

**N L Dalmia Institute of Management Studies and Research**

Summer Internship Report

SUBMITTED BY

Name: Shivangi Gupta

Roll No: PBA2325-G020

Batch: 2023-2025

|  |
| --- |
| **Table of Contents** |
| **Company Overview: Cubic Tree Technology Solutions**...................................................... 2 The Industry: Legal Tech & SaaS...........................................................................................3 CubicTree’s Product Offerings............................................................................................. 4 Banking Industry: Litigation & Recovery Solutions............................................................ ..4 Corporate Sector: Legal & Compliance Management......................................................... 5 Law Firms and Lawyers: Case Management Tools............................................................. .6 Forensic and Risk Management Solutions.......................................................................... .7 |
| **Gaba Projects Private Limited - Initial Project Assignment**..................................................6 Introduction to the Project.................................................................................................... 8 Analyzing Existing Systems: Asana & Excel.............................................................................9 Creation of CubiTask Pro - A Tailored Solution..................................................................... 11 Data Gathering and Analysis................................................................................................. 12 Efficiency Gains and Improvements...................................................................................... 13 Comparison of CubiTask Pro with Excel................................................................................ 14 |
| **Tata Steel: Contract Management Case Study**.................................................................. 15 Introduction to Tata Steel's eContract Management System............................................ .16 System Design and Client Requirements Gathering............................................................ 17 ICMS Modules: Lifecycle, Compliance, Vendor Management............................................. 18 Python’s Role in Backend Development.............................................................................. 22 Frontend Development & Integration..................................................................................27 Final Product Delivery and Benefits for Tata Steel.............................................................. 30 |
| **AU Small Finance Bank: LegalTrack NPA Platform**............................................................31.  Introduction to LegalTrack NPA..........................................................................................34  Streamlining Litigation and NPA Processes.........................................................................36 API & Backend Development..............................................................................................41 Frontend Development Process and Design.......................................................................42 User Interface & Dashboard Overview...............................................................................45 Final Product Delivery and Bank Benefits...........................................................................50 |
| **Newberry Group Internship: Algorithmic Trading**............................................................51 Company Overview: Newberry Group...............................................................................53 Role of a Quantitative Analyst............................................................................................55 Trading Strategies: SMA Crossover & Inside Candle.......................................................... 56 Algorithm Development and Monitoring........................................................................... 57 Collaboration and Performance Monitoring.......................................................................59 |
| **Internship Reflections and Learning Outcomes**.................................................................60 Conclusion of the Internship Experience............................................................................ 61 |
| **References**.......................................................................................................................... 62 |

**Company Overview: Cubic Tree Technology Solutions**

Cubic Tree Technology Solutions has positioned itself as one of the leading providers within the legal technology market with innovative Software as a Service (SaaS) products serving diverse domains including banking, corporate, law firms, as well as forensic/risk management operations. With a goal of eliminating…and reinventing how legal work is done, Cubic Tree helps clients enhance their productivity by providing data, notifications, and management for legal processes. As the company grows and acquires such technologies as concepts of innovation and the use of automation, clients are able to conduct complex litigation as well as handle contract management with ease and within a limited time frame that is less than before.  
  
What makes Cubic Tree unique is its potential to offer solutions which include Litigation & Recovery Solutions, Borrower Profile Reports, Legal Audit Reports, Contract Management System among others. The company has managed to tap into products that collect, analyse and update legal and financial information instantaneously hence making Cubic Tree relevant for legal risk management with diverse portfolios on litigation and compliance.

**The Industry: Legal Tech & SaaS**

Increased demands and complexities in legal procedures, greater need for regulatory compliance, and a need for more efficient business processes are turning the legal technology industry into an ever-rising business. The sector uses emerging technologies in the AI, machine learning, and data analytics space in a bid to automate traditionally manual exercises like contract drafting and litigation tracking, as well as managing compliance and researching laws.  
SaaS has been one of the most dominant models in this niche because it is flexible and cost-effective, yet scalable. Companies, from legal firms to corporate bodies and financial institutions, are embracing SaaS solutions in order to automate their legal workflows and ensure minimal operational costs. It helps them make better decisions, utilizing data-driven insights. Cubic Tree fits perfectly in such an evolving landscape because it offers a product portfolio not only to the legal practitioners but also the businesses that demand a structured, automated method of dealing with their legal affairs.  
It led to an increased demand for special SaaS products because a large amount of litigation and compliance issues were to be serviced along with the contracts, which had to be processed faster. Companies like Cubic Tree make businesses better equipped to face litigation challenges combined with overall improvement in their risk management strategies while creating a competitive edge.

**CubicTree’s Product Offerings**

Banking Industry: Today's banking institutions operate in the most sophisticated manner and are becoming more digital in relation to legal procurement, borrower profiling, and financial risk reduction. The Litigation & Recovery Solution of Cubic Tree is designed for banks to be able to monitor and manage their litigation portfolios, especially the ones related to loan recoveries and disputes. The Borrower Profile Reports offer deep insights into the legal and financial situation of borrowers to the financial institutions, thus facilitating making wise lending decisions.

E-Contract Management Systems have simplified the contract lifecycle management process and provided banks with a way to maintain compliance and to free up time during contract renewals, amendments, and execution. Signals and E-Causelist Alerts are the real-time notifications of litigation that keep institutions abreast of the latest changes, thereby helping them know the legal developments that might impact their business.

Corporate Sector: For corporates, legal operations are integral to maintaining compliance, managing risk, and ensuring smooth business operations. Cubic Tree’s Casetrack/LMS Lite helps corporates track and manage litigation portfolios across different jurisdictions, centralizing information and improving case management. The E-Contract Management System offers the corporate sector the ability to automate and digitize contract creation, execution, and management, leading to increased efficiency in legal processes.

Products such as Borrower Profile Reports as well as E-Causelist Alerts also helps corporates by seeing the **whole legal and financial data it** allows the company to avoid risks when dealing with external bodies. These tools are important to decision-making processes, especially in mergers, acquisitions, and partnerships, where the legal standing of partners is a critical factor, which can be fulfilled by the products.

Law firms and lawyers are two examples of how law firms and independent legal practitioners are greatly benefiting from the Cubic Tree solution which is designed for the management of cases and researches in the legal field. The Legal Audit Reports give lawyers a complete view of case portfolios, making it easier for them to keep track of key legal parameters and to make decisions that are backed up by data.

The company’s flagships for the law firms which are Cubictreesmart is an AI-powered tool that provides insights into legal cases, suggesting strategies based on past outcomes and current legal landscapes. Through the E-Causelist Alerts, law firms will be able to keep themselves updated on court schedules, judgments, and legal precedents, thus their response time and preparedness for ongoing cases will be greatly improved.

The forensic and risk sector often requires real-time data in order to take concrete steps on litigation and compliance issues. The Borrower Profile Reports and Signals provided by Cubic Tree is one of the important products which can be used in this domain where a detective working for a bank would utilize a combination of financial and legal data to determine the proper level of risk associated with borrowers, companies, or transactions. Forensic teams are given the advantage to assess the risk related to (borrowers, companies, or transactions) by using the loaner’s profile data alongside the legal information.

Secondly, CoTrack, which is the other product provided by Cubic Tree, is the one that specifically focuses on managing complex legal cases that involve different entities or jurisdictions recently which are the frequent situations in forensic and risk investigations. The product assists forensic teams to monitor the development of cases by providing the information on the progress of cases that they can use to decide and take measures proactively and, thus, risks can be managed more effectively.

**Gaba Projects Private Limited - Initial Project Assignment at CubicTree**

1. A brief introduction to the project

At the first day of my internship at CubicTree, they obliged me to investigate several types of software solutions that CubicTree offers in different fields like Banking, Corporate, Legal and Risk Management. So my first task was to analyze the value our most recent client realized from the project management system built by Cubictree.

Gaba Projects Private Limited (GPPL) is a New Delhi, India-based IT. Before partnering with Cubic Tree, GPPL had been managing its projects on Asana along with Excel. They needed a way more robust and customizable solution as they scaled their company

Analyzing the existing systems

As part of this project, I studied GPPL's existing project management systems, which included Asana and Excel. Despite being integral to their daily operations, these tools presented several significant challenges as the company grew. In my analysis and discussions with key stakeholders, including GPPL's managers and directors, Manjeeri Desai and Mukta Bhardwaj, I identified the following operational challenges:

1. A lack of customization: GPPL was unable to customize Asana to meet their specific project management requirements. In order to manage multiple departments, approval stages, and project types, they needed more flexibility in configuring workflows. Asana's rigid templates, however, cannot be adapted to meet the specific requirements of different projects or teams.
2. Manual workflows with Excel: Asana handled task assignment, but Excel handled financial tracking and reporting. As a result of this dual-system approach, data entry was manual, which resulted in significant inefficiencies, duplication of work, and a higher risk of human error. The lack of real-time accuracy was caused, for example, by the manual updating of financial reports and resource allocations.
3. Asana's inability to notify project managers when a resource (team member or equipment) is already occupied was one of the biggest concerns. Overbooked team members were unable to fulfill their assigned tasks, resulting in frequent workflow interruptions. There was no resource allocation or alert system to manage workloads effectively, resulting in delays and confusion.
4. It was difficult for GPPL to track multiple project phases and financials with Asana's built-in reporting features. The manual creation of reports in Excel, on the other hand, was time-consuming and error-prone. Managing projects, budgeting, and resource utilization in real time was challenging, resulting in slower and less informed decision-making.
5. As GPPL began managing more complex projects, Asana and Excel struggled to handle workflows involving multiple steps, dependencies, and cross-departmental collaboration. Asana, for instance, did not provide a clear Gantt chart view or easy mapping of dependencies across tasks, making it difficult to track long-term project timelines. Often, this limitation resulted in missed deadlines and inefficient resource management.

