A

PROJECT ON

“Universal Ai powered document search library ”

Submitted In Partial Fulfilment Of The Requirements For The

Award Of The Degree Of B.Sc.

“Bachelor Of Science In computer science (Honors)”

Submitted By:

Tanuj kashyap

Roll No. – 22070170

Enrollment No. – GGV/22/05170

UNDER THE GUIDANCE OF:

Mr. Vivek Kumar Sarathe

(Department of computer science and information technology)

Guru Ghasidas Central University, Bilaspur (C.G.)

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

गुरु घासीदास विश्वविद्यालय, बिलासपुर
Guru Ghasidas Vishwavidyalaya, Bilaspur
A Central University established by the Central Universities Act 2009 No. 25 of 2009

GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR (C.G.)

(A Central University Established by Central University Act, 2009 No 25 of 2009)

(Session 2023-24)

CERTIFICATE OF APPROVAL

This is to certify that **Tanuj kashyap** of B.Sc. 6th semester at Department of computer science and information technology of Guru Ghasidas Vishwavidyalaya, Bilaspur (C.G.) bearing Roll no. 22070170, Enrollment no. GGV/22/05170 has developed project work entitled

" Universal Ai powered document search library "

is hereby approved as a credible work for the award of the degree of

Bachelor of Science in Computer science(Hons.)

Prof. Ratnesh Shrivastava

Head of the Department

Department of Computer Science and information technology

GGV, Bilaspur (C.G.)

CERTIFICATE FROM THE EXAMINER

This is to certified that the project work entitled **" Universal Ai powered document search library "** submitted by Tanuj Kashyap has completed under the guidance of Mr. Vivek Kumar Sarathe, Department of Computer Science and information technology , GGV Bilaspur(C.G.) has been examined by the undersigned as a part of the examination for the award of the **B.Sc. Computer Science Hons.** (Bachelor of Science in Computer Science honours) Degree in Department of "Computer Science and information technology" in **Guru Ghasidas Central University, Koni, Bilaspur (C.G.).**

Date :

_____

Examiner

Tanuj Kashyap

Computer Science and

information technology

गुरु घासीदास विश्वविद्यालय, बिलासपुर
Guru Ghasidas Vishwavidyalaya, Bilaspur
A Central University established by the Central Universities Act 2009 No. 25 of 2009

CERTIFICATE OF THE GUIDE

This is to certify that Tanuj Kashyap a student of Bachelor of Science in Computer Science Honours (6th Semester, Enrollment No. GGV/21/07604), **Department of Computer Science and information technology** , **Guru Ghasidas Central University Bilaspur (C.G.)** has successfully completed his project entitled **"UNIVERSAL AI POWERED DOCUMENT SEARCH LIBRARY "** under my guidance and supervision for the award of the Degree of **Bachelor of Science in Computer Science (Hons.)**.

This project is original, as it has not previously formed the basis for the award of any other degree.

_____

(Signature of the Guide)

Mr. Vivek Kumar Sarathe

Assistant Professor

(Dept. Computer Science and information technology)

DECLARATION

I hereby, declare that the work presented in this project report entitled **" UNIVERSAL AI POWERED DOCUMENT SEARCH LIBRARY "** in partial fulfilment for the 6 th semester of Degree of B.Sc. Computer Science (Hons.) is a record of my own investigations carried out under the guidance of my institute's faculty member **Mr. Vivek Kumar Sarathe (Professor, Department of Computer Science and information technology)**. It is not copied from any external sources. If anything related to my project founds incorrect then the institute/University has the right to disapprove my project.

(Signature)

Tanuj Kashyap

Date :

B.Sc. 6 th Semester Computer Science (Hons.)

Department of Computer Science and information technology

GGV, Bilaspur (C.G.)

ACKNOWLEDGEMNT

It is a great privilege for me to represent this project report on **" UNIVERSAL AI POWERED DOCUMENT SEARCH LIBRARY ”**

I would like to thank my project guide **Mr. Vivek Kumar Sarathe (Professor, Department of Computer Science and information technology)** for his extended support in boosting me during the study period.

