**INTERNSHIP REPORT**

**On**

**BUSINESS CONTRACT VALIDATION**

**By**

**BU21CSEN0100908 P. Sayed Sab Vali**

A logo with text on it

Description automatically generated

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Gandhi Institute of Technology and Management**

**(DEEMED TO BE A UNIVERSITY)**

**BENGALURU, KARNATAKA, INDIA**

**SESSION:2021-2025**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM SCHOOL OF TECHNOLOGY**

**GITAM**

A logo with text on it

Description automatically generated

**DECLARATION**

We, hereby declare that the Internship report entitled **“Business Contract Validation-To Classify Content within the Contract Clauses and Determine Deviations from Templates and highlight them”** is an original work done in the **Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) Bengaluru** submitted in partial fulfilment of the requirements for the award of the degree of **B.Tech.** in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree.

Date:13-08-2024

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM SCHOOL OF TECHNOLOGY**

**GITAM**

A logo with text on it

Description automatically generated

**CERTIFICATE**

This is to certify that the mini project report entitled “Business Contract Validation” is a bonafide record of work carried out by Patan Sayed Sab Vali submitted in partial fulfillment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering.

Dr. Bhavya KR

Asst.Professor,

Department of CSE,

GST, Bengaluru

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose consistent guidance and encouragement crowned our efforts with success.

We consider it our privilege to express our gratitude to all those who guided us in the completion of the project.

We express our gratitude to Director Prof. **Basavaraj Gundappa Katageri** for having provided uswith the golden opportunity to undertake this project work in their esteemed organization.

We sincerely thank **Dr. Y. Vamshidhar,** HOD, Department of Computer Science and Engineering, Gandhi Institute of Technology and Management, Bengaluru for the immense support given to us.

We express our gratitude to our project guide **Dr. Bhavya KR, Asst.Professor**, Department of Computer Science and Engineering, Gandhi Institute of Technology and Management, Bengaluru, for their support, guidance, and suggestions throughout the project work.

**TABLE OF CONTENTS**

**Title Page No**

Internship 1

Declaration 2

Certificate 3

Acknowledgement 4

Table of Contents 5

Abstract 6-7

Introduction

* 1. Background of a research problem 7-8
  2. Challenges 8-9
  3. Scope 9-10
  4. Applications 10-11
  5. Stages Of Validating A Business Contract 11-13
  6. Project objectives 13-14

2. Features Offered 14-15

3. Process Flow 15-16

4. Technologies Used 16-17

5.Details about Project 17-21

6.Conclusion 21-22

**Abstract :**

The application revolutionizes the contract validation process by employing cutting-edge natural language processing (NLP) and text comparison algorithms. It offers an all-encompassing solution for analyzing contract clauses, identifying deviations from established templates, and highlighting these differences for further review. By utilizing NLP, the application automatically detects and categorizes each clause within a contract into relevant categories, such as payment terms, obligations, warranties, and dispute resolution, ensuring precise labelling and comprehension of each contract component.

After classification, the application conducts a comparison with a predefined standard template. This comparison is powered by advanced text comparison algorithms that evaluate whether the clauses conform to the specified standards and expectations set out in the template. The system identifies and highlights any deviations, providing a thorough analysis of these discrepancies. This includes recognizing clauses that differ in language, scope, or intent from the template, which might pose risks or compliance issues.

The application generates detailed reports that outline the deviations and their potential impacts. It also provides recommendations for adjusting the contract to align with the template, helping users make informed decisions and modifications to ensure consistency and adherence to best practices. By automating the review process, the application greatly reduces the time and effort required for manual inspections. It improves accuracy and consistency in contract management, minimizes oversight risks, and ensures compliance with regulatory requirements. This innovative approach streamlines operations, enhances risk management, and contributes to superior contract governance. In essence, this application represents a major advancement in contract validation technology, offering a robust tool for organizations aiming to optimize their contract management through automation and advanced analytics.

**CHAPTER 1**

**INTRODUCTION**

* 1. **Background of a research problem**

When validating a business contract by classifying content within clauses and identifying deviations from templates, the process involves meticulous scrutiny. Each clause is categorized according to its purpose, such as defining obligations, specifying terms, or outlining responsibilities. By comparing these clauses against established templates or standard industry practices, any discrepancies or variations can be pinpointed and highlighted. This method ensures that the contract is not only comprehensive but also aligns closely with best practices, enhancing its clarity, enforceability, and legal compliance. Through this systematic approach, potential risks and ambiguities are minimized, facilitating smoother negotiations and clearer expectations between parties involved.