Due to these challenges, GPPL needed a more robust, integrated project management solution that could address these pain points and scale as its project portfolio grew.

3. Creation of a tailored solution - CubiTask Pro

The Cubic Tree team, in collaboration with GPPL, developed a custom project management system called CubiTask Pro to overcome the limitations of Asana and Excel.

The following features were included in CubiTask Pro:

* CubiTask Pro's customizable workflows allowed GPPL to design workflows suited for different departments and types of projects, unlike Asana's limited options.
* CubiTask Pro automates GPPL's budget and expense tracking by integrating with its internal systems, eliminating the need for manual data entry.
* In addition, a reporting engine provides real-time insights into project progress, resource allocation, and financial metrics, addressing a major weakness of Excel.
* CubiTask Pro integrates Gantt charts to better visualize project timelines and dependencies, a feature not available in Asana.

**4**. Gathering and analyzing data

I interviewed GPPL's management team, particularly Manjeeri Desai and Mukta Bhardwaj, to evaluate the effectiveness of the newly implemented system. Data collected helped quantify the impact of CubiTask Pro.

The following are the main findings:

* CubiTask Pro's customizable workflows and automation have improved task assignment and project tracking by 35%.
* The integration of financial tracking reduced manual errors by 90%, resulting in more accurate budgets and expense reports.
* Compared to manual Excel reports, automated reporting tools reduced report generation time by 50%.
* Real-time communication reduced project delays by 20% by enabling team members to resolve issues within the platform, unlike Asana's scattered communication.

A graph of a bar chart

Description automatically generated

How the Percentage of improvement is calculated   
As shown in the bar graph above, CubiTask Pro has impacted Gaba Projects Private Limited's (GPPL) operations compared to benchmarks. Each category measures a key operational area.

1. Gains in efficiency (35%):
   * Benchmark : It used to take team leaders about 48 hours to update project statuses and manage tasks manually on Excel.
   * After Implementation : With automated workflows and better integration across departments, the time has been reduced to approximately 31.2 hours, resulting in a 35% increase in efficiency.
2. Reduction of errors (90%):

* Benchmark : The initial benchmark was 10-12 errors per project due to incorrect assignee data, inaccessible resources, or inaccurate deadlines and progress updates, which necessitated frequent corrections.
* After Implementation : As a result of CubiTask Pro's automation and validation rules, errors have been drastically reduced. Currently, these types of errors occur less than once per project on average, resulting in a 90% decrease in error frequency and an increase in data accuracy and operational reliability.

1. Time Savings (50%):
   * Benchmark : Initially, generating reports and assigning projects took about 20 hours per week.
   * After Implementation : Due to streamlined operations and automation features, this process now takes about 10 hours, which is a half-hour reduction in time.
2. Collaboration improved (20%):
   * Benchmark : Collaboration metrics were based on project completion times and cross-departmental communication ease.
   * After Implementation : As a result of improved system integration and communication tools, collaboration has increased by 20%, resulting in faster decision-making and more cohesive project execution.

Comparison of CubiTask Pro with Excel

During my interviews with Manjeeri Desai and Mukta Bhardwaj, I focused specifically on how CubiTask Pro improved GPPL's project management processes in comparison to their previous reliance on Excel for financial tracking and project reporting. Here’s a detailed comparison based on the data collected:

**1.** Improved efficiency by 35% compared to Excel

In terms of financial tracking and project management, Excel was a highly manual tool for GPPL. In order to keep track of each project's status, budget, and resource allocation, team members had to manually update them. It resulted in project managers spending considerable time updating spreadsheets, delaying other critical tasks.

CubiTask Pro automates many of these processes, particularly task assignment and progress tracking. A 35% improvement in efficiency was achieved through automated workflows and integration of tasks across departments.

Differences between the two:

* A high workload and delays are caused by manual entry of project updates in Excel.
* Improved operational efficiency and reduced manual intervention with CubiTask Pro.

**2**. Reduced errors by 90% compared to Excel

Due to the manual entry of financial data into Excel, using Excel for financial tracking introduced a high risk of errors. The result was not only an increased likelihood of making mistakes, but also a difficulty maintaining real-time accuracy across project budgets. There were often discrepancies in GPPL's reports that had to be manually corrected.

**By automating financial tracking, CubiTask Pro integrates directly with GPPL's internal systems and eliminates manual data entry.** By automating the data transfer between systems, 90% of errors were reduced. As a result, accuracy improved significantly.

Differences between the two:

* As a result of frequent data entry, Excel is prone to manual errors in tracking budgets and expenses.
* By automating financial tracking, CubiTask Pro reduces errors and improves financial accuracy by 90%**.**

**3.** Reduction in report generation time by 50% when compared to Excel

The process of creating reports using Excel was time-consuming. To track project progress, project managers had to manually collect data from various sources, update financial figures, and create charts. The decision-making process was delayed by several hours as a result.

Using CubiTask Pro's advanced reporting tools, reports were generated in real-time, with all relevant data automatically updated. By doing this, report generation was reduced by 50%, allowing managers to access the latest project information instantly.

Differences between the two:

* Creating reports manually in Excel was labor-intensive and prone to delays.
* Report generation time is reduced by half with CubiTask Pro's real-time automated reports.

**4.** Improved collaboration - 20% improvement over Excel

In Excel, team members could not work simultaneously or communicate efficiently within the tool due to a lack of collaborative features. A lack of communication, overbooked resources, and delays in project execution often resulted. To collaborate, team members had to use external communication tools, which further complicated workflows.

As opposed to CubiTask Pro, CubiTask Pro allows team members to collaborate and communicate in real-time. Additionally, the system enabled visibility into resource allocation, preventing overbooking and improving task completion times by 20%.

Differences between the two:

* Lack of collaboration and resource allocation tools led to frequent workflow disruptions.
* Project delays were reduced thanks to CubiTask Pro's real-time communication and resource allocation tools.

A screenshot of a computer

Description automatically generated

**5.** Conclusion

The successful implementation of CubiTask Pro at Gaba Projects Private Limited demonstrated Cubic Tree’s ability to tailor software solutions to meet specific business needs. By replacing the combination of Asana and Excel with a customized, integrated platform, GPPL experienced significant improvements in efficiency, accuracy, and collaboration. This project provided me with valuable insights into the practical application of customized software in real-world settings and laid the foundation for my further work at Cubic Tree.

This report marks the completion of my first assignment, where I had the opportunity to contribute to analyzing and developing tailored solutions for a client’s operational needs.

**Interface Overview of CubiTask Pro**

**A brief introduction to interfaces**

A major component of my internship project at Cubic Tree involved implementing and customizing CubiTask Pro for Gaba Projects Private Limited (GPPL). The following is an overview of CubiTask Pro, which has been specifically developed for GPPL to enhance project management processes.

**Dashboard -** Home

Here is a screenshot of the dashboard on the home page

CubiTask Pro's Home Dashboard serves as the central hub for daily activities. A personalized welcome message, "Good evening, Shivangi," enhances the user experience. This dashboard displays the user's weekly tasks, including a count of completed tasks and number of collaborators. Customizable widgets and an intuitive navigation bar allow users to switch between views and tasks easily.

A screenshot of a chat

Description automatically generated

Organizing tasks

The interface for managing tasks is shown in screenshot 2

Tracking and organizing daily responsibilities are crucial in the Task Management section. 'Tasks I've assigned' and 'My tasks' are divided into two panels, each with a filter for upcoming, overdue, and completed tasks. Dual-panel setup allows users to keep track of their own tasks as well as the progress of tasks assigned to them. Team members can use this interface to increase efficiency and accountability.

Overview of the Gantt chart

A Gantt chart is shown in screenshot 3

CubiTask Pro includes a Gantt Chart for managing more complex projects that require detailed planning and time management. By using the tool, users are able to visualize project timelines and dependencies at a glance. In order to maintain project timelines, this feature enables effective scheduling and identifies potential bottlenecks before they cause delays.

A screenshot of a computer

Description automatically generated

Statistics on tasks

The task statistics are shown in screenshot 4

This section provides a comprehensive analysis of task performance across various metrics**.** A detailed breakdown of tasks by section is also displayed, along with the number of tasks completed, in progress, and overdue. Managers can make informed decisions based on real-time data when assessing team productivity and project health.

A screenshot of a phone

Description automatically generated

**Utilizing SaaS-based solutions to improve contract management at Tata Steel**

Second task: Contribute to the development of the eContract Management System (ICMS)

**Tata Steel's Contract Management Database: A Case Study**

My internship at Cubic Tree focused on contract management efficiency for one of our major clients, Tata Steel. Cubic Tree provides Tata Steel with a comprehensive SaaS-based vendor procurement management solution. In order to procure essential raw materials such as iron ore and coal, Tata Steel engages with a vast network of vendors around the world. Tata Steel uses CubicTree's SaaS-based Integrated Contract Management System (ICMS) to track contracts from inception to completion, monitor compliance, and evaluate vendor performance.

To analyze the efficiency of contract management, I created a detailed sample dataset that mimics the real-world procurement data from Tata Steel. Key contract information is contained in this dataset, including vendor type (domestic or international), material type, contract value, and crucial dates such as initiation, review, approval, and signing. As well as tracking compliance status, breach occurrences, approval cycles, and vendor performance, it also tracks the performance of vendors. This sample data was carefully designed to match the columns and structure of Tata Steel's actual data, ensuring both accuracy and relevance of the analysis.