I am also heartily grateful to the faculties of the department as it was their exemplary encouragement and enthusiasm that they exhibit during their lectures for promoting the idea of an innovative and entertaining education, which inspired me to imagine the importance and need of my project's idea.

I am indebted my mother, my father and friends for all the encouragement and moral support.

Yours Sincerely

Tanuj Kashyap

CONTENTS

ABSTRACT

The Universal AI Document Search Library is an advanced, AI-powered system designed to revolutionize the way users interact with large collections of digital documents. This library provides the ability to perform deep, semantic searches *within* any document—across various formats including PDFs, Word files, text documents, and more—rather than limiting queries to filenames or basic metadata. Users can input natural language queries and receive accurate, contextually relevant results that point directly to the most pertinent sections within documents.

At its core, the system is built using the Django web framework, ensuring a secure, scalable, and maintainable backend architecture. The intelligence behind the search functionality is powered by Google's Gemini AI model, which enables the system to understand complex language patterns, extract meaning, and deliver results that go far beyond traditional keyword-based search methods. This integration allows for intuitive, human-like interactions with the document library, making information retrieval significantly faster and more efficient.

The library is designed to be highly adaptable and can be integrated into existing digital infrastructures, whether for educational institutions, research organizations, legal firms, or enterprise content management systems. It supports multilingual queries, context-aware searching, and has the capability to process and index large volumes of data in real time. By combining cutting-edge AI with a robust backend system, the Universal AI Document Search Library offers a transformative approach to document exploration—bridging the gap between raw data and actionable insights.

This solution empowers users to find exactly what they need, when they need it, regardless of document complexity or size, enhancing productivity, knowledge discovery, and decision-making processes.

CHAPTER- 1

-- Introduction to UNIVERSAL AI POWERED DOCUMENT SEARCH LIBRARY --

In the digital age, the volume of documents generated, shared, and archived has grown exponentially across industries. From academic papers and legal contracts to scanned forms, handwritten notes, and corporate reports, organizations handle a wide range of document formats. However, retrieving relevant information from this vast pool remains a major challenge—especially when the documents differ in structure, readability, and format. Traditional file search systems are limited by basic keyword matching or metadata filtering, which fails when users need to search *within* documents or across multiple formats.

The **Universal AI Document Search Library** addresses this problem by offering an intelligent, cross-format search system that enables users to perform semantic searches across any document stored in a digital library. Whether the document is a text-based PDF, a Word file, a scanned image, or a complex non-readable format, this library makes its contents searchable using natural language input. The system is built using **Django** for a scalable backend and leverages **Google's Gemini AI model** to perform advanced language understanding and content extraction.

1.1 Motivation

Professionals often waste valuable time searching for information hidden in large volumes of documents. Manual reading or keyword-based tools are inefficient and ineffective, especially when dealing with scanned documents or when users don't know the exact wording of the content they're searching for. The motivation behind this project is to eliminate these barriers by creating a universal solution that can handle:

- **Machine-readable formats:** TXT, DOCX, HTML, and standard PDFs.
- **Scanned documents:** Including image-based PDFs, JPG, PNG files, processed via OCR (Optical Character Recognition).
- **Non-readable formats:** Handwritten notes, poorly scanned documents, or proprietary layouts, which are converted to readable text before indexing.
- **Multilingual or structured content:** Enabling semantic search even in documents with complex formatting or multiple languages.

1.2 Objectives

- To create a unified search interface that can understand and respond to natural language queries.
- To support full-text search across diverse document formats—readable, scanned, or otherwise.
- To integrate AI-driven semantic analysis for context-aware results, beyond simple keyword matching.
- To build a backend architecture using Django that can scale and integrate with various frontends or existing systems.
- To enhance document accessibility, saving time and boosting productivity for users across domains.