Validating a business contract through detailed clause analysis and deviation detection is an essential step in contract management, ensuring that the agreement is well-structured and adheres to best practices. This process starts with the careful examination of each clause, categorizing them based on their function—such as defining obligations, specifying terms, or outlining responsibilities. This classification helps in understanding the purpose and importance of each clause within the overall contract.

Following categorization, each clause is compared to established templates and standard industry practices. These templates are based on legal precedents, industry norms, or organizational standards that aim to protect interests and ensure compliance. The goal of this comparison is to identify any deviations or differences from these benchmarks. Deviations may include variations in language, terms, or conditions that could affect the clarity or enforceability of the contract.

**1.2 Challenges in Business Contract Validation**

1. **Complexity of Clause Classification:** Classifying contract clauses accurately—whether they address obligations, terms, or responsibilities—can be complex due to the intricate and varied nature of contract language. Distinguishing between similar clauses and understanding their roles in different contexts demands precise knowledge and skill.
2. **Variability in Contract Language:** Contracts frequently employ diverse terminology and phrasing, which complicates the process of aligning clauses with standard templates or industry norms. This variability can create difficulties in effectively comparing clauses and identifying deviations, especially if the contract language diverges from the template or includes unique elements.
3. **Template Alignment:** Developing a relevant and up-to-date template that reflects current industry standards and best practices presents a challenge. Templates must be regularly revised to include changes in regulations, standards, or organisational needs and must be sufficiently detailed for each specific contract.
4. **Handling Ambiguities and Discrepancies:** Contracts often contain vague or unclear language, which can complicate the identification of deviations. Interpreting the meaning and implications of such language requires thorough analysis, as discrepancies might involve subtle differences with significant legal or operational consequences.
5. **Integration of Advanced Technology:** While natural language processing (NLP) and text comparison algorithms are valuable tools, they are not perfect. Ensuring these technologies accurately interpret and analyse complex legal text is challenging. Limitations or inaccuracies in algorithms can result in overlooked deviations or false positives, affecting the validation process's reliability.
6. **User Expertise and Interpretation:** The success of the contract validation process relies on both the technology used and the expertise of the users interpreting the results.

**1.3 SCOPE**

1. **Advanced NLP Techniques**: Python offers powerful libraries like SpaCy, NLTK, and Transformers that can be used to handle the complexity of legal language. You can explore custom-trained models to improve clause classification and interpretation accuracy.
2. **Template Management and Version Control**: Utilize Python’s capabilities to build a dynamic template management system that can handle the variability of contracts across industries. This system can automate updates and track versions, ensuring that templates remain current and relevant.
3. **Scalable NLP Models**: Python’s scalable frameworks, like TensorFlow and PyTorch, can help manage large datasets as the volume of contracts grows. You can design the system to handle increasing amounts of data without losing efficiency or accuracy.
4. **Customization Options**: Python's modularity makes it easier to develop a customizable system that allows organizations to define their own templates and review criteria. This can be done through configurable settings or plugins.
5. **Data Privacy and Security**: Implement security measures in Python, such as encryption and secure data handling practices, to ensure that sensitive contract data is protected. Python libraries like PyCryptodome can be used to secure data storage and transmission.
6. **Integration with Existing Systems**: Python's extensive libraries and APIs enable easy integration with existing contract management systems, allowing seamless data exchange and collaboration

**1.4 APPLICATIONS**

1. **Legal Contract Review**: The application can be used by law firms and in-house legal teams to automate the review of business contracts, ensuring compliance with legal standards and reducing manual effort.
2. **Compliance Management**: Companies can use the tool to verify that contracts adhere to regulatory requirements and industry best practices, minimizing the risk of legal penalties or disputes.
3. **Real Estate Transactions**: Real estate firms can leverage the application to validate property contracts, ensuring that all clauses are in compliance with real estate laws and regulations.
4. **Insurance Policy Review**: Insurance companies can use the tool to analyze policy documents, ensuring that all clauses meet regulatory requirements and protecting against potential legal challenges.
5. **Financial Agreements**: Banks and financial institutions can apply the system to review loan agreements, investment contracts, and other financial documents, ensuring compliance with financial regulations and reducing the risk of contractual disputes.
6. **Employment Contracts**: Human resources departments can use the application to standardize and validate employment contracts, ensuring that they comply with labour laws and internal policies.
7. **Intellectual Property (IP) Agreements**: The application can help in reviewing IP contracts, ensuring that all clauses related to patents, trademarks, and copyrights are clear, enforceable, and legally compliant.