A dataset containing 50 contracts was used to evaluate key performance indicators, such as turnaround times for contract approvals, compliance rates, and vendor performance scores. As a result of the analysis, inefficiencies in the procurement process were identified, including delays with international vendors and compliance problems in specific types of contracts. Taking advantage of these findings will enable Tata Steel to improve its procurement efficiency by automating processes, improving compliance checks, and improving vendor management, ultimately contributing to a more efficient process.

**Tata Steel: How Cubic Tree Built the eContract Management System**

Through my internship at Cubic Tree, I gained a deep understanding of how the company tailors its SaaS products for clients like Tata Steel. The development of Tata Steel's eContract Management System (ICMS) required a structured approach to meet their unique requirements, streamline their contract processes, and ensure compliance with internal policies and external regulations. In the following table, you will find a breakdown of how Cubic Tree designed and developed Tata Steel’s ICMS:

**1.     Gathering client requirements**

During the initial consultation:

During the development process, Cubic Tree met with external departments at Tata Steel, such as procurement (who handle purchasing materials), legal (who check contracts), and IT (who manage technology). Tata Steel wanted to understand the specific problems it faced with contract management. They discussed the following challenges:

Tata Steel works with numerous vendors (suppliers) all over the world, so a system was needed that could manage all these contracts, regardless of the location or type of vendor.

The process of reviewing, approving, and signing contracts was mostly manual and took a lot of time. In order to make the process faster and more efficient, they needed a system that could automate these steps.

Tata Steel already had systems in place for managing finances and procurement (like an ERP system), so the new contract system had to integrate seamlessly.

Documentation of Detailed Requirements:

As a result of these discussions, Cubic Tree's team wrote a detailed document listing all the features Tata Steel needed in the new eContract Management System. The document acted as a blueprint for developing the system, ensuring Tata Steel's requirements were clearly understood and would be implemented.

1. **Understanding the Data in the Context of System Architecture:**

**A screenshot of a computer screen

Description automatically generated**

The modular design consists of:

Lifecycle Management Module: The extracted data will include key details such as Contract\_ID, Vendor\_Name, Material\_Type, and various important dates such as are the crucial stages of contract lifecycle management. The ICMS modular system would be designed to handle each stage of this lifecycle, allowing the system to:

1. Monitor the status of contracts at each stage (pending review, under approval, signed).
2. Use workflow rules to automatically move contracts through stages.
3. Allow users to see real-time contract progress.

For example, a contract might be initiated on "02/01/2024" (as seen in **TA456001**) and moved automatically through the review and approval process, with notifications sent to stakeholders at each stage.

Risk and Compliance Module:

1. Compliance\_Status and Breach\_Status columns indicate whether the contract is in compliance with Tata Steel's internal policies and legal requirements. The ICMS would automatically check each contract against Tata Steel's internal guidelines and external regulations, flagging any non-compliance or breaches for review.
2. A risk assessment would be provided by the system as well. In this example, if the contract TA456002 has a "Yes" under Breach\_Status and is marked as non-compliant, the system would alert legal or compliance.
3. By implementing this module, Tata Steel is able to stay within regulatory frameworks, reducing legal risks.

Vendor Performance Module:

1. The Vendor\_Performance\_Score is crucial for tracking how well vendors fulfill their obligations (such as delivering materials on time and maintaining quality). The ICMS would store this data and use it to generate performance reports for Tata Steel.
2. Over time, Tata Steel can analyze these scores to determine which vendors consistently perform well and should be prioritized for future contracts, as well as which vendors need improvement. For example, TA456003 has a Vendor\_Performance\_Score of 9, indicating strong performance, while TA456017 with a score of 6.2 may require further evaluation.
   1. The system could also provide insights into how vendor performance correlates with contract outcomes (e.g., delays in signing or breaches).

**The customization process for Tata Steel is as follows:**

* Depending on the material, vendor type (domestic or international), and contract value, each contract follows a different workflow. The sample data indicates that contracts with international vendors (TA456002) may require additional approval steps than domestic vendors (TA456001), which can be configured in the ICMS.
* Tata Steel's system might use custom templates for different types of contracts (iron ore, coal, etc.). It provides predefined templates and approval paths that can be customized based on factors such as material type, vendor performance, and contract value. As an example, a contract with a value of $7,800,000 may require approval from senior management, as oppose to a contract with a value of $4,500,000.

The key takeaways are:

* ICMS allows Tata Steel to track and monitor all contracts in real time, ensuring that each contract's lifecycle is monitored.
* Management of compliance and risk: The system automatically flags non-compliant contracts and helps mitigate risk.
* In order to make data-driven decisions regarding future procurement contracts, Tata Steel evaluates vendor performance over time.

As a result of this architecture, Tata Steel manages large volumes of contracts efficiently, ensures compliance, and improves vendor relationships, ultimately optimizing their procurement activities.

**In what ways will Python be used in the final software**

**1. Development of the backend**

In addition to managing contract data, automating workflows, and conducting calculations, Python serves as a powerful backend language. This application interacts with databases, processes business logic, and ensures that all contract-related operations run smoothly. This is how it fits into the system:

* **Data Management: Contract data (e.g., contract ID, vendor name, dates, compliance status) will be stored in a relational database such as MySQL or PostgreSQL.** The data will be fetched, processed, and updated using Python scripts. Through APIs or web services, data flows between the user interface (UI) and the backend.
* **Business Logic**: Python is responsible for automating the business logic. For example:
  + Contract approval is automated by moving contracts through various stages according to predefined rules.
  + Creating risk alerts and checking compliance.
  + The calculation and updating of vendor performance scores.
* **APIs: Python code exposes APIs that enable user interfaces to interact with backends.** As an example, the Python code returns contract status information when a user logs into the system using an API.

Backend development for Tata Steel's Contract Management System focused on creating a robust, scalable solution. Below is a Python script that shows how contract details are managed, including adding new contracts, updating contract statuses, and checking for expiration and compliance:

****

**2. Development of the user interface (frontend)**

HTML, CSS, JavaScript, and ReactJS are web development technologies that can be used for the frontend of Tata Steel's application. Python handles the backend logic, but HTML, CSS, JavaScript, and ReactJS are web development technologies that can be used for the frontend.

* Scripts (running on the backend) will communicate with the frontend through APIs and AJAX calls. Front-end users can view contract statuses, input data, and receive real-time feedback on contract progress.
* **The system's interface will be used by Tata Steel employees (e.g., procurement and legal departments) to upload contracts, track approvals, check vendor scores, and manage compliance.** These interactions are powered by Python logic running in the background.

As an example:

* Upon logging into the system, users are able to view the status of their contracts. As soon as the front-end sends a request to the back-end (Python), Python fetches the relevant contract data from the database and displays it on the user interface.

**A computer screen shot of a code

Description automatically generated**

**Frontend Fetches Data from API**

In order to retrieve contract data, the frontend (HTML/JavaScript) will send a request to the backend (via API) when an employee opens the dashboard, searches for contracts, or filters data.

**Behind the Scenes Process:**

* **Frontend (UI)**: The JavaScript code sends a request to the Python backend's /api/contracts endpoint when an employee clicks "View Contracts.".
* **Backend (API)**: Python Flask retrieves the contract data from the database and returns it in JSON format to the frontend.

**from flask import Flask, jsonify, request**

* **Flask**: This imports the code for the Flask class used to create web applications.
* **jsonify**: Converts Python dictionaries (or lists) into JSON responses that can be understood by the front end (HTML/JavaScript).
* **Request: This object allows you to capture and handle incoming HTTP requests.**

**A screenshot of a computer program

Description automatically generated**

An intuitive, user-friendly interface was developed for Tata Steel's Contract Management System. This HTML code illustrates the layout and design of the system, which enables users to view and manage contract details effectively:

**3. Integration of databases**

In the database, Python will also be responsible for storing and retrieving data. There are several types of data stored in the database of the system, including:

* The vendor's details
* Lifecycle stages of a contract
* Metrics of performance
* Risk and compliance data

**Python scripts can interact with the database using SQLAlchemy (an Object Relational Mapping tool) or raw SQL queries.** The Python interpreter will handle:

* Adding contract details to the database by reading and writing them.
* Update contract statuses after approvals or sign-offs.
* Auditing contracts by keeping a log of all actions.

**4. Scheduling and automation of tasks**

There are some Python scripts that need to run periodically (such as compliance checks and sending notifications). It is possible to automate recurring tasks using Celery (task queue) and Redis (task scheduling).

As an example:

* Schedule Python scripts to run every day to check if any contracts are nearing compliance deadlines and flag non-compliant contracts. The responsible teams will receive alerts.
* The Python language can be used to automate the sending of email notifications when important contract milestones are approaching, such as contract renewals or expirations.

**5. Microservices and APIs**

As part of a microservices architecture, Python will be used in large systems like Tata Steel's eContract Management System. The different services (such as contract approval, compliance checking, vendor performance tracking) will run on separate Python servers and communicate using APIs.

* An API based on REST will expose the business logic via REST APIs, and will be written as Python (using frameworks like Flask or Django). When a user requests vendor performance metrics, the Python API will retrieve, process, and return the information requested by the user.

**6. Advanced Feature: Machine Learning and Analytics**

In the future, Python will be used to integrate machine learning models to provide predictive insights into vendor risks and contract performance. A few examples are:

* Based on certain patterns, the system can predict which contracts are likely to face compliance issues or delays using Python's scikit-learn library.
* In addition, Python could analyze vendor performance over time and recommend high-performing vendors for future contracts, optimizing the procurement process.