1.3 Scope

- The system is designed to serve a wide range of users and applications—from educational and legal institutions to corporate environments and government agencies. It focuses on making large document collections easily searchable regardless of technical limitations such as format type or file readability. The core functionality includes:
- Uploading and indexing documents.
- Running OCR on non-text-based inputs.
- Storing and querying embeddings using the Gemini model.
- Returning relevant text snippets based on user queries in real time.
- By combining machine learning, natural language understanding, and robust backend engineering, this project sets the foundation for a smarter, more accessible approach to document search.

CHAPTER-2

-- Understanding Documents –

To ensure comprehensive search functionality, the Universal AI Document Search Library is designed to handle a wide variety of document types. Each document type presents unique challenges in terms of accessibility, parsing, and text extraction. Understanding these document types is crucial to building a robust, flexible, and intelligent search system.

Below are the major categories of documents supported by the system, along with an explanation of each:

2.1a Machine-Readable Text Documents

These documents contain text that can be directly parsed by a computer without the need for additional processing. They are structured and standardized, making them the easiest to work with.

Examples include:

- .txt (plain text)
- .doc / .docx (Microsoft Word)
- .pdf (text-based PDFs)
- .html / .xml

Processing Approach:
These documents are directly read and tokenized. Content is extracted using file parsers like python-docx for Word files and PyMuPDF or pdfplumber for PDFs.

2.1b Scanned Documents (Image-Based PDFs)

These files contain images of text, such as scanned books, forms, or handwritten notes saved as PDF files. The text is not directly accessible until processed with OCR.

Examples include:

- Scanned PDFs
- Digitized physical forms or letters
- Printed contracts scanned into image format

Processing Approach:
OCR (Optical Character Recognition) tools like **Tesseract** or **EasyOCR** are used to convert image content into searchable text. Post-processing is applied to clean up and correct OCR results.

2.1c Image Files with Embedded Text

Images such as photographs, screenshots, or scanned pages containing textual data fall under this category. Like scanned PDFs, they require OCR to extract text.

Examples include:

- .jpg, .jpeg, .png, .bmp images of documents
- Screenshots of documents or chat messages

Processing Approach:
OCR is applied directly to the image. Preprocessing techniques like grayscale conversion, binarization, and noise removal may be used to improve accuracy.

2.1d Non-Machine-Readable PDFs

These are PDF files that may appear to contain text but are actually built from embedded images or vector content that is not directly parsable.

Examples include:

- PDFs exported from graphic design tools
- PDF scans that look like text but are not selectable

Processing Approach:
Each page is treated as an image and passed through the OCR pipeline before indexing.

2.1e Handwritten Documents

These include notes or forms written by hand and then scanned or photographed. Handwritten text recognition is significantly more complex than printed text recognition.

Examples include:

- Scanned handwritten notebooks
- Digital handwritten notes from tablets
- Old archives or historical documents

Processing Approach:
Advanced OCR models trained on handwriting (e.g., Google Vision AI, Microsoft Azure OCR) are used. Handwriting recognition models tend to be slower and less accurate than those for printed text.

2.1f Semi-Structured and Structured Documents

These documents contain a defined format with a mix of text, tables, headers, and sections. The challenge is preserving layout and context during extraction.

Examples include:

- Research papers with sections and references
- Financial reports with tables
- Legal contracts with clauses and definitions

Processing Approach:
Parsing tools extract hierarchical data and preserve context using NLP-based structure detection. Table parsers may be employed for accurate extraction from PDFs.

-- Feasibility—

Before building any intelligent document search system, it is essential to assess the technical and practical feasibility of the proposed solution. While the ultimate vision of the Universal AI Document Search Library is to support a wide range of document types, the current prototype is limited to PDF and Word (DOCX) documents. This feasibility study evaluates the current capabilities and justifies the selected scope.

2.2a Supported Document Types (Feasible)

PDF Documents (Text-Based or Mixed Content) PDFs are among the most common and widely used digital document formats. They are well-supported in most environments and can contain structured or unstructured data, including text, tables, and embedded images.