**1.5 STAGES OF VALIDATING A BUSINESS CONTRACT**

Business Contract Validation Involves Several Stages, Each with Specific Tasks and Techniques. Here's a Breakdown of the Stages:

**Requirement Gathering and Analysis** :

Objective :- Understand the specific needs of the end-users, including legal professionals and businesses. Identify the types of contracts, templates, and compliance requirements the application should support.

Outcome :- A detailed project plan, outlining the features, functionalities, and scope of the application.

**Data Collection and Preprocessing** :

Objective :- Collect a diverse set of contracts and templates across various industries. Clean, preprocess, and annotate the data to train and test NLP models.

Outcome :- A well-prepared dataset for model training, ensuring that the application can handle real-world contract language and variations.

**NLP Model Development** :

Objective :- Develop and train NLP models to classify clauses and detect deviations. Fine-tune the models using Python libraries like SpaCy, NLTK, or Hugging Face Transformers.

Outcome :- A robust NLP model capable of accurately interpreting and classifying legal language.

**Algorithm Development** :

Objective :- Implement text comparison algorithms to detect deviations from predefined templates. Ensure the system can highlight differences effectively.

Outcome :- An algorithm that can automatically identify and flag non-standard clauses within contracts.

**Backend Development** :

Objective :- Develop the server-side components, including database management, API development, and security features. Ensure the system can handle user requests, data processing, and secure data storage.

Outcome :- A fully functional backend that supports the core functionalities of the application.

**Frontend Development** :

Objective :- Create a user-friendly interface where users can upload contracts, view flagged deviations, and make decisions based on the analysis. Ensure that the interface is intuitive and accessible.

Outcome :- A responsive and easy-to-use frontend that meets the needs of the target audience.

**Testing and Validation** :

Objective :- Rigorously test the application for accuracy, performance, and security. Validate that the NLP models and algorithms work as intended across different contract types.

Outcome :- A thoroughly tested application, with identified bugs fixed and performance optimized.

**Clause Classification** :

Objective :- Categorize contract clauses based on their function, such as obligations, liabilities, terms, and conditions.

Tasks :- Use NLP models to classify clauses into predefined categories, ensuring all necessary topics are covered.

Outcome :- A structured breakdown of the contract, with clauses organized into relevant categories.

**Negotiation and Revision** :

Objective :- Address any identified issues and negotiate necessary revisions.

Tasks :- Collaborate with the other party to clarify ambiguous clauses, negotiate better terms, or resolve risks identified in the previous stages.

Outcome :- A revised contract that addresses concerns and aligns with both parties’ expectations.

**Entity Detection** :

Objective :- Identify and extract key entities within the contract, such as parties, dates, monetary values, and legal terms.

Tasks :- Use entity recognition techniques (e.g., Named Entity Recognition, or NER) to detect and categorize entities, ensuring accuracy in identifying critical components.

Outcome :- A list of identified entities, making it easier to analyze and verify crucial contract elements.

**1.7 Project objectives**

**Automate Contract Validation**: Develop a system that automates the validation of business contracts by classifying clauses, detecting key entities, and identifying deviations from predefined templates.

**Enhance Accuracy and Consistency**: Utilize advanced NLP techniques to ensure accurate interpretation and classification of legal language, reducing the potential for human error in contract reviews.

**Streamline Compliance**: Create tools within the system that automatically check contracts for compliance with relevant legal and regulatory standards, ensuring that contracts are legally sound and adhere to best practices.

**Reduce Manual Effort**: Minimize the time and effort required for manual contract review by automating the identification of key entities, clause classification, and risk assessment.

**Facilitate Customization**: Allow users to customize templates, review criteria, and compliance checks according to their specific industry or organizational needs.

**Improve Risk Management**: Implement a risk assessment feature that identifies potential legal, financial, and operational risks associated with non-standard clauses or entities, enabling proactive mitigation.

**Features Offered**:

**PDF File Upload** :

Allows users to upload PDF files of business contracts.

**Text Extraction** :

Extracts text from the uploaded PDF files using PyMuPDF.

