding of text preprocessing, summarization models, and similarity-based ranking. I also developed strong backend programming skills using FastAPI, enabling smooth communication between the frontend and AI modules. On the frontend side, I worked with Streamlit to create an interactive and user-friendly interface. I became familiar with integrating multiple technologies to build a complete end-to-end solution.

This project also helped me improve my problem-solving skills as I had to debug OCR errors, optimize summarization outputs, and ensure the application performed accurately across different document types. Overall, it was a valuable experience that strengthened both my technical and analytical capabilities.

## **CONCLUSION**

This project has provided a valuable opportunity to apply and integrate various AI and data processing techniques in a practical setting. The core objective—automated document analysis through OCR, text summarization, and similarity ranking—was achieved by building an end-to-end system that reads unstructured documents, extracts key information, and presents meaningful results to the user. Through this process, I gained hands-on experience with tools like Tesseract OCR for text extraction, NLP libraries for summarization, and cosine similarity methods for document comparison. The system architecture, developed using FastAPI and optionally integrated with a simple UI (e.g., Streamlit), allowed for real-time processing and efficient output generation. Challenges

such as handling noisy data, optimizing text extraction accuracy, and ensuring relevant summaries taught me critical debugging, testing, and evaluation skills. Moreover, working with real-world data helped me understand how to convert theoretical models into scalable, useful applications. This experience has enhanced my technical confidence, problem-solving abilities, and project management skills. Overall, the project reinforced the importance of AI in automating routine but essential tasks across domains like insurance, recruitment, and document management. It has laid a strong foundation for future work in intelligent document processing systems and similar AI-powered solutions.

# INTERNSHIP CERTIFICATE :



Tel: 020 8757 5700

Email id: admin@jagsconsultancy.com

## INTERNSHIP COMPLETION LETTER

### TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Raahul Eshwar** has successfully completed an internship with our organization. The internship commenced on **3<sup>rd</sup> June 2025** and concluded on **4<sup>th</sup> July 2025**, covering a total duration of **120 hours**.

During the period of the internship, Raahul demonstrated commendable diligence, enthusiasm, and dedication towards assigned tasks. Raahul actively participated in team activities, consistently contributed valuable insights, and completed assigned responsibilities efficiently.

We wish Raahul all the best for future endeavors.

### AUTHORISATION

A handwritten signature in blue ink, appearing to read "Abinandhanan Sripal Udhaykumar".

Signed on behalf of the company – Jags Consultancy Ltd

Name : Abinandhanan Sripal Udhaykumar  
Position : Managing Partner 2025 Dated : 1<sup>st</sup> July 2025

Registered Office Address: Embassy Golf Links Road, Embassy Golf Links Business Park,  
Challaghatta, Bengaluru, Karnataka 560071.

## PERMISSION LETTER :



Email: admin@jagsconsultancy.com

Date: May 30, 2025.

Dear Raahul Eshwar,

We are pleased to inform you that you have been selected for a Summer Internship in **Jags Consultancy Ltd.** for 1 month. You will be appointed **Analyst Intern** within the company. The internship commences on **3rd June, 2025**. You will be reporting to the Head of Technology, **Abinandhanan S U**, throughout the duration of your internship. Your responsibilities will include assisting in data analysis, supporting research projects, and contributing to ongoing compliance management tasks.

You will be joining the company on **3<sup>rd</sup> June 2025**, and your internship will end by **3<sup>rd</sup> July 2025**. If you have any questions about the terms of the Internship outlined above, please feel free to contact me at [admin@jagsconsultancy.com](mailto:admin@jagsconsultancy.com)

Warm Regards,

A handwritten signature in blue ink, appearing to read "Abinandhanan S U".

**Abinandhanan S U**  
Head of Technology (Managing Partner)  
Jags Consultancy

---

Registered Office Address: Level 7, Fairway Business park, Survey Nos 10/1 11/2 and 12/2B of Challaghatta Village, Next to 'Embassy Golf Links' in Domlur, Bangalore Karnataka, 560071, India.

**Digitally Signed Copy – Does not need a Seal**

# ABOUT THE INDUSTRY/COMPANY/PROFILE/ADDRESS/CONTACT

 Jags Consultancy Ltd

Home   [About Us](#)   What We Offer ▾   Case Studies ▾   Careers  
Contact Us

[Book an Appointment](#)

## We are Jags Consultancy

Jags Consultancy Limited is an IT consulting and software development company founded in 2015, headquartered out of UK. We offer technical experts to provide insights and capability to deliver IT software solutions for enterprise and middle-market customers across multiple target industries. We carefully study each customer's requirements to understand the client's needs and objectives and deliver a reliable and efficient solution. Our tailor-made flexible approach enables us to work on client's strict timeframes, cost, and quality.



 Jags Consultancy Ltd

Home   [About Us](#)   What We Offer ▾   Case Studies ▾   Careers  
Contact Us

[Book an Appointment](#)



### Foundation of Excellence

Established to deliver bespoke software solutions. Early projects focused on IT and Energy industries for medium enterprises. Set up our Offshore-Development Centre in India.



### Scaling Expertise

Expanded services to cloud computing and data analytics. Developed partnerships with AWS and Microsoft Azure. Secured our first major enterprise client, marking entry into large-scale projects.



### Embracing Innovation

Launched AI and machine learning services to automate operations. Opened our USA branch and introduced our first SaaS product. Gained industry recognition for innovation and impact.

 Jags Consultancy Ltd

Home   [About Us](#)   What We Offer ▾   Case Studies ▾   Careers  
Contact Us

[Book an Appointment](#)

**Company**

Registered at  
450 Bath Road, Longford,  
Heathrow, London, UB7  
0EB

sales@jagsconsultancy.com  
+44 20 8757 5700

**Services**

LLM Based Solutions  
IT Services  
Cloud Computing  
Professional Services

**Our Brands**

**InsurBe**  
 opulentbridge

## **SCHEDULE COPY :**

<b>DATE</b>	<b>DAY</b>	<b>WORK COMPLETED</b>
03/06/25	1	Orientation about the Company
04/06/25	2	Introduction to the Software Development Team
05/06/25	3	Project Briefing & Goal Setting
06/06/25	4	Tools and Technology Setup
07/06/25	5	Literature Review – NLP, Word2Vec, and Gemini AI
09/06/25	6	Requirement Analysis & Functional Specifications
10/06/25	7	Design of Text Summarization Architecture
11/06/25	8	Module 1: Text Extraction from PDF/DOCX/Image
12/06/25	9	Module 1: OCR Integration and Testing
13/06/25	10	Module 2: Word2Vec Vector Embedding Implementation
14/06/25	11	Module 2: Vector Similarity Search & Chunking
16/06/25	12	Mid-Module Review and Feedback
17/06/25	13	Module 3: Gemini API Integration for Summarization
18/06/25	14	Module 3: Summary Style Customization
19/06/25	15	Module 4: Front-End UI with Streamlit
20/06/25	16	Front-End and API Integration
21/06/25	17	Testing & Debugging – Round 1
23/06/25	18	Deployment Preparation and User Testing
24/06/25	19	Resume Ranking Module Setup
25/06/25	20	Insurance Policy Analyzer Module Setup
26/06/25	21	Final Sprint Review
27/06/25	22	Integration of All Modules
28/06/25	23	Project Testing with Real-Time Inputs
30/06/25	24	Project Final Demo Preparation
01/07/25	25	Final Project Presentation
02/07/25	26	Project Documentation
03/07/25	27	Project Report Review and Edits
04/07/25	28	Knowledge Transfer & Exit Interview

## SAMPLE PHOTO :