- **Feasibility:** Fully supported
- **Processing Tools:** PyMuPDF, pdfplumber for text extraction
- **Challenges:** OCR for image-based PDFs is excluded in this prototype
- **Status:** Feasible and implemented

Microsoft Word Documents (.docx) Word documents are another popular and widely used format, especially in professional and academic environments.

- **Feasibility:** Fully supported
- **Processing Tools:** python-docx for parsing and text extraction
- **Challenges:** Minor layout inconsistencies, but text is easily retrievable
- **Status:** Feasible and implemented

3.2 Unsupported Document Types (Out of Scope for Prototype)

While future versions of the system aim to support various additional formats, these are **not feasible in the current prototype**:

❌ **Scanned PDFs / Image-Based PDFs**

These require OCR integration, which is not part of the current implementation.

❌ **Image Files (JPEG, PNG, etc.)**

Support for OCR on image files is excluded for this prototype to reduce complexity.

❌ **Handwritten Notes / Non-Machine-Readable Formats**

Handwriting recognition demands advanced OCR models and training data, which is out of scope.

❌ **Multilingual or Structured Table-Heavy Documents**

While partially possible, handling multilingual content or complex layouts like tables is not supported in this version.

CHAPTER – 3

--Backend : Python Django framework --

The backend of the Universal AI-Powered Document Search Library is developed using Django, a high-level Python web framework that enables rapid development, scalability, and clean design. Django offers an MVT (Model-View-Template) architectural pattern, making it an ideal choice for building a robust backend for AI-based applications.

3.1.1 Architecture Overview

The backend is responsible for:

- **Handling user authentication and permissions.**
- **Managing document uploads and metadata.**
- **Indexing documents for AI-powered search.**
- **Integrating AI models for semantic and keyword search.**
- **Serving API endpoints for the frontend and third-party clients.**

The application is structured into multiple Django apps for modularity:

- **users: Handles registration, login, and user roles.**
- **documents: Manages file uploads, storage, and metadata.**
- **search: Implements AI-based indexing and querying.**
- **api: Provides RESTful API endpoints using Django REST Framework (DRF).**

3.1.2 Key Technologies and Libraries

- **Django: Core web framework.**
- **Django REST Framework: For building APIs.**
- **Celery with Redis: For asynchronous processing (e.g., indexing).**
- **PyMuPDF / textract: For extracting text from PDFs and other formats.**
- **FAISS / Weaviate / Elasticsearch: For storing vector embeddings.**
- **Transformers / Sentence-BERT: For generating semantic embeddings.**
- **PostgreSQL: Primary relational database.**

3.1.3 Document Upload and Processing Flow

1. **User Upload: Users upload documents via the frontend or API.**
2. **File Storage: Files are stored in the file system or cloud storage.**
3. **Text Extraction: Upon upload, a background Celery task extracts text from documents.**
4. **Embedding Generation: Text is converted to vector embeddings using a pretrained Sentence-BERT model.**
5. **Vector Indexing: Embeddings are stored in a vector database for fast retrieval.**

3.1.4 AI Search Implementation

- **Semantic Search: When a user submits a query, it is converted to an embedding using the same AI model. This embedding is compared with indexed document vectors to find the most relevant results.**
- **Keyword Matching (Optional): A fallback TF-IDF or BM25-based search is implemented for exact keyword matches.**
- **Result Ranking: Results are ranked based on vector similarity and relevance scoring.**

3.1.5 API Design (Sample Endpoints)

Endpoint	Method	Description
/api/upload/	POST	Upload a new document
/api/search/	POST	Submit a search query
/api/documents/	GET	List all documents
/api/documents/<id>/	GET	Retrieve a specific document
/api/users/register/	POST	Register a new user
/api/users/login/	POST	Login and obtain authentication

3.1.6 Scalability and Optimization

To ensure the system can handle large-scale document sets and concurrent search queries:

- **Vector indexing is offloaded to specialized databases like FAISS or Elasticsearch.**
- **Asynchronous tasks are used for embedding and indexing.**
- **Caching is implemented using Redis to store recent search results.**
- **Load balancing and containerization (using Docker) are used for deployment.**

-- React.Js frontend –

3.2.1 Introduction

The frontend of the Universal AI-Powered Document Search Library is developed using **React.js**, a popular JavaScript library for building user interfaces. React provides a modular and component-based approach, allowing for a responsive and interactive user experience. The frontend communicates with the Django backend via RESTful APIs, handling document uploads, search queries, and displaying results powered by AI.

3.2.2 Technology Stack

- **React.js** – Core UI library
- **Axios** – For HTTP requests
- **React Router** – For client-side routing
- **Tailwind CSS** (or Bootstrap) – For styling and responsive design
- **Context API / Redux** – For state management (optional, based on complexity)
- **Framer Motion** – For smooth animations (optional)

3.2.3 project structure-

```
/frontend
|
 ├── /public
 ├── /src
|   ├── /components      # Reusable components (Navbar,
                                    SearchBar,DocumentCard)
|   ├── /pages          # Main pages (Home, Upload, Results)
|   ├── /services       # API handling logic (axios requests)
|   ├── App.js          # Root component
|   ├── index.js        # Entry point
|   └── styles.css      # Global styles
```

a) SearchBar.js

This component allows users to enter natural language queries.

b) UploadPage.js

Allows users to upload documents (PDFs, DOCXs, etc.) to be indexed by the backend AI service.

c) ResultsPage.js

Displays AI-enhanced search results with context and snippets from documents.

3.2.5 API Integration

Using axios, the frontend communicates with the backend for document uploads and semantic search.

Example: API Call for Search

```
import axios from 'axios';

export const searchDocuments = async (query) => {
  const response = await axios.post('/api/search/', { query });
  return response.data.results;
};
```

3.2.6 User Experience and Design

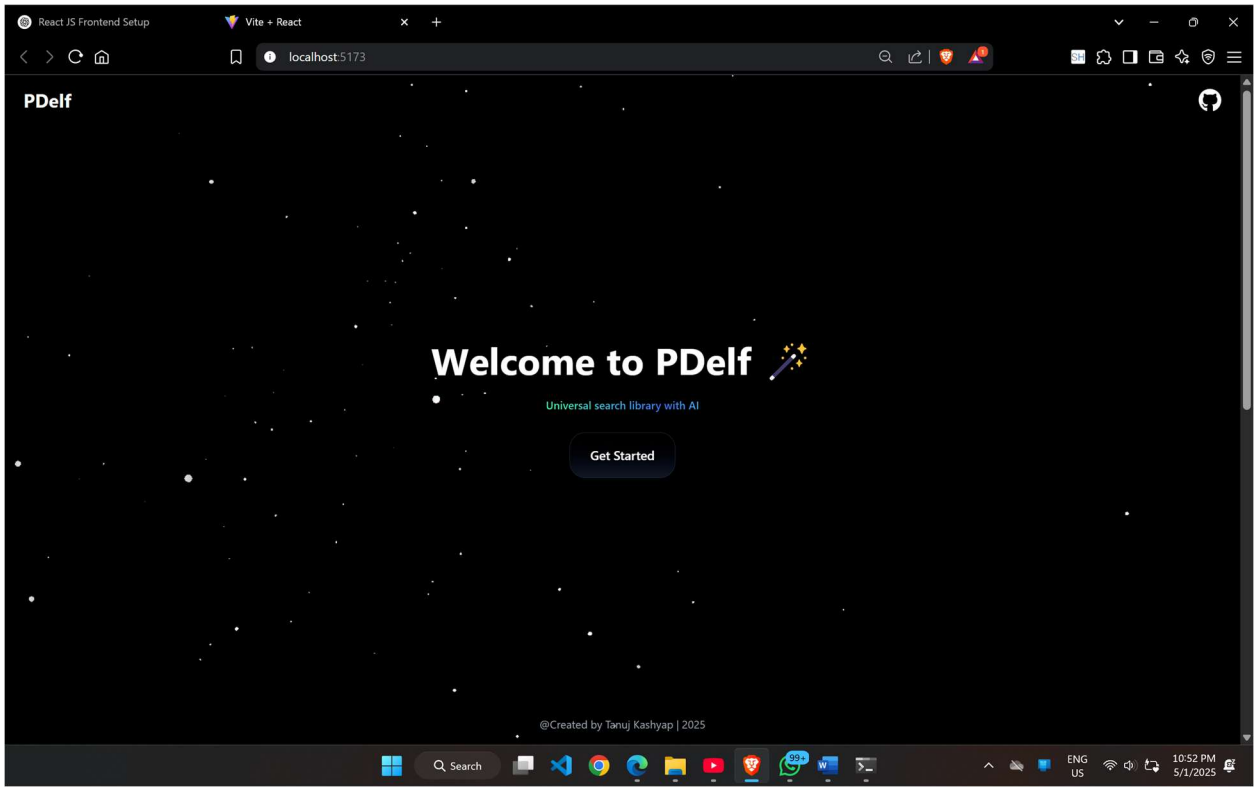
The frontend ensures:

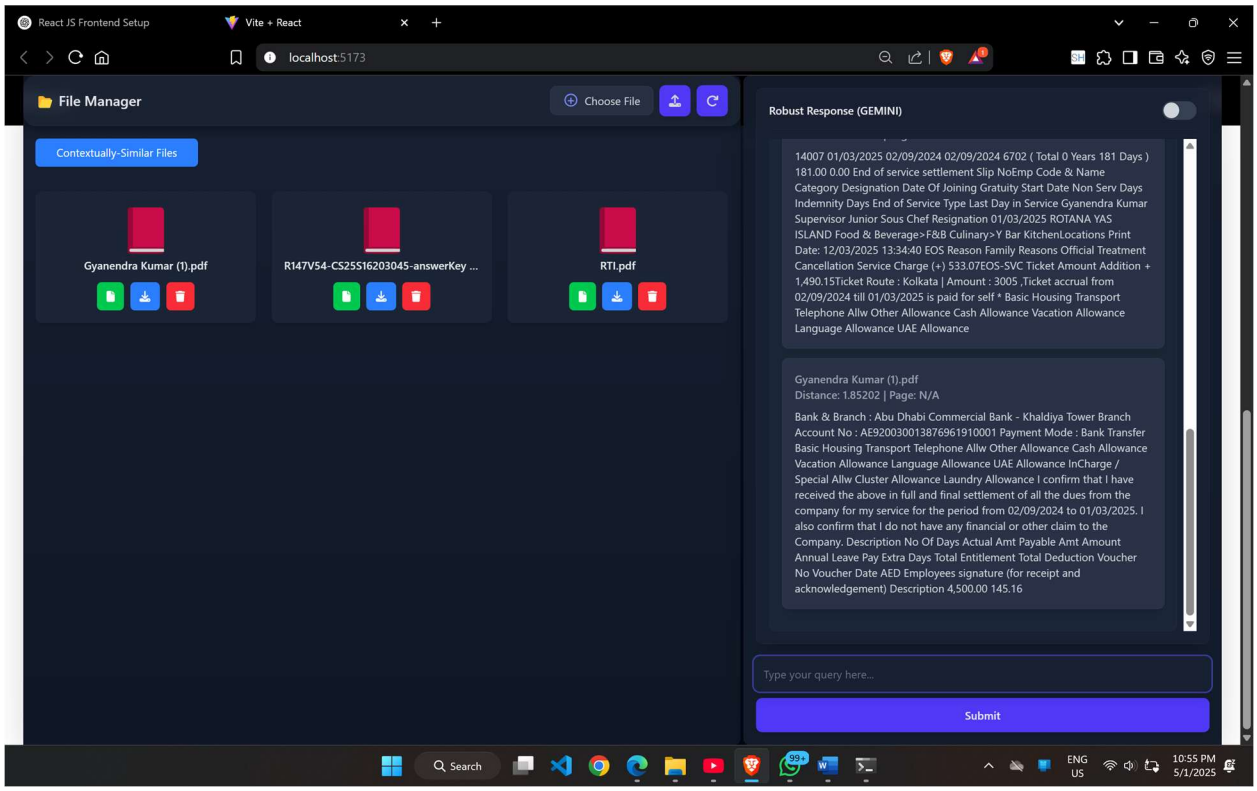
- **Minimalist Design**: Focus on usability and clarity.
- **Responsiveness**: Works across devices using responsive CSS frameworks.
- **Performance**: Lightweight React components with lazy loading if needed.
- **Accessibility**: Proper labeling and keyboard navigation support.