**Template Comparison**:

Compares the extracted text against a predefined contract template to

Identify Deviations.

**Clause Highlighting**:

Highlights important clauses and terms within the contract text.

**Deviation Highlighting**:

Highlights deviations from the template in different colors for easy

identification (e.g., red for missing clauses, green for added clauses).

**Named Entity Recognition (NER)** :

Identifies and extracts key entities (e.g., names, dates, addresses) from

the contract text using spaCy.

**Web Interface** :

Provides a user-friendly web interface for uploading files and displaying

results.

**Process Flow:**

* **User Uploads Contract** :

The user navigates to the web application and uploads a PDF file of the business contract.

* **Text Extraction** :

The uploaded PDF is processed to extract its text content using the extract\_text\_from\_pdf function, which utilizes PyMuPDF.

* **Template Comparison** :

The extracted text is compared with a predefined template using the extract\_important\_contents function. The differences are highlighted to show deviations.

* **Highlighting Terms** :

Specific important terms and clauses within the contract text are highlighted using the highlight\_contract\_terms function.

* **Entity Extraction** :

The text is processed with spaCy to extract named entities (such as names, addresses, dates) using the extract\_entities function.

* **Results Displayed** :

The processed text, highlighted terms, deviations, and extracted entities are displayed on the results page.

* **Flask** :

A lightweight WSGI web application framework in Python used for building the web interface.

**Technologies Used:**

* **PyMuPDF (fitz)** :

A library used for reading and extracting text from PDF files.

* **spaCy** :

An advanced NLP library in Python used for named entity recognition.

* **Difflib** :

A module for comparing sequences, used to compare contract text with the template and highlight differences.

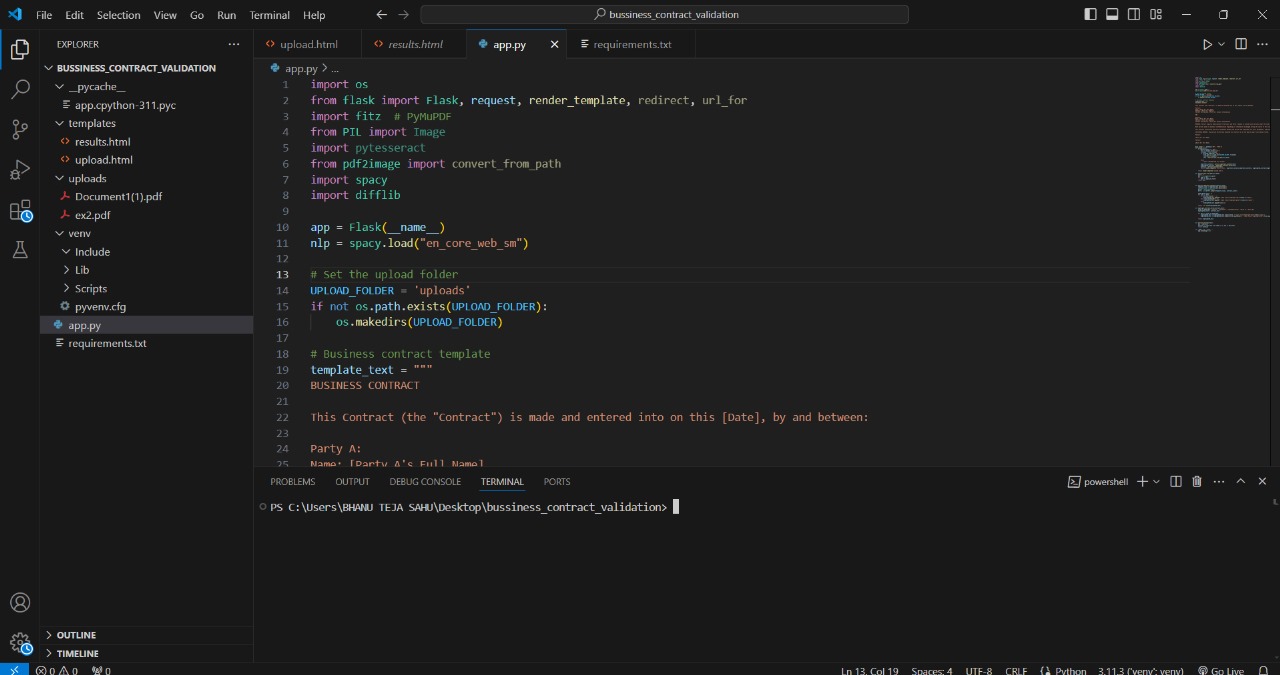
* **HTML/CSS** :

Used for creating the web interface and styling the highlighted text and results.