**7. The deployment process**

The Python backend of the system, along with the entire ICMS, will be deployed to a cloud infrastructure or a Tata Steel server once it has been developed, tested, and validated. The deployment process is as follows:

* With Docker, Python applications can be packaged and deployed in containers. The application can be deployed in any environment (cloud or on-premise).
* A continuous integration/continuous deployment (CI/CD) process will ensure automated testing and smooth deployment updates by pushing Python code to repositories (e.g., GitHub).

**8. Support and training for users**

Tata Steel's users will be trained on how to use the interface once the system goes live. The Python logic developed in the backend will power contract lifecycle management, compliance checks, and performance tracking.

**9. Maintenance and monitoring**

In the post-deployment phase, Python's logging and monitoring tools (such as Flask logging or third-party services like New Relic) will monitor the application's performance.

* Monitoring Performance: Python scripts will monitor backend performance, track database queries, and ensure the system remains responsive.
* Error Handling and Bug Fixes: Python's error logs catch any issues that arise, providing quick and efficient troubleshooting for the Cubic Tree development team.

**Delivery of the final product to Tata Steel**

Our final product for Tata Steel will be a fully integrated web-based application with Python as the backend. The Python code itself will not be visible to Tata Steel's users, but it will be the engine that drives all the business logic, automation, and contract processing.

The high-level process is as follows:

1. A user-friendly web interface is used by Tata Steel employees.
2. Python handles the data processing, contract tracking, compliance checks, and vendor performance evaluations.
3. We maintain a secure database that stores all contract data, milestones, and performance metrics.
4. A smooth data exchange is assured by Python's REST APIs that communicate with the frontend.
5. Python automates key tasks such as compliance alerts and contract approvals, reducing Tata Steel's manual workload.

In this way, Tata Steel has been able to streamline their procurement processes, track vendor performance, and ensure compliance with internal and external regulatory requirements through a comprehensive, highly efficient eContract Management System (ICMS).

**Frontend Interaction (UI for Tata Steel Employees)**

The Tata Steel dashboard (User Interface - UI) will be built using HTML/JavaScript or a more advanced frontend framework like ReactJS. When they log in to manage or view contracts, they see this dashboard. They will not see raw JSON data like the example below, but rather a table, chart, or form displaying the data meaningfully.

**Dashboard Interaction by Employees**

* An employee will open a browser and navigate to a secure URL such as https://icms.tatasteel.com/contracts to access the dashboard.
* Feature of the dashboard:
  + Contract View: The employee will see a table with all contract details (Contract ID, Vendor Name, Material, Status, etc.).
  + Contracts can be searched by vendor name or ID.
  + A user can filter contracts according to the status of the contract (Pending, Approved, Signed).
  + Organize Contracts: Contracts can be sorted by their value or by date of signing in order to find the most relevant ones.

**ICMS Interface Overview for Tata Steel**

**1. Dashboard Overview**

* **Screenshot**: (Reference to the first screenshot)  
  A screenshot of a computer

  Description automatically generated
* **Description**: The main dashboard provides a succinct overview of the entire contract management lifecycle at Tata Steel. The welcome banner "Welcome, Shivangi Gupta!" personalizes the experience for the logged-in user, enhancing user interaction with the system.
  + Total Contracts: Displays the total number of active contracts across all departments and vendors, providing a quick numerical insight into ongoing operations.
  + Pending Approvals: Shows contracts that are awaiting approval, highlighting bottlenecks and prioritizing tasks that require immediate attention.
  + Compliant Contracts: Indicates the percentage of contracts that meet compliance standards, ensuring that contract practices adhere to regulatory and internal guidelines.

**2. Contract Overview**

* **Screenshot**: (Reference to the second screenshot)  
  A pie chart with a pie chart and a blue background

  Description automatically generated with medium confidence

**Description : -** An important feature of the dashboard is the distribution and status section, which provides detailed information about the stages of the lifecycle of a contract.

* Contract Status Distribution: A pie chart illustrating the proportion of active, pending, and completed contracts. This allows quick assessment of workflow efficiency.
* Vendor Value by Vendor Type: A bar graph illustrating the differences in the price between domestic and international vendors, which aids in the management of strategic vendors and financial oversight.

**3. Vendor Performance**

* **Screenshot**: (Reference to the third screenshot)  
  A screenshot of a graph

  Description automatically generated

**Description:** This interface of the site shows how easy it is for a visual and analytical review of vendor performance, which is essential for maintaining quality and efficiency. This vendor performance feature on the site is proven to be very helpful when choosing vendors for the set products to be procured by the company.  
  
A bar graph helps us identify which vendors are performing well and which ones are underperforming by visualizing their performance scores. A strategic choice on how to partner with vendors and renew contracts can be made with the help of this tool.  
  
In-depth Vendor Analysis: A table that provides detailed insights that influence procurement strategies, including average contract value, compliance rate, and approval times for each vendor.

**4. Compliance Monitoring**

* **Screenshot**: (Reference to the fourth screenshot)  
    
  A screenshot of a computer

  Description automatically generated

**Description**: -Focuses on the implementation of compliance standards within all contracts to ensure that they are adhered to over time, which is crucial for minimizing the risk and maintaining the legal integrity of the contract. A screenshot of a computer

Description automatically generated

A compliance status pie chart shows how compliance status of contracts is distributed across the contract life cycle, highlighting the percentage of contracts that are compliant and those that are non-compliant.

Button to Generate Compliant eContracts: A large green button that has been strategically placed on the site allows users to generate new contracts with minimal effort, simplifying the contract creation process.

In addition to the standard compliance tools, the system provides advanced features such as real-time compliance alerts and automated contract audits, which enhance the system's capability to manage contract risks and compliance dynamically at the same time.

Lastly,

The ICMS designed by CubicTree for Tata Steel is a comprehensive contract management solution that streamlines processes and incorporates advanced analytical tools to ensure compliance and efficiency. Tata Steel's customized interface, from the dashboard overview to detailed compliance monitoring, demonstrates a deep understanding of enterprise-level contract management challenges and solutions. In addition to enhancing operational effectiveness and vendor relationships, this system significantly facilitates strategic decision-making.

**Banking Litigation, NPA & Recovery Platform for AU Small Finance Bank**

As part of my internship at CubicTree, I was assigned another critical project that involved assisting in the development of a **Banking Litigation, NPA (Non-Performing Assets) & Recovery Platform** for **AU Small Finance Bank**. This platform aimed to streamline the bank’s complex processes around litigation management, NPA tracking, and recovery workflows. The objective was to provide AU Small Finance Bank with a centralized, automated solution to handle their legal cases, identify and manage NPAs, and ensure efficient recovery efforts.

The project was a large-scale initiative, involving multiple teams, and I played a key role throughout the development process. My primary contribution focused on **data management**, where I worked on organizing and structuring the platform’s backend data models, ensuring that all the necessary information about litigation cases, NPAs, and recovery tasks was stored and processed efficiently. This data-centric approach was essential for the platform’s success, as it enabled the automation of critical processes, the generation of real-time insights, and seamless integration with the bank’s existing systems.

The platform aimed to enhance AU Small Finance Bank’s operational efficiency, reduce manual workloads, and ensure compliance with banking regulations, particularly around the handling of NPAs and legal cases.

**Streamlining litigation, NPAs, and recovery with LegalTrack NPA**

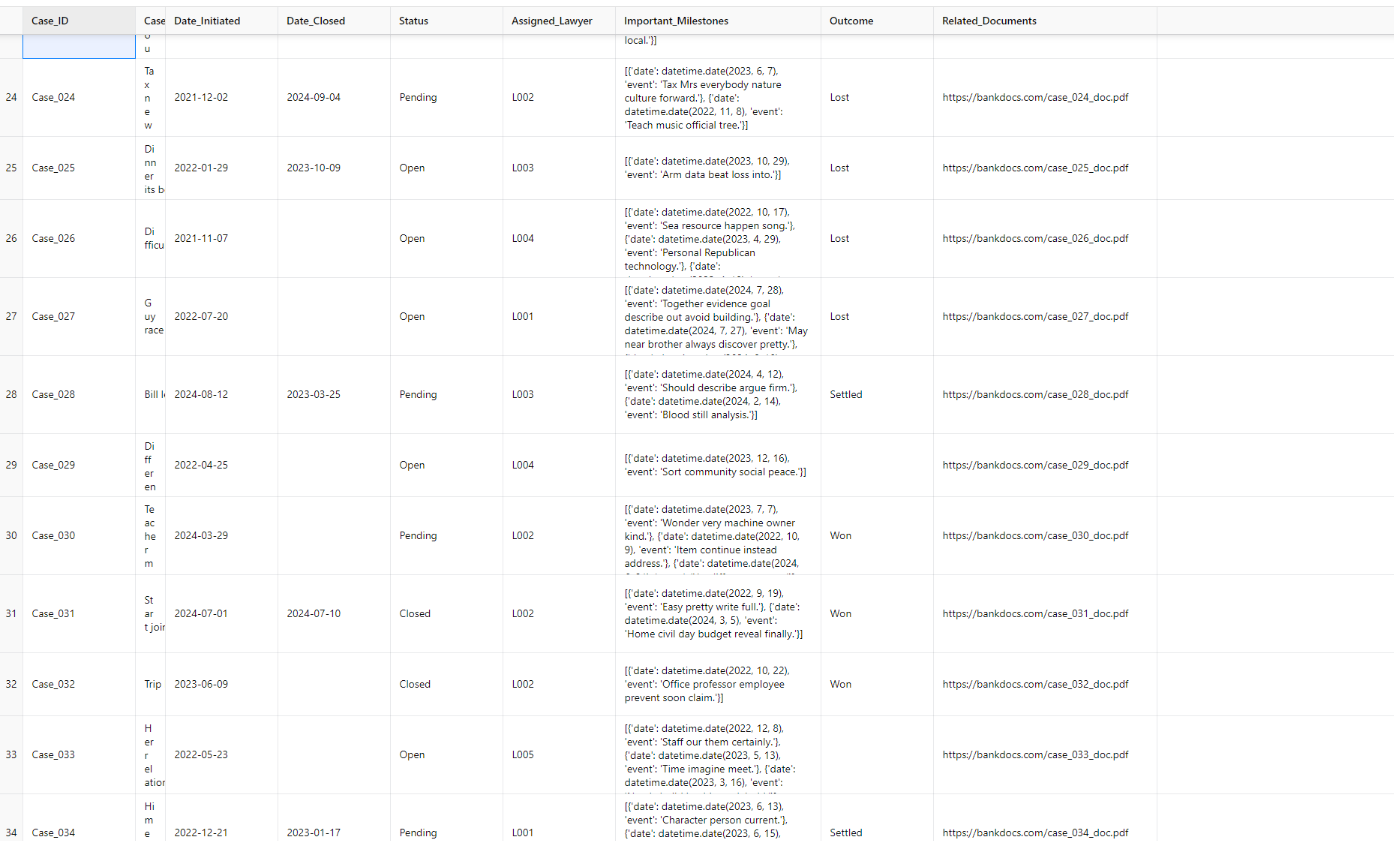
A platform was developed for AU Small Finance Bank and positioned as "LegalTrack NPA", a comprehensive solution that was designed in order to centralize and automate the bank's litigation management, NPA tracking, and recovery processes. The purpose of LegalTrack NPA is to help the bank meet its growing need for a robust system that would be able to streamline operations, reduce manual workloads, and ensure compliance with regulatory requirements as well.