3.2.7 Security Considerations

- File inputs are sanitized before uploading.
- API errors are gracefully handled.
- CSRF tokens are managed when communicating with Django if session-based auth is used.

-- Outcome –


React JS Frontend Setup
Vite + React
localhost:5173
PDelf
Welcome to PDelf
Universal search library with AI
Get Started
@Created by Tanuj Kashyap | 2025
Search
ENG
US
10:52 PM
5/1/2025


React JS Frontend Setup
Vite + React
localhost:5173
File Manager
Choose File
Contextually-Similar Files
Gyanendra Kumar (1).pdf
R147V54-CS25S16203045-answerKey ...
RTI.pdf
Robust Response (GEMINI)
14007 01/03/2025 02/09/2024 02/09/2024 6702 (Total 0 Years 181 Days) 181.00 0.00 End of service settlement Slip NoEmp Code & Name Category Designation Date Of Joining Gratuity Start Date Non Serv Days Indemnity Days End of Service Type Last Day in Service Gyanendra Kumar Supervisor Junior Sous Chef Resignation 01/03/2025 ROTANA YAS ISLAND Food & Beverage>F&B Culinary>Y Bar KitchenLocations Print Date: 12/03/2025 13:34:40 EOS Reason Family Reasons Official Treatment Cancellation Service Charge (+) 533.07EOS-SVC Ticket Amount Addition + 1,490.15Ticket Route : Kolkata | Amount : 3005 ,Ticket accrual from 02/09/2024 till 01/03/2025 is paid for self * Basic Housing Transport Telephone Allw Other Allowance Cash Allowance Vacation Allowance Language Allowance UAE Allowance
Gyanendra Kumar (1).pdf
Distance: 1.85202 | Page: N/A
Bank & Branch : Abu Dhabi Commercial Bank - Khaldiya Tower Branch Account No : AE920300138769619100 01 Payment Mode : Bank Transfer Basic Housing Transport Telephone Allw Other Allowance Cash Allowance Vacation Allowance Language Allowance UAE Allowance InCharge / Special Allw Cluster Allowance Laundry Allowance I confirm that I have received the above in full and final settlement of all the dues from the company for my service for the period from 02/09/2024 to 01/03/2025. I also confirm that I do not have any financial or other claim to the Company. Description No Of Days Actual Amt Payable Amt Amount Annual Leave Pay Extra Days Total Entitlement Total Deduction Voucher No Voucher Date AED Employees signature (for receipt and acknowledgement) Description 4,500.00 145.16
Type your query here...
Submit
Search
ENG
US
10:55 PM
5/1/2025

-- REFERENCES –

[1] https://youtu.be/FxgM9k1rg0Q?si=HmVZBvZloP4j0mz3

[2] https://w3schools.org/abs/2309.10813

[3] https://geeksforgeeks.com/abs/2209.07184

[4] https://chatgpt.com/ 2305.13898

[5] https://www.semanticscholar.org/paper/d8fb2a4b684dac23a05e3c8ab56f0b437f47819c