

**AWS IDP-Invoice Extraction with GenAI**

## A PROJECT REPORT

***submitted by***

**Barath Kumar J** **(CB.EN.U4CSE21608)**

**Rohith M (CB.EN.U4CSE21048)**

**Harish T.S. (CB.EN.U4CSE21022)**

**Guhanesh T (CB.EN.U4CSE22015)**

**Ajayraj M (CB.EN.U4CSE21004)**

**Adhvaith Sankar (CB.EN.U4CSE21003)**

**Department Computer Science and Engineering**

**Amrita Vishwa Vidyapeetham,**

**Coimbatore-641112**

***in collaboration with***

**SAP Labs, Bengaluru**

## April 2024

**BONAFIDE CERTIFICATE**

Certified that the mini-project titled **“AWS IDP-Invoice Extraction with GenAi”** is the bonafide work of “**Barath Kumar J (CB.EN.U4CSE21608), Rohith M (CB.EN.U4CSE21048), Harish T.S. (CB.EN.U4CSE21022), Guhanesh (CB.EN.U4CSE22015), Ajayraj M (CB.EN.U4CSE21004), and Adhvaith Sankar (CB.EN.U4CSE21003).”** which is carried out under our mentoring.

|  |  |
| --- | --- |
| **------------------------- SIGNATURE**  **INDUSTRY MENTOR**  **Siddiqui Faiz**  Spend Engineering, SAP Labs,  Bengaluru, Karnataka, India | **------------------------- SIGNATURE**  **ACADAMIC MENTOR**  **Dr. Sabarish B. A.**  Asst. Professor,  Department of CSE,  School of Computing,  Amrita Vishwa Vidyapeetham,  Coimbatore-641112 |

|  |  |
| --- | --- |
| **------------------------- SIGNATURE**  Srivatsan Santhanam  **VICE PRESIDENT**  Spend Engineering, Head of Concur R&D, SAP Labs,  Bengaluru, Karnataka, India | **------------------------- SIGNATURE**  **Dr. Vidhya Balasubramanian**  **HEAD OF DEPARTMENT**  Professor,  Department of CSE,  School of Computing,  Amrita Vishwa Vidyapeetham,  Coimbatore-641112 |

## Submitted for evaluation on

**ACKNOWLEDGEMENT**

We would like to express our deepest gratitude to our beloved Principal,

**Dr. Vidhya Balasubramanian** for her overwhelming support and encouragement on this project.

We extent our indebted to **Mr. SRIVATSAN SANTHANAM,** Vice President, Spend Engineering, Head of Concur R&D, SAP Labs India for the opportunity and support which was instrumental in the completion of this project.

We are greatly indebted to **Dr. Vidhya Balasubramanian** Head of the Department, Computer Science and Engineering for her guidance and continuous support which was instrumental in the completion of this project.

We extend our thanks to our industry mentor, Siddiqui Faiz, SAP Labs, Bengaluru, India and academic mentor Dr. Sabarish B.A, Computer Science and Engineering for his technical support and constant supervision without which we could not have completed this project study.

**Barath Kumar J**

**Harish T.S**

**Guhanesh T**

**Ajayraj M**

**Rohith M**

**Adhvaith Sankar S.S.**

**ABSTRACT**

This project aims to develop a robust and efficient system for automating the extraction of invoice details and seamlessly populating them into relevant forms. Traditional manual methods of data entry for invoices are time-consuming and error-prone, often leading to inefficiencies and inaccuracies. Our system will streamline this process by automatically extracting key information from invoices and intelligently auto filling corresponding form fields.

In this project, we present two distinct approaches for automating the extraction of invoice details and populating them into forms. In the first approach, we utilize a model known as Meta-7b, which is designed to efficiently extract structured data from documents. This model returns the extracted data in JSON format, providing a structured output that can be easily integrated into downstream processes. By leveraging Meta-7b, we achieve robust extraction of invoice details, including crucial information such as invoice numbers, dates, vendor details, and line items.

In the second approach, we employ the Haystack framework along with the Gemini Pro API key to retrieve invoice data. Haystack is a powerful tool for building end-to-end question answering systems and information retrieval pipelines. By fine-tuning Haystack with the Gemini Pro API key and incorporating a Pydantic validator, we enhance the system's ability to accurately extract and validate invoice details. This approach offers flexibility and scalability, allowing for seamless integration with existing workflows and applications.

**Keywords:** Haystack, Pydantic, Gemini Pro, Meta-7b, Json

## TABLE OF CONTENTS

CHAPTER NO TITLE PAGE NO ABSTRACT iii

LIST OF TABLES v

LIST OF FIGURES vi

[LIST OF ABBREVIATIONS](#_TOC_250010) vii

1. INTRODUCTION
   1. [PROJECT MOTIVATION 1](#_TOC_250009)
   2. [PROBLEM STATEMENT AND OBJECTIVES](#_TOC_250008) 2
   3. [SCOPE AND LIMITATIONS OF THE PROJECT](#_TOC_250007) 3
2. PROJECT ARCHITECTURE, DESIGN AND IMPLEMENTATION
   1. [SYSTEM ARCHITECTURE](#_TOC_250006) 4
   2. [OVERVIEW OF THE DESIGN PROCESS](#_TOC_250005) 6
   3. [EXPLANATION OF THE ENGINEERING PRINCIPLES USED IN THE DESIGN 8](#_TOC_250004)
   4. [DESCRIPTION OF THE STEPS TAKEN TO IMPLEMENT THE PROJECT DESIGN](#_TOC_250003) 11
3. RESULTS AND ANALYSIS
   1. [VALIDATION PROCEDURES](#_TOC_250002) 15
   2. [TEST RESULTS](#_TOC_250001) 16
   3. [ANALYSIS OF RESULTS](#_TOC_250000) 19
4. LEARNING OUTCOMES 20
5. CONCLUSION AND RECOMMENDATIONS 21

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **LIST OF TABLES**  **TITLE** | **PAGENO** |
| 2.1 | Configurations for different cloud services | 8 |

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **LIST OF FIGURES**  **TITLE** | **PAGENO** |
| 2.1 | Haystack Pipeline | 4 |
| 2.2 | System Architecture | 5 |
| 2.3 | JSON former | 13 |
| 3.1 | Login Page | 16 |
| 3.2 | User Dashboard | 17 |
| 3.3 | User Profile | 17 |
| 3.4 | Dashboard for finance | 18 |
|  |  |  |
|  |  |  |

## LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| API | Application Programming Interface |
| CSS | Cascading Style Sheets |
| GPU | Graphical Processing Unit |
| JSON | JavaScript Object Notation |
| LLM | Large Language Model |
| ML | Machine Learning |
| REST | Representational State Transfer |
|  |  |

## CHAPTER 1 INTRODUCTION

## PROJECT MOTIVATION

The motivation behind our invoice extraction and auto filling application stems from the need to automating the manual processes, enhance accuracy, and improve efficiency in handling invoices. Below, we delve into the key drivers for this project:

**Cost Reduction and Resource Optimization:**

Manual data extraction from invoices is time-consuming and often error-prone.

By automating the extraction process, organizations can reduce costs associated with hiring manpower for repetitive tasks. Employees can then focus on more productive and value-added activities rather than mundane data entry.

**Error Minimization and Data Accuracy:**

Extracting information from invoices is challenging due to varying formats and layouts. Human errors during manual extraction can lead to data loss and inaccuracies. With the help of our app, we can get accurate results. By automating the process, we enhance the overall quality of extracted information.

**Ready Availability and Security:**

Once invoices are fed into the system, LLM extracts the text without requiring human intervention. This ready availability of data streamlines workflows and eliminates delays. Additionally, a fully automated extraction process provides data-level security to the organization. Sensitive information remains within the system and is not easily visible to external parties. In summary, our project aims to bridge the gap between manual invoice handling and efficient automation, benefiting organizations across industries. By leveraging deep learning techniques, we enhance accuracy, reduce costs, and improve overall productivity.

## PROBLEM STATEMENT AND OBJECTIVES

## Problem Statement:

## Expense creation within the application currently demands users to manually input details alongside uploading invoices. This includes information such as invoice number, date, amount, and vendor details. Despite this data being available within the uploaded invoice, users are burdened with redundant data entry. In short, the manual input process for extracting invoice details lacks efficiency and is prone to errors due to redundant data entry tasks.

## Objectives:

## Develop an automated system capable of extracting key invoice details (invoice number, date, amount, vendor details) directly from uploaded invoices.

## Ensure high accuracy and reliability in the extraction process to minimize errors and discrepancies in expense creation.

## Enhance user experience by reducing the need for manual data entry tasks and simplifying the expense creation process.

## Improve overall efficiency and productivity for users by eliminating redundant data entry tasks and optimizing time spent on expense management.

## Provide comprehensive support and guidance to users for effectively utilizing the automated invoice data extraction feature.

## Ensure compliance with data privacy and security regulations throughout the invoice extraction and expense creation process.

## Gather user feedback to assess satisfaction levels and identify areas for further improvement in the automated invoice data extraction feature.

## SCOPE AND LIMITATIONS OF THE PROJECT

The scope of the invoice extraction and auto filling project is multifaceted, encompassing various aspects related to efficient invoice handling and automation. Let’s explore the project’s scope in detail:

**Automated Data Extraction from Invoices:**

The primary objective is to automate the extraction of relevant information from invoices.

This includes extracting data such as vendor details, invoice numbers, dates, line items, and total amounts.

The scope extends to handling various invoice formats, including scanned, digital, and handwritten invoices.

**Integration with Form Filling:**

Once the relevant data is extracted, the project aims to populate predefined forms or templates.

Forms can be in different formats, such as PDFs, spreadsheets, or web-based interfaces.

The system dynamically fills in the appropriate fields based on the extracted information.

**Benefits and Business Impact:**

* Cost Reduction: By automating data extraction, organizations can reduce costs associated with manual data entry.
* Error Minimization: Automation helps minimize human errors, ensuring accurate and consistent data.
* Ready Availability: Extracted data is readily available for further processing without delays.
* Security: A fully automated process provides data-level security, preventing unauthorized access.

**Future Enhancements:**

* Machine Learning: Investigate ML models for better extraction accuracy, especially for unstructured invoices.
* Semantic Understanding: Move beyond simple text extraction to understand invoice semantics (e.g., identifying line items, due dates, and payment terms).
* Scalability: Ensure the system can handle a large volume of invoices efficient

## CHAPTER 2

**PROJECT ARCHITECTURE, DESIGN AND IMPLEMENTATION**

## SYSTEM ARCHITECTURE

PREPROCESSING ARCHITECTURE:

* To preprocess the input data (pdf/png) a custom workflow was used.
* The pdf pages were converted to image fragments.
* Text is extracted from image inputs, or the image fragments developed from the pdf using pyTesseract.

LLM ARCHITECTURE:

Haystack pipeline diagram

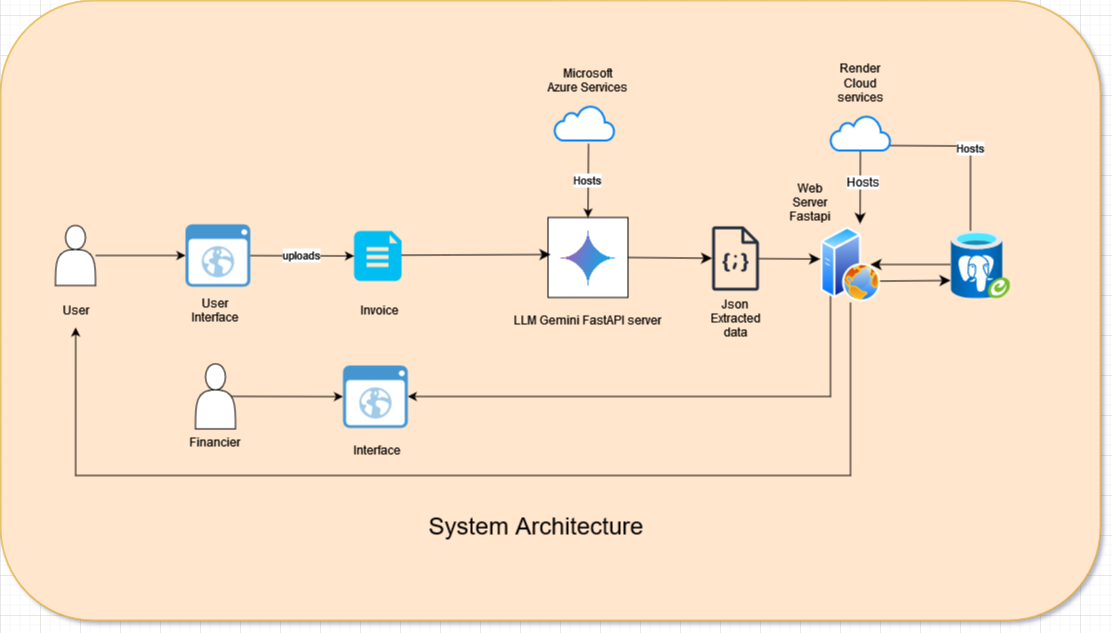
A diagram of a computer program

Description automatically generated

**Fig 2.1 Haystack pipeline**

POSTPROCESSING ARCHITECTURE:

* The results from the pipeline in the format List {Union [Str, Dict[str,str]]} was converted to proper JSON outputs for easier use.



**Figure 2.2 System Architecture**

## OVERVIEW OF THE DESIGN PROCESS

## Requirements Gathering and Analysis:

## Understand the specific needs of your target users and stakeholders.

## Gather detailed requirements related to invoice handling, data extraction, and form filling.

## Identify the key features the website must offer, such as user authentication, file upload, and data processing.

## User Experience (UX) Design:

## Create wireframes and mockups to visualize the website’s layout and flow.

## Focus on intuitive navigation, clear call-to-action buttons, and user-friendly interfaces.

## Consider responsive design to ensure the website works well on various devices (desktop, tablet, mobile).

## Backend Development:

## Set up the server-side infrastructure for the website.

## Choose a suitable backend technology stack (e.g., FastAPI, Python)

## Implement APIs for file upload, data extraction, and form filling.

## Integrate with external services (e.g., OCR APIs, form libraries).

## Frontend Development:

## Develop the user interface (UI) using HTML, CSS, and JavaScript React.

## Create pages for uploading invoices, displaying extracted data, and form filling.

## Implement AJAX or RESTful APIs to communicate with the backend.

## Invoice Extraction and Data Processing:

## Integrate an LLM service (such as Gemini) to extract data from uploaded invoices.

## Process the extracted data (e.g., invoice number, vendor details) and store it in a database.

## Form Filling and Automation:

## Design form templates for various types of invoices (e.g., purchase orders, expense reports).

## Dynamically populate form fields with the extracted data.

## Implement logic to handle different form structures and variations.

## Security and Authentication:

## Implement user authentication and authorization mechanisms.

## Restrict access to sensitive features (e.g., admin-only form approval).

## EXPLANATION OF THE ENGINEERING PRINCIPLES USED IN THE DESIGN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine type** | **Virtual CPUs** | **Memory** | **Price (USD)** | **Preemptible price**  **(USD)** |
| B2s-Standard -64 | 2 | 4 GB | $0.065 | $50/month |
| Render Free Instance(web service) | 0.1 | 256Mb | Free | Free |
| Render PSQL Instance(Login) | 0.1 | 256Mb | Free | Free |
| Render PSQL Instance(Data) | 0.1 | 256Mb | Free | Free |

**Table 2.1 Configurations for different cloud services**

**Modularity and Scalability**: Our design adheres to the principle of modularity, allowing different components to be easily integrated or replaced as needed. This modularity ensures flexibility in adapting to varying requirements and future scalability, enabling our solution to grow with the evolving needs of our stakeholders.

We have 3 modules for backend that is being ran on cloud:

* One being the FastAPI python server that handles all the get and post operations between the frontend and database services that is run in render web service cloud.
* Second being the python FastAPI server for Gemini API LLM that is run in microsoft azure as it requires higher ram requirements whereas the above doesn’t.
* Third being the PostgreSQL Database service that is run on render handles only read and write from the first server that requests.

About the frontend we have used Reactjs that is being run on vite+reactjs instead of webpack/Babel so in order to reduce the build time hopefully. Frontend has been hosted in a separate service for multiple clients to access which makes it easier and faster for end users for it to render.As a whole we have used a microservices architecture where each module is separate and runs in different instances.

**Structural Integrity and Material Selection**: The structural integrity of our design is paramount to its overall performance and longevity. We meticulously analyzed the mechanical stresses and environmental factors to select appropriate materials with the requisite strength, durability, and resistance to corrosion. By prioritizing structural integrity, we ensure the reliability and safety of our solution under diverse operating conditions.As mentioned due to the microarchitecture we have used different ram specifications and CPUs’ for each service as mentioned above:

**Backend :**

* Gemini LLM service is running on Microsoft azure 4GB 2 CPU specification for its heavy usage.
* Python Fast API server for get,post services between frontend and backed uses 0.1CPU and 256mb RAM since , Fast API is being used it has proved to be faster with the available resources.
* Database is being hosted on same specifications on another instance which was found to be enough for the load operations.

**Safety and Reliability**: Safety considerations are paramount in our design, particularly in applications where human lives or critical assets are at stake. We conducted rigorous hazard analyses and implemented redundant safety mechanisms to mitigate risks and ensure fail-safe operation. By prioritizing safety and reliability, we instill confidence in the end-users and stakeholders regarding the dependability of our solution. For all the assets and information’s not being leaked and to keep it secure even though if one module is compromised we made sure location transparency between modules so others won’t be under risk too.

* First step being the details of employee and login details are being hosted in two different server instances to maintain security and if attacker compromises employee details he won’t be able to reach log in password and other details required to access the software and vice versa.
* Second being the same for invoice details that is being uploaded too. We store it in a separate database again to achieve location transparency.
* Third being the modularity and direct connection between modules are avoided for chained attacks.
* Fourth being data transparency and how and where data is being transferred and changed all are hidden or abstracted from the view of the user for clean and safe usage.
* Fifth being adding users and generating passwords follow a very strict process of leaving it to the admin to add that to the data which a private endpoint has been created by which if attack or compromise happens it is easier to pinpoint who could have done it.

**Usability and Human-Centered Design**: A user-centric approach guided our design process, emphasizing intuitive usability and ergonomic considerations. By incorporating feedback from end-users and conducting usability tests, we optimized the user experience and minimized the learning curve associated with our solution.As it will be shown in the upcoming sections the UI has been made very abstract and easy to follow/ Through human-centered design principles, we enhance user satisfaction and productivity, ultimately driving the adoption and success of our product in the market.

## DESCRIPTION OF THE STEPS TAKEN TO IMPLEMENT THE PROJECT DESIGN

**Approach 1: Utilizing Meta-7b Model**

* **Data Collection and Annotation**:
  + Gather a diverse set of sample invoices representing various formats, layouts, and languages.
  + Annotate the invoices to create labeled datasets, marking key information such as invoice numbers,dates, vendor details, and line items.
* **Preprocessing:**
  + Preprocess the collected data to standardize formats, remove noise, and ensure consistency.
  + Convert the invoices into a format suitable for input to the Meta-7b model.
* **Model Selection and Integration**:
  + Select the Meta-7b model for its ability to efficiently extract structured data from documents.
  + Integrate the model into the system, configuring it to accept invoice documents as input and return structured data in JSON format.
* **Training and Fine-Tuning:**
  + Fine-tune the Meta-7b model using the annotated dataset to improve its performance on invoice extraction tasks.
  + Validate the model's performance using cross-validation techniques and adjust parameters as necessary.
* **Integration with Form Population:**
  + Develop a module to parse the JSON output from the Meta-7b model and populate the relevant form fields with extracted invoice details.
  + Implement error handling mechanisms to handle cases where certain details may not be extracted accurately.
* **Testing and Validation:**
  + Conduct thorough testing of the system using a variety of invoice samples to ensure robustness and accuracy.
  + Validate the populated form fields against ground truth data to verify the correctness of extraction.
* **Deployment:**
  + Deploy the system in a production environment, ensuring scalability, availability, and security.
  + Monitor the system's performance post-deployment and address any issues that arise.

**Figure 2.3 JSONformer**

A screenshot of a diagram

Description automatically generated

**Approach 2: Employing Haystack Framework with Gemini Pro API Key**

* **Data Collection and Preprocessing:**
  + Follow similar steps as in Approach 1 to collect and preprocess the invoice data.
* **Integration of Haystack Framework:**
  + Integrate the Haystack framework into the system, configuring it to serve as the information retrieval/pipeline.
  + Utilize the Gemini Pro API key for enhanced document understanding capabilities within Haystack.
* **Fine-Tuning and Validation:**
  + Fine-tune the Haystack framework with the Gemini Pro API key using the annotated dataset to improve its performance on invoice extraction.
  + Validate the model's performance and adjust parameters as necessary.
* **Integration with Pydantic Validator:**
  + Incorporate a Pydantic validator to validate the extracted invoice details for accuracy and completeness.
  + Develop mechanisms to handle validation errors and provide feedback to users.
* **Testing and Deployment:**
  + Follow similar testing and deployment procedures as described in Approach 1.

## CHAPTER 3 RESULTS AND ANALYSIS

## VALIDATION PROCEDURES

Sample Invoice Inputs:

* The sample invoices included single-page invoices, multi-page invoices and invoices with tables

Ground Truth Annotation:

* Each sample invoice was manually annotated with the correct extraction results for key fields such as invoice number, date, vendor name, total amount, etc.
* The ground truth annotations served as a reference for evaluating the accuracy of the LLM model's predictions.

Execution of Validation Tests:

* The sample invoices were input into the LLM model integrated with Haystack and the Gemini API generator.
* The model extracted the relevant details, and the results were compared against the ground truth annotations.

Analysis of Results:

* Performance Evaluation: The extraction speed and resource consumption of the LLM model were assessed.

## TEST RESULTS

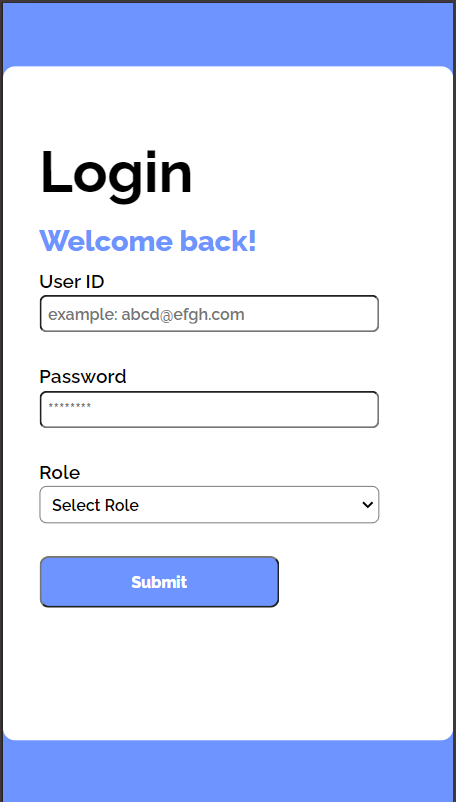
## LLM MODEL:

* APPROACH TWO:

Average Response Latency: 7 seconds

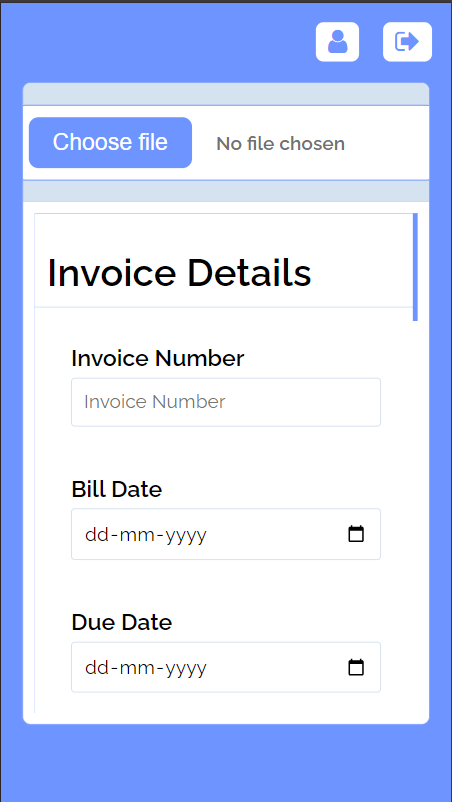
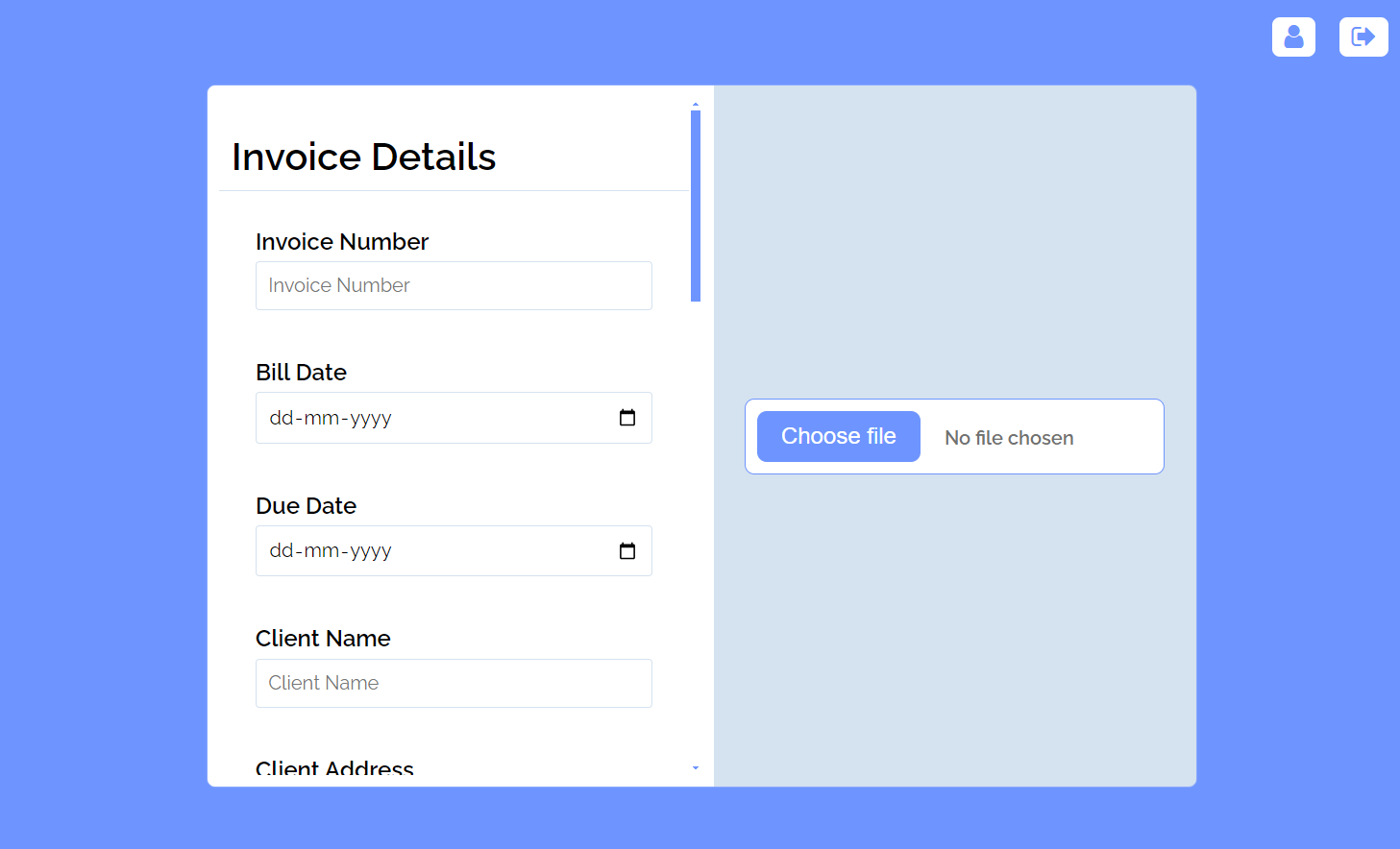
* APPROACH ONE:(discarded approach)

Average Response Latency: 23 seconds

****

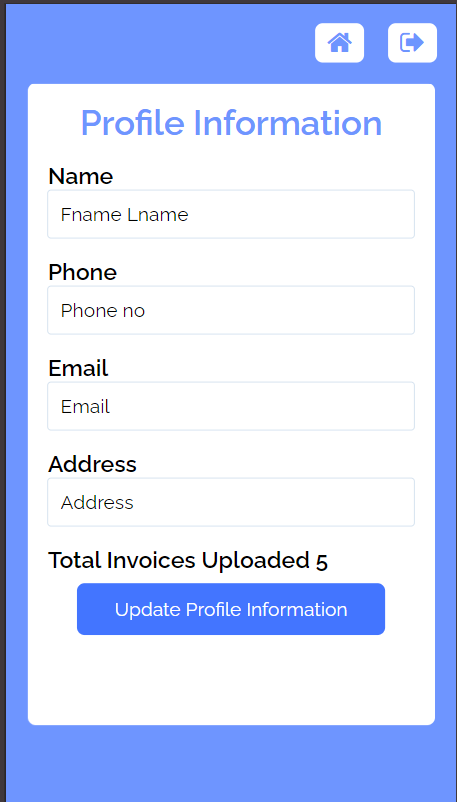