The combination of these technologies allows for a robust and efficient system to automate the contract validation process, enhancing accuracy and saving time.

**Details about the project :**

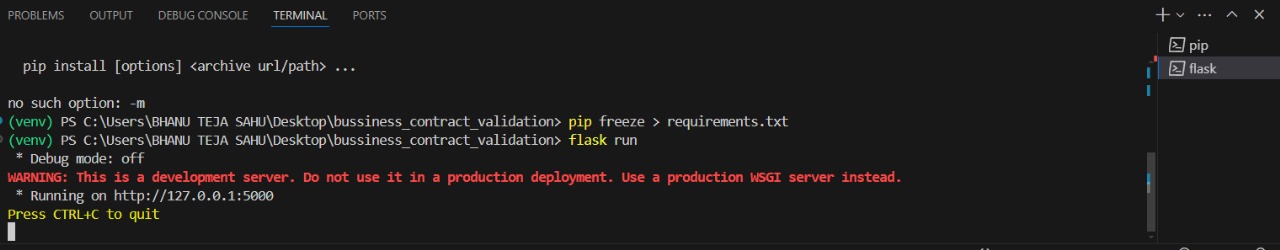
1.Open the source code to validate the contract



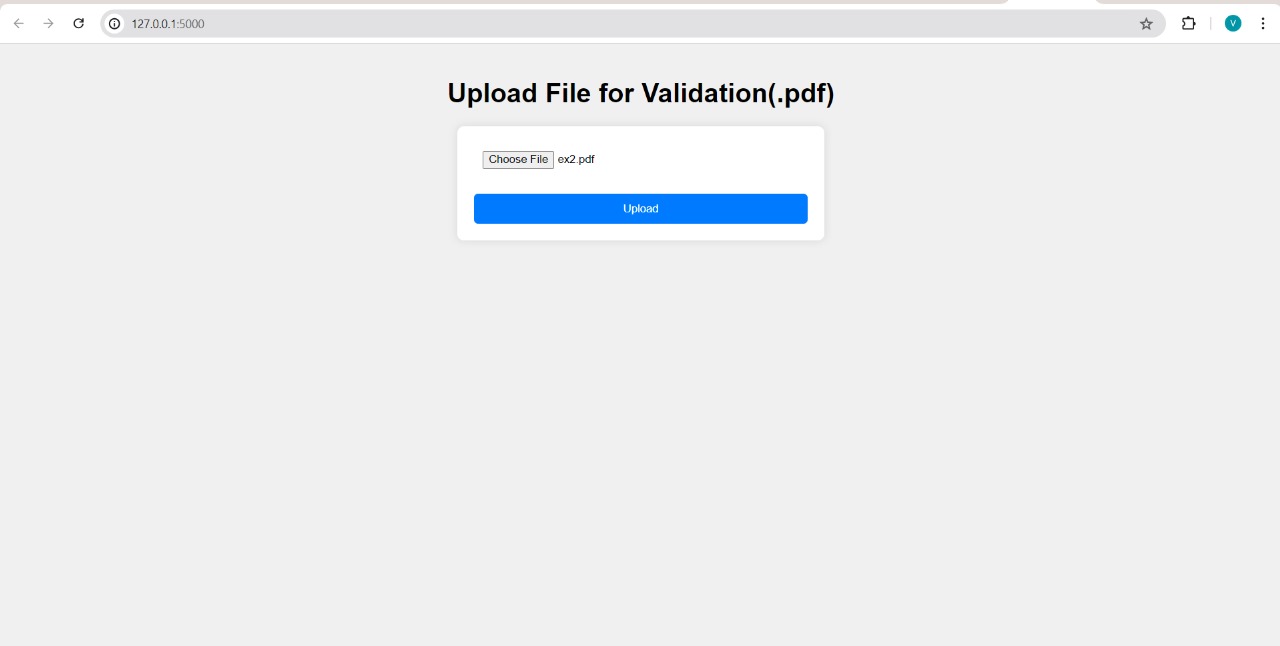
1. Install the “flask “ package using terminal, and run the code as “**flask run**”