Being an essential team member within the development team, I was deeply involved in the design and management of the data infrastructure that would be used to power the LegalTrack NPA application. In the course of my work, I primarily focused on the creation of efficient data models for storing, processing, and analyzing all relevant information about litigation cases, NPAs, and recovery tasks in litigation cases. With this structured data approach, we were able to automate processes such as case tracking, risk scoring for non-performing assets, and automated recovery workflows based on structured data.

**The main objectives of LegalTrack NPA are as follows:**

1. Providing real-time updates to the legal teams and allowing seamless document management, Litigation Case Management tracks all ongoing and closed litigation cases. This helps ensure accuracy and up-to-date case information, and makes it easier to track the progress of cases. It also helps to reduce administrative costs and increase the efficiency of legal processes.
2. In order to take proactive measures to recover NPAs, the bank will be able to identify and categorize NPAs based on predefined risk factors in order to identify and categorize them automatically. The bank will also be able to use AI-powered analytics to identify patterns and predict future trends of the NPAs. This will enable them to take timely actions to prevent NPAs from escalating.
3. An automated recovery workflow will automate the entire recovery process with real-time notifications, task assignments, and tracking the success of the recovery efforts in real-time. This automation will streamline the recovery process and provide visibility into the process. It will also reduce the time and resources needed for the recovery process.

Through real-time data insights, the system provides AU Small Finance Bank with the tools it needs to significantly improve its operational efficiency in managing NPAs and litigations, reducing recovery time, and improving decision-making capabilities.



**Litigation management in LegalTrack NPA**

This screenshot shows how LegalTrack NPA utilizes CubiData Suite (The database management system from cubictree) to process and manage detailed litigation case information. Here is a brief description of how each component of the data contributes to litigation management:

1. In order to facilitate quick referencing and searching within the system, each litigation case is assigned a unique identifier and a brief title.
2. The initiation date and possible closure date are crucial to tracking the timeline of a case, helping legal teams manage deadlines based on the urgency of the case, and prioritizing actions based on the urgency.
3. Status and Outcome: The system tracks the status of cases (e.g., Open, Pending, Closed) as well as the outcome of cases that are resolved (e.g., Won, Lost, Settled), as well as the outcome of open cases. In order to assess the effectiveness of legal strategies and adjust litigation approaches for the future, this information is vital.
4. The LegalTrack NPA assigns cases to lawyers based on their expertise and current caseload, ensuring efficient resource allocation. The tracking of lawyer assignments facilitates accountability and performance evaluation. All cases are handled by experienced lawyers in a timely manner through this system. Also, it ensures that lawyers receive appropriate feedback and are held accountable for their performance.
5. Each case contains a timeline of key events and milestones that are logged with dates and descriptions. The feature facilitates detailed monitoring and ensures timely completion of critical actions.
6. Documents related to the case: Each case record contains direct links to all the relevant legal documents related to the case, which makes it easier for any lawyer to access the papers and evidence that they may need. In the end, this feature enhances the flow of documentation, allowing all documents relevant to legal proceedings to be easily reviewed and used as evidence in the course of legal proceedings, ensuring that they are easily accessible during the course of legal proceedings.
7. By integrating the data with other modules within LegalTrack NPA, you will be able to generate insightful reports about litigation trends, attorney performance, and case outcomes by integrating these data with other modules within LegalTrack NPA. It will then be possible to use this data to identify areas of improvement and to better manage litigation costs. It is also important to note that it is still possible for it to provide valuable insights into the effectiveness of legal strategies.PA. By integrating the two, a holistic picture of the efficiency and effectiveness of the legal department can be achieved.



The following spreadsheet is a practical application of LegalTrack NPA's robust data management for Non-Performing Assets (NPA). Each NPA account is outlined in the report with critical financial and operational information. The spreadsheet's columns are explained in detail below along with the value they provide for effectively managing NPAs:

1. **In the system, each row represents a unique NPA case identified by an account number, which serves as a primary key.**
2. **Identifying the customer associated with each NPA is essential for personalizing communication and recovery efforts.**
3. **Obtains an understanding of the scale of the financial agreement by displaying the original loan amount issued to the customer.**
4. **Current Outstanding Amount: Displays the current outstanding balance on the loan, which is crucial for determining the amount to be repaid.**
5. **A customer's last payment date is used to track customer activity and identify prolonged periods of inactivity.**
6. **The date on which the loan was classified as a non-performing account, providing a timeline for how long it has been non-performing.**
7. **Categorizes each NPA by risk level (Low, Medium, High), guiding recovery efforts according to potential loss severity.**
8. **A recovery status indicates the current stage of the recovery process (e.g., in progress, not started, recovered), which assists in prioritizing actions and allocating resources.**
9. **A recovery officer is responsible for managing the recovery process, ensuring accountability, and facilitating performance evaluations of NPAs.**

The backend development phase was primarily handled in Python, which was the first and most critical phase of software development. As a result of this phase, the foundational functionalities of the system were laid, which ensured that the system was capable of managing both Non-Performing Assets (NPAs) and litigation cases efficiently for AU Small Finance Bank.

**Overview of backend development:**

All data processing, business logic, and database interactions take place in the backend of LegalTrack NPA. In order to ensure its robustness, scalability, and security, our team took meticulous steps in its design and implementation.

  1. **Creating a development environment:**

* First, we set up a Python-based development environment that included an integrated development environment (IDE), dependency management tools, and a version control system. As a result, all team members were able to collaborate effectively and maintain consistency throughout the development process.

  2. **Integration of databases:**

* By using libraries such as SQLAlchemy for ORM (Object-Relational Mapping), we were able to seamlessly integrate Python with the database management system chosen for this project. For handling the vast amount of data related to NPAs and legal cases, this integration enabled efficient data transactions and complex queries.

  3. **Developing the core functionality:**

* In order to serve the frontend requests, robust Python code was written to create APIs (Application Programming Interfaces). APIs were designed to handle tasks such as user authentication, data retrieval, data updating, and analytics.
* Since these APIs would be handling sensitive financial and legal data, special attention was given to their security and efficiency.

4. **Business logic implementation:**

* One of the most critical components of the backend was the business logic layer. The rules and procedures governing NPA operations and the progression of litigation cases are encoded here.
* Due to Python's clear syntax and powerful libraries, we were able to implement complex algorithms to automate risk assessments, categorize NPAs, and route cases through the appropriate recovery or legal processes.

  5. **Measures for security:**

* It was imperative to implement robust security measures given the sensitive nature of the data involved. Python's capabilities were used to enforce secure access controls, data encryption, and secure data transmission protocols.
* To identify and mitigate potential vulnerabilities promptly, regular security audits were integrated into the development lifecycle.

  6. **Optimization and testing:**

* Testing was an integral part of the development process. To ensure the backend's reliability and performance under various scenarios, we used Python's testing frameworks to conduct unit tests, integration tests, and stress tests.
* Additionally, the backend was optimized to handle high volumes of data and requests, ensuring the system remained responsive and stable.

  7. **Supporting documentation:**

* Throughout the development process, comprehensive documentation was maintained. Both the current maintenance team and future developers will benefit from this documentation, which covers setup instructions, API usage, known issues, and troubleshooting tips.

A screenshot of a computer code

Description automatically generated

**Backend Development Conclusion:**

The backend development phase laid the technical foundation for LegalTrack NPA to function as an effective tool for AU Small Finance Bank. Python's capabilities enabled us to build a secure, efficient, and scalable backend that underpins the bank's complex functionality for managing NPAs and legal cases. As a result of this phase, the system's core operations were made robust and reliable, laying the foundation for subsequent front-end development.

A computer code with text

Description automatically generated with medium confidenceA screenshot of a computer code

Description automatically generated

In the LegalTrack NPA project, APIs provide the interface between the backend, which handles data and business logic, and the frontend, which provides the user interface. On the basis of the provided code snippets, here's how the API functions are implemented and used:

**API Functionality Overview**

1. The API route and method are as follows:
   * The API route /api/npa/<int:id> is intended for GET requests. In this route, users can request information about a specific Non-Performing Asset (NPA) using its unique identifier, denoted by <int:id>.
   * Using the int in the route ensures that the ID is interpreted as an integer, which is a common practice in databases.
2. Data retrieval for NPAs:
   * A NPA case is retrieved from the database by its ID using the function get\_npa(id) mapped to the route.
   * By calling NPA.query.get(id), it retrieves an instance of the NPA model whose primary key matches the given ID. In order to find the corresponding data, this method interacts directly with the database.
3. Representation of data:
   * The to\_dict() method of the NPA model class is called if an NPA case is found. In this method, the NPA object is converted into a dictionary format suitable for serialization in JSON. Keys in this dictionary include "id", "customer\_name", and "loan\_amount", which represent the details of the NPA.
   * In the client-side of the application, the dictionary is returned as a JSON response using Flask's jsonify function.
4. Handling errors:
   * It returns a JSON response with an HTTP status code of 404 if no NPA case is found for the given ID (i.e., NPA.query.get(id) returns None). As a result, the client receives an error message stating that the resource requested does not exist.

An example in practice:

Consider a scenario in which a user on the frontend wants to view details about an NPA with ID 123. A GET request would be made to /api/npa/123 by the frontend. This request is handled by the backend as follows:

* With id set to 123, the get\_npa function is triggered.
* An attempt is made to retrieve an NPA case from the database using this function.
* A JSON object is returned if the case exists.
* A message indicating that no NPA case was found is returned if no case is found.

Using this API structure ensures that the frontend can effectively present NPA data to users, supporting efficient decision-making processes in managing NPAs. Furthermore, it separates the business logic from the client side, allowing the backend to handle data management and security independently.

For the LegalTrack NPA project, APIs provide the interface between the backend system, which handles data and business logic, and the frontend, which provides user interfaces. Based on the provided code snippets, here is how API functions are implemented and utilized within the project:

An overview of API functionality

1. The API route and method are as follows:
   * The API route /api/npa/<int:id> is intended for GET requests. In this route, users can request information about a specific Non-Performing Asset (NPA) using its unique identifier, denoted by <int:id>.
   * Using the int in the route ensures that the ID is interpreted as an integer, which is a common practice in databases.
2. Data retrieval for NPAs:
   * A NPA case is retrieved from the database by its ID using the function get\_npa(id) mapped to the route.
   * By calling NPA.query.get(id), it retrieves an instance of the NPA model whose primary key matches the given ID. Directly interacting with the database, this method returns the corresponding data.
3. Representation of data:
   * NPA model class's to\_dict() method is called if an NPA case is found. The NPA object is converted into a dictionary format suitable for serialization using JSON. Keys in this dictionary include "id", "customer\_name", and "loan\_amount", which represent the details of the NPA.
   * In the client-side of the application, the dictionary is returned as a JSON response using Flask's jsonify function.
4. Handling errors:
   * It returns a JSON response with an HTTP status code of 404 if no NPA case is found for the given ID (i.e., NPA.query.get(id) returns None). As a result, the client receives an error message stating that the resource requested does not exist.

An example:

Consider a scenario in which a user on the frontend wants to view details about an NPA with ID 123. A GET request would be made to /api/npa/123 by the frontend. This request is handled by the backend as follows:

* With id set to 123, the get\_npa function is triggered.
* An attempt is made to retrieve an NPA case from the database using this function.
* The case details are returned as a JSON object if the case exists.
* A message indicating that no NPA case was found is returned if no case is found.

Using this API structure ensures that the frontend can effectively present NPA data to users, supporting efficient decision-making processes in managing NPAs. Furthermore, it separates the business logic from the client side, allowing the backend to handle data management and security independently.

Our task as part of the development team is to build an intuitive and user-friendly interface that interacts with the backend APIs we've established as part of the next critical phase of the LegalTrack NPA project. In this phase, you will determine how end-users will interact with the system to manage and view Non-Performing Assets (NPAs) and associated litigation.

The frontend development process is as follows:

1. Prototyping and Design:
   * User Interface and User Experience (UI/UX) Design: The first step is to design the user interface and user experience (UI/UX). It involves creating a layout that is intuitive and aligns with the LegalTrack NPA system's functional requirements. To create wireframes and prototypes, tools such as Adobe XD, Sketch, and Figma are typically used.
   * The prototypes are reviewed by stakeholders from the AU Small Finance Bank to gather feedback and ensure the design meets their expectations.
2. Selection of the technology stack:
   * Framework selection: For frontend development, we might choose frameworks such as React, Angular, or Vue.js based on their scalability, ease of integration with the backend, and the team's expertise.
   * The setup will include version control systems like Git and task runners or bundlers like Webpack to manage project files and dependencies.
3. Frontend component development:
   * Using HTML, CSS, and JavaScript, code the frontend components while integrating the chosen framework. Forms are used to enter data, tables are used to display NPA and litigation cases, and interactive elements are used to engage users.
   * Integration with APIs: Implement API calls to interact with the backend. We use the endpoints we've developed to fetch, display, and update data, such as NPA details or case status updates.
4. Management of the state:
   * Managing state is crucial for applications with complex data handling and multiple views. A central and reactive state management can be achieved by using libraries like Redux (for React) or Vuex (for Vue.js).
5. Designing a responsive website:
   * Improve accessibility and user experience by ensuring the frontend is responsive and works seamlessly across different devices and screen sizes.
6. Measures taken to ensure security:
   * Implement client-side security measures such as input validation to prevent XSS (Cross-Site Scripting) and CSRF (Cross-Site Request Forgery) attacks. Use tokens or other authentication methods to secure all data interactions with the API.
7. The testing process:
   * Test each component individually and when integrated using frameworks like Jest or Mocha.
   * Conduct user testing sessions to gather feedback on usability and identify potential interface issues.
8. The optimization and deployment process:
   * Optimize the application's performance for speed and efficiency. It is crucial to use techniques such as lazy loading, effective image management, and caching.
   * Deployment: Prepare the frontend for deployment, which may include setting up servers, configuring domains, and implementing continuous integration/continuous deployment (CI/CD) practices.
9. The documentation and training process consists of:
   * For future development and end-user training, thoroughly document the frontend codebase and user guides.
   * Conduct training sessions for staff of the AU Small Finance Bank to ensure they are comfortable with the new system.

In conclusion:

As a result of this detailed approach, LegalTrack NPA's frontend is not only aesthetically pleasing, but also functional, secure, and user-friendly. In addition to the backend capabilities, it provides a holistic solution for managing NPAs and litigation efficiently. Contributing significantly to the project's success, I oversee these processes, ensure integration fluidity, and maintain alignment with the project's goals.

**Overview of the LegalTrack NPA System's user interface**

LegalTrack NPA System is meticulously designed to ensure a user-friendly, efficient, and visually appealing experience for users. A detailed explanation of the system's functionality and design is provided, accompanied by screenshots demonstrating the practical application of these design elements.

1. Consistent navigation: sidebar design

1. A sidebar provides consistent navigational capabilities across the entire system.
2. The following features:
   1. It offers uninterrupted access to all major sections, fixed to the left.
   2. A user-friendly layout allows users to quickly navigate through the Dashboard, NPA Cases, Litigation Cases, Reports, and Settings.
   3. The main content area is rendered in a soothing teal, which contrasts effectively against the white background, making it easy to read and navigate.
3. User Interaction:
   1. Visual feedback is provided by hover effects on the links, enhancing the interactive experience.
4. **Dashboard Overview: Quick Statistical Insights**

The purpose is to provide users with a snapshot of critical metrics as soon as they log in.

Functions:

Displays key data points such as total NPAs, high-risk NPAs, and pending recoveries and litigation cases as widgets.Clicking on the widgets leads to more detailed data pages that can be explored in greater detail.

1. **Detailed Data Management: NPA and Litigation Cases Pages**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Detailed structure:

* These pages display extensive data using detailed tables, including NPA account details and litigation case details.
* The following columns should be included: Account Number, Customer Name, Loan Amount, Status, Lawyer Assigned, and Next Steps.

The interaction:

* Managing and navigating large datasets is made easier with features such as sorting and filtering.

1. **Comprehensive Reporting: Financial Overview and Trends**

**A screenshot of a computer

Description automatically generated**

In terms of layout:

The metrics table displays financial metrics necessary for fiscal analysis such as the total amount of NPAs, recoveries, and risk assessments.

Data Representation: Line charts provide a visual representation of quarterly NPA recovery trends, making it easier to absorb the information.

Capability to download:

A prominently located button allows users to download comprehensive reports for fiscal years for offline analysis and record-keeping.

5. Adaptability and Accessibility: Ensure Broad Usability

* Designing responsively:
  + A flexible layout ensures functionality across desktops, tablets, and smartphones.
* Features that make it accessible:
  + The use of high contrast and accessible typography improves the readability of the document for visually impaired users.

6. **Footer: Constant Information**

* Aspects of functionality:
  + Copyright and ownership information is visible on all pages, reinforcing the system's security and professionalism.

LegalTrack NPA System's user interface combines functional design with aesthetic appeal to enhance its users' efficiency and satisfaction. Its intuitive navigation, detailed data presentation, and comprehensive reporting ensure that the system not only meets but exceeds users' operational needs.

* + Copyright and ownership information is visible on all pages, reinforcing the system's security and professionalism.

LegalTrack NPA System's user interface combines functional design with aesthetic appeal to enhance its users' efficiency and satisfaction. Its intuitive navigation, detailed data presentation, and comprehensive reporting ensure that the system not only meets but exceeds users' operational needs.

I assisted and managed the development team in any technical aspect that was required during the development process. I learned a lot about the entire development process. I gained a deep understanding of the challenges and complexities of creating a comprehensive system for handling both non-performing assets (NPAs) and litigation cases while working on LegalTrack NPA for AU Small Finance Bank.

**Responsibilities and learnings:**

1. Assistance with technical issues and team management:
   * I was primarily responsible for providing technical support to the development team. Troubleshooting issues, reviewing code for optimization, and ensuring seamless integration of various components were all part of the process.
   * I also managed the development team, coordinating tasks, managing timelines, and ensuring that all team members were aligned with the project objectives.
2. Designing and architecting systems:
   * During the design sessions, I actively participated in discussions about the system architecture. As a result, I gained a firsthand understanding of how complex systems are structured, as well as the importance of designing scalable and robust architectures for them.
3. The Agile Development Process:
   * My team followed agile development practices, which was a significant learning curve for me. As part of daily stand-ups, sprint planning, and retrospectives, I gained insight into agile methodologies and their impact on productivity and team dynamics.

1. Security and compliance:
   * It was crucial that the LegalTrack NPA adhered to all legal and regulatory requirements. Monitoring the integration of security protocols into our development process was one of the things I learned about the various compliance measures we needed to implement.
2. Feedback from users:
   * For our iterative development approach, I was responsible for gathering feedback from preliminary users. Developing product enhancements based on user feedback was an invaluable aspect of my job.
3. Deployment and training:
   * Training and deployment were other areas where I contributed. In preparing training materials and conducting sessions for AU Small Finance Bank's staff, I learned about the challenges of software adoption and the importance of communication.
4. Learning and improving continuously:
   * My technical skills were enhanced and my understanding of financial and legal software solutions deepened throughout this project. There was always something to learn, whether it was a new programming technique or a deeper understanding of financial regulations.

**The conclusion of my internship at Cubic Tree Technology Solutions**

It has been an honor to be a part of such a wide range of projects during my internship at Cubic Tree Technology Solutions. My journey has significantly enriched my technical and analytical capabilities, providing me with an in-depth understanding of SaaS software development, client collaboration, and process optimization. Each project presented a unique set of challenges and opportunities that contributed to my professional development.

1. Analysis and development of Gaba Projects Private Limited solutions

CubiTask Pro, one of the first projects I worked on during my internship with Gaba Projects Private Limited (GPPL), was one of the first projects I worked on during my internship. Before using Cubic Tree's solutions, GPPL relied heavily on tools like Asana and Excel, which while functional, were not able to meet their growing project management and financial tracking needs.

The inefficiencies in their workflow were identified through extensive interviews with key stakeholders, including Manjeeri Desai and Mukta Bhardwaj, including manual data entry, lack of real-time financial tracking, and inability to scale project management for large, multi-departmental projects. I developed a more tailored solution that introduced automation, integrated real-time tracking, and provided more flexible reporting options by analyzing these challenges.

CubiTask Pro had a significant impact on the organization. Due to the elimination of manual data entry, automating project tracking and financial reporting resulted in a 35% increase in efficiency and a 90% reduction in errors. As team members were able to work in real-time within the system, report generation times were reduced by 50%, and collaboration across departments improved by 20%.

Through this project, I developed a solid understanding of project management systems and how software can be applied to solve real-world operational problems. In addition, it set the stage for my more complex future projects at Cubic Tree.

2. Development of SaaS solutions for Tata Steel and AU Small Finance Bank

As a result of CubiTask Pro's success, I was assigned to larger and more complex projects involving Tata Steel and AU Small Finance Bank.

ICMS, Tata Steel's eContract Management System

As part of Tata Steel's eContract Management System, I gained insight into the complexities of managing large-scale contracts in an enterprise setting. Vendor procurement was streamlined, compliance was monitored, and performance tracking was automated. Using Python, I helped implement modules that tracked vendor performance, contract lifecycle management, and regulatory compliance in the backend.

I gained a deeper understanding of database management and workflow automation through this project. By integrating AI-powered analytics, we were able to predict contract risks, evaluate vendor performance, and automate compliance checks. The project involved working with complex datasets, building scalable APIs, and ensuring data security, all while integrating the system with Tata Steel's existing enterprise resource planning system.

AU Small Finance Bank's LegalTrack NPA System

Due to its focus on managing Non-Performing Assets (NPAs) and litigation cases, AU Small Finance Bank's LegalTrack NPA platform presented unique challenges. Unlike Tata Steel's system, which focused on contracts, LegalTrack's NPA platform streamlines legal workflows, manages high-risk assets, and optimizes recovery.

As part of this project, I structured the data models for managing NPAs and litigation cases, ensuring that the backend could handle large volumes of data related to legal cases, financial risk, and recovery efforts efficiently. My role also included creating APIs that connect the bank's backend and frontend systems, enabling real-time updates and seamless integration.

As a result of this project, I gained experience in the legal aspects of financial management, such as compliance with banking regulations, risk mitigation, and litigation case tracking. The LegalTrack NPA system improved my ability to handle data-intensive applications, manage complex legal processes, and automate decision-making.

3. Developing my technical skills

My internship provided me with the opportunity to work across different areas of software development. As a result of each project, I gained a deeper understanding of key technical areas:

* The Python backend language allowed me to build robust and scalable systems capable of handling complex business logic and data transactions. As a result of my experience, I learned how to manage data securely, handle large amounts of concurrent requests, and optimize backend performance for enterprise-level applications.
* Both LegalTrack NPA and ICMS require API development. The REST APIs I built ensured seamless communication between the frontend and backend, real-time data retrieval, and secure data exchange between different services.
* I gained hands-on experience designing intuitive and functional user interfaces, despite my primary focus being backend development. In the case of LegalTrack NPA, where users manage high-stakes litigation and asset recovery processes, ease of use and real-time access to data were crucial.

4. Identifying client requirements and tailoring solutions to them

For each project I worked on, I needed a deep understanding of the client's unique operational requirements. Whether it was improving contract workflows for Tata Steel, managing litigation and NPAs for AU Small Finance Bank, or helping GPPL streamline their project management, I learned how to translate complex business requirements into functional software.

Working directly with stakeholders across different industries has given me a holistic perspective on how technology can improve operational efficiency, reduce manual workloads, and provide real-time data insights.

5. The importance of teamwork and collaboration

My internship provided me with the opportunity to work with talented developers, project managers, and client representatives. In this project, I learned how to manage a development project from start to finish using agile methodologies, team coordination, and project management tools. Continual feedback loops, daily standups, and sprint planning allowed me to improve our systems incrementally.

**Reflections on my internship**

Cubic Tree Technology Solutions gave me a lot of opportunities, and I am extremely grateful for them. As a result of working with high-profile clients like Tata Steel and AU Small Finance Bank, tackling complex technical challenges, and collaborating with a skilled team, I have not only improved my technical skills, but gained invaluable insight into SaaS product usage in the real world.

As a result of this experience, I have strengthened my foundation in software development, project management, and client collaboration. In the end, I am confident that I am capable of handling complex software projects, developing scalable solutions, and contributing meaningfully to future endeavors.

I continue to be thankful for the mentorship, learning opportunities, and real-world challenges I encountered during my time at Cubic Tree Technology Solutions as I move forward in my career.

**Newberry Group internship (15th April to 25th June 2024)**

From 15th April to 25th June 2024, I interned at Newberry Group, an esteemed investment boutique headquartered in Mumbai. In my role as a Quantitative Analyst, I had the unique opportunity to gain a deeper understanding of algorithmic trading and portfolio management in the dynamic Indian capital markets.

**Newberry Group: a brief overview**

A leading player in the Indian investment and financial sector, Newberry Group is registered with SEBI as a Portfolio Management Service (PMS) provider and Category-I Merchant Banker. With an emphasis on delivering expert guidance on corporate structuring and capital market strategies, the firm provides corporate advisory, structured financing, and resource-raising solutions to companies.

Several prominent financial institutions, including the National Stock Exchange (NSE), Bombay Stock Exchange (BSE), NSE F&O (Futures & Options), and Central Depository Services (India) Limited (CDSL), bolster the firm's credibility and reach. Consequently, Newberry Group is a trusted advisor for several high-profile business entities in India due to its comprehensive market access and expertise.

**Quantitative Analyst's Role**

In my role, I primarily focused on quantitative analysis and gained first-hand experience implementing and monitoring algorithmic trading strategies. It applies both traditional and innovative trading strategies to enhance portfolio performance in the futures and options markets, especially in the Nifty and Bank Nifty indices.

Learning and Responsibilities:

1. Trading algorithm:
   * Algorithmic trading is utilized by Newberry Group to run automated trading strategies that execute orders based on pre-programmed parameters in real-time. Developing and monitoring algo trading systems was my main responsibility.
   * The backend code for these systems was implemented and ensured their efficient operation. To test and validate the system's performance in live market conditions, I worked closely with senior quantitative analysts and developers.
2. Strategies for trading:
   * A number of trading strategies are employed by the company to trade futures and options, with a particular focus on:
     + Fast Moving Average (FMA) and Simple Moving Average (SMA) are foundational strategies for tracking short-term and long-term price trends.
     + With exponential moving averages (EMAs), traders are able to capture trends and reversals faster than with simple moving averages (SMAs).
   * Additionally, I learned about newer strategies such as:
     + Identifying potential market directions based on the price range set in the early hours of the market.
     + It involves forming smaller price ranges within previous candles' highs and lows during price consolidation periods.
     + A more complex trading strategy that takes into account specific price action movements during the day.
   * In high-frequency trading environments, I gained a deep understanding of technical indicators by learning and applying these strategies.
3. Deploying and monitoring algorithms:
   * The algorithmic systems exposed me to how trading algorithms are developed, tested, and deployed. By assisting with the coding and debugging of trading strategies within the system, I contributed to the deployment process.
   * I was also responsible for monitoring the performance of these systems. Tracking market fluctuations, ensuring timely execution of trades, and improving profitability were some of my responsibilities.
4. Experience in collaboration:
   * Working closely with the development and quantitative teams, I identified and fixed potential algorithmic trading system issues. Furthermore, I attended regular strategy meetings to analyze the performance of various trading strategies and discuss ways to refine and improve them.
   * As a result of this hands-on experience, I gained a strong understanding of risk management and the importance of data integrity in financial models.

A screenshot of a phone

Description automatically generated

The above screenshots depict a variety of aspects of algorithmic trading and trading platform user interfaces. The following is an explanation of what each image shows:

1. **Interface of the Tradetron App:**
   * The Tradetron algorithmic strategy marketplace interface is shown in this image. Creating and deploying trading strategies is made simple thanks to the platform's user-friendly nature, which supports code-less strategy creation. There is also a sign-in and sign-up option, indicating that the service requires user registration.

A diagram of a trading process

Description automatically generated

**Components of algorithmic trading:**

* A diagram of the basic components involved in setting up and executing algorithmic trading is shown below. Several factors are shown, such as Entry Price, Exit Price, Stop Loss, Target Price, Quantity, and Order Type, which are essential in defining the parameters of an automated trading strategy. An algorithmic trading system's layout simplifies understanding how these elements interact.

A diagram of a algorithm

Description automatically generated

**Algorithmic Trading Flowchart from Investopedia:**

* According to Investopedia, the algorithmic trading process is broken down into three fundamental components: the Programmer, Code, and Dynamic Assessment. By automating these processes, algorithmic trading saves traders time. The visual illustrates the workflow from development to deployment, demonstrating the efficiency and automation that algorithmic trading provides.

**What I learned from my internship**

I gained a deeper understanding of algorithmic trading and quantitative analysis through this internship. I gained practical technical experience as well as a better understanding of how to apply theoretical financial models to real-life situations. I gained valuable insight into the following areas:

1. Expertise in technical fields:
   * Working on real-time trading infrastructure sharpened my skills in backend coding. Monitoring system performance and deploying algorithmic trading strategies became my specialties.
2. Expertise in algorithmic trading:
   * Moving averages, exponential moving averages, and advanced concepts such as Open Range and Inside Candle helped me gain a deeper understanding of how financial markets work and how to take advantage of market volatility.
3. Knowledge of the market:
   * I learned how to anticipate how changes in market dynamics could affect the performance of algo strategies by observing how market conditions affected the performance of algo strategies.
4. Solving problems and managing risks:
   * There were numerous system-related issues, such as delays in trade execution or data discrepancies, that required real-time troubleshooting. I gained a comprehensive understanding of continuous monitoring through these experiences and learned how to manage risks associated with algorithmic trading.

**By working on two of the company's most effective trading strategies**, I honed my backend coding skills. The Simple Moving Average Crossover (SMA Crossover) and the Inside Candle strategies have been instrumental in managing a portfolio that has generated substantial returns.

* + SMA Crossover Strategy: This strategy involves calculating and monitoring short-term and long-term moving averages of stock prices in order to identify potential buy and sell signals. When the short-term moving average crosses above or below the long-term moving average, bullish or bearish trends are indicated.
  + A candle that is fully encompassed by the previous candle's range is considered to be a consolidation candle, suggesting a potential breakout. According to their position and frequency, inside candles are useful for predicting short-term directional movements in the market.
  + In order to generate actionable trading signals, these strategies used Python for data fetching, processing, and visualization. The experiences I gained not only enhanced my programming skills, but also deepened my understanding of financial markets and algorithmic trading.
  + Below are screenshots illustrating the use of these strategies with historical data, illustrating the trading signals and trends identified through my code implementations. Portfolio management decisions were significantly influenced by these visual tools that validated the effectiveness of each strategy and provided a clear picture of market behavior.

A screenshot of a computer program

Description automatically generated

A graph with lines and numbers

Description automatically generated with medium confidence

A computer code with text

Description automatically generated with medium confidence

A graph showing a line graph

Description automatically generated with medium confidence

A diagram of a algorithm

Description automatically generatedReflections : This experience taught me the importance of real-time problem solving and risk management in algorithmic trading. I also learned the importance of developing a comprehensive understanding of continuous monitoring and troubleshooting.

I am deeply grateful for the opportunities and experiences I gained as a Quantitative Analyst at Newberry Group. My exposure to investment banking, algorithmic trading, and quantitative strategies has greatly enhanced my understanding of the financial markets. My technical and financial knowledge was put to use in a real-world setting by working on a live trading system.

Furthermore, I am grateful for the mentorship and guidance provided by the Newberry Group team. I was encouraged to take on new challenges and was supported in learning the nuances of the Indian capital markets and cutting-edge trading techniques.

In my professional career, I hope to use the knowledge and experiences I gained during my internship at Newberry Group to my advantage.

**References**

1. Investopedia. (2023). *Algorithmic Trading*. Retrieved from https://www.investopedia.com/terms/a/algorithmictrading.asp
2. Heroku. (n.d.). *Getting Started on Heroku with Python*. Retrieved from https://devcenter.heroku.com/articles/getting-started-with-python
3. Flask. (2023). *Flask Documentation*. Retrieved from https://flask.palletsprojects.com/en/2.2.x/
4. AWS. (2023). *Deploying Flask Applications on AWS*. Retrieved from <https://aws.amazon.com/getting-started/hands-on/deploy-python-flask-application/>
5. Google Cloud. (2023). *Deploy Flask Applications on Google App Engine*. Retrieved from https://cloud.google.com/appengine/docs/standard/python3/building-app
6. QuantInsti. (2023). *Price Action Trading and Algorithmic Strategies*. Retrieved from https://blog.quantinsti.com/price-action-trading-algo-trading-python/
7. GeeksforGeeks. (2023). *API in Flask*. Retrieved from https://www.geeksforgeeks.org/creating-first-restful-api-using-python-flask/
8. Real Python. (2023). *Introduction to Flask*. Retrieved from https://realpython.com/tutorials/flask/
9. Towards Data Science. (2023). *Creating RESTful APIs with Flask*. Retrieved from https://towardsdatascience.com/restful-api-with-python-flask-and-mysql-d1a759a50a6c
10. SQLAlchemy. (2023). *SQLAlchemy Documentation*. Retrieved from https://docs.sqlalchemy.org/en/14/
11. Medium. (2023). *How to Build a Flask API and Deploy It on Heroku*. Retrieved from https://medium.com/codex/how-to-build-and-deploy-your-flask-api-on-heroku-3bf24a68f54e
12. IBM Cloud. (2023). *Deploying Python Applications on IBM Cloud*. Retrieved from https://www.ibm.com/cloud/blog/deploying-python-flask-app-on-ibm-cloud
13. Hackernoon. (2023). *A Guide to Flask APIs and Heroku Deployment*. Retrieved from https://hackernoon.com/guide-to-flask-and-heroku-deployment
14. Stack Overflow. (2023). *Common Issues in Flask APIs*. Retrieved from <https://stackoverflow.com/questions/tagged/flask>
15. DigitalOcean. (2023). *Deploying Flask Applications on DigitalOcean*. Retrieved from https://www.digitalocean.com/community/tutorials/how-to-deploy-a-flask-app-on-digitalocean
16. Django Documentation. (2023). *Differences Between Flask and Django*. Retrieved from <https://docs.djangoproject.com/en/stable/>
17. Python.org. (2023). *Python Official Documentation*. Retrieved from <https://docs.python.org/3/>
18. W3Schools. (2023). *API Tutorial for Beginners*. Retrieved from https://www.w3schools.com/python/python\_api.asp
19. Coding the Smart Way. (2023). *Build and Deploy Flask APIs*. Retrieved from <https://codingthesmartway.com/building-flask-api/>
20. Gartner. (2023). Software Reviews and Comparisons. Retrieved from https://www.gartner.com
21. Capterra. (2023). Compare SaaS and Legal Tech Software. Retrieved from https://www.capterra.com
22. G2 Crowd. (2023). Software Reviews & Insights. Retrieved from https://www.g2.com
23. TrustRadius. (2023). Software Comparison and User Reviews. Retrieved from https://www.trustradius.com
24. Software Advice. (2023). Business Software Reviews and Recommendations. Retrieved from https://www.softwareadvice.com
25. TechRepublic. (2023). SaaS and IT Software Reviews. Retrieved from https://www.techrepublic.com
26. LegalTech News. (2023). Legal Technology News and Analysis. Retrieved from https://www.law.com/legaltechnews/
27. PCMag. (2023). Business Software Buying Guides and Reviews. Retrieved from https://www.pcmag.com
28. Forrester Research. (2023). Market Research and Vendor Analysis. Retrieved from https://www.forrester.com
29. Crunchbase. (2023). Company Profiles and Industry Trends. Retrieved from https://www.crunchbase.com
30. Udemy. (2023). *Flask Web Development Tutorials*. Retrieved from https://www.udemy.com/course/flask-web-development/

**Footnote**: Some parts of this document, including information on algorithmic trading strategies and the development of API integration, were assisted by AI, specifically **ChatGPT** by OpenAI, for information gathering and paraphrasing.