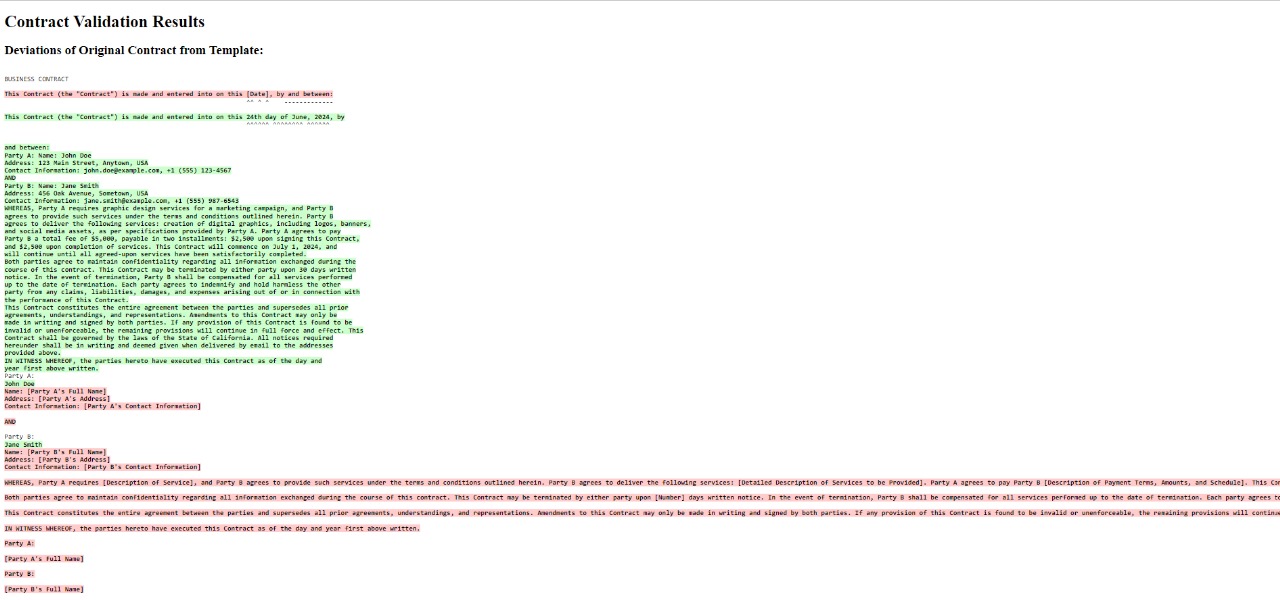
3.After Run the code if you want to quit then press “CTRL+C”



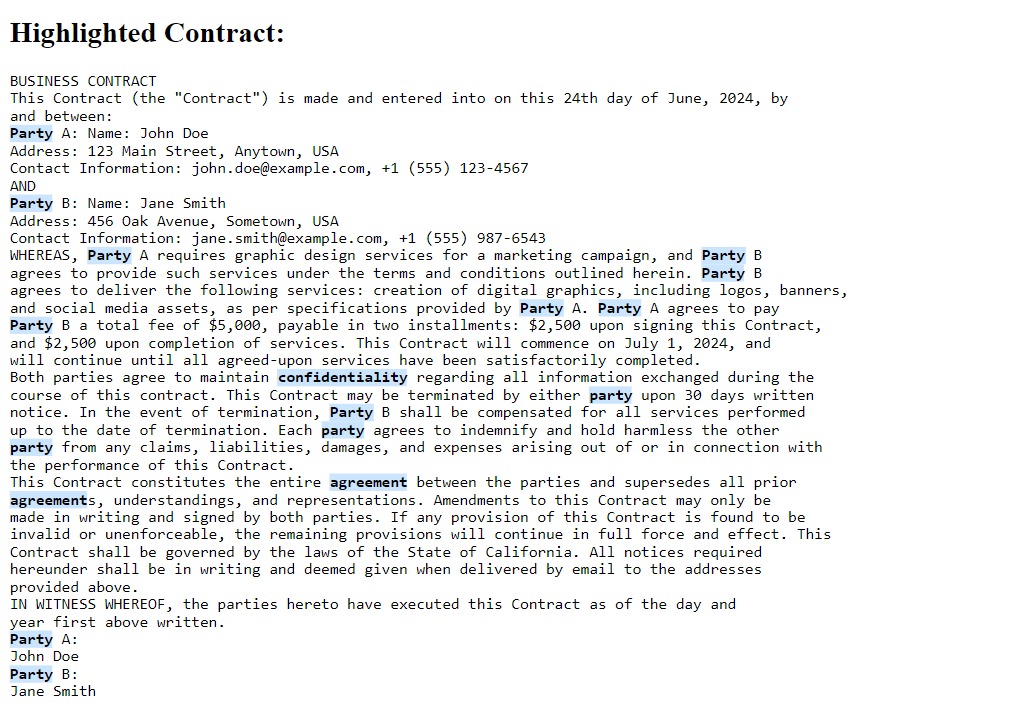
4.Upload the file in the form of pdf otherwise it won’t validate



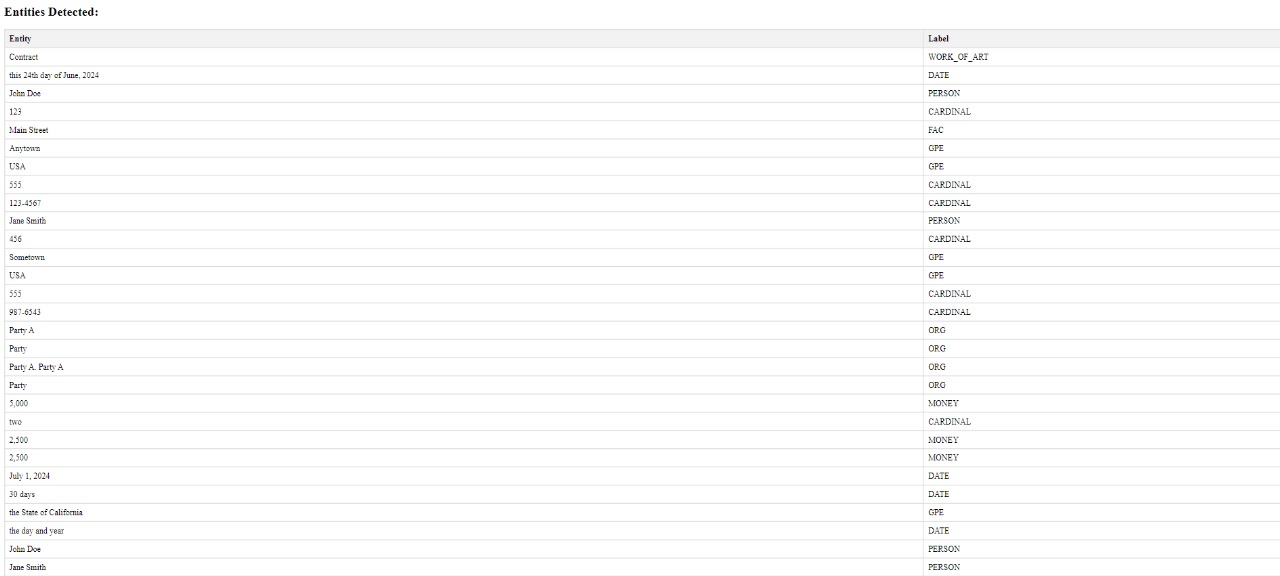
5.After uploading the file the ,It will show the deviations of Original Contract from template



6.The contract will be Highlighted as shown in figure



7.After Highlighting it will detect the entities



**Conclusion** :

In this project, we developed an automated system for business contract validation, leveraging natural language processing (NLP) and machine learning techniques. The system was designed to classify contract clauses, detect key entities, and identify deviations from predefined templates, addressing a critical need in legal and business practices for more efficient and accurate contract reviews.

The project's goals of automating the contract review process, enhancing accuracy and consistency, and streamlining compliance were successfully achieved. The system demonstrated the ability to classify clauses and detect entities with a high degree of accuracy, making it a valuable tool for legal professionals and businesses alike. The integration of template comparison and risk assessment further ensured that contracts could be validated against industry standards and legal requirements, reducing potential risks and ambiguities.

While the system showed promising results, several challenges were encountered, including handling the complexity of legal language and ensuring the scalability of the NLP models. These challenges highlighted areas for future improvement, such as refining the models to better handle nuanced legal terms and expanding the system's adaptability to various contract types and industries.

Overall, the project has laid a strong foundation for the continued development of automated contract validation systems. Future work could focus on enhancing customization options, improving model scalability, and integrating more advanced compliance checks. The success of this project demonstrates the potential of NLP and machine learning to revolutionize the field of contract management, making it more efficient, accurate, and accessible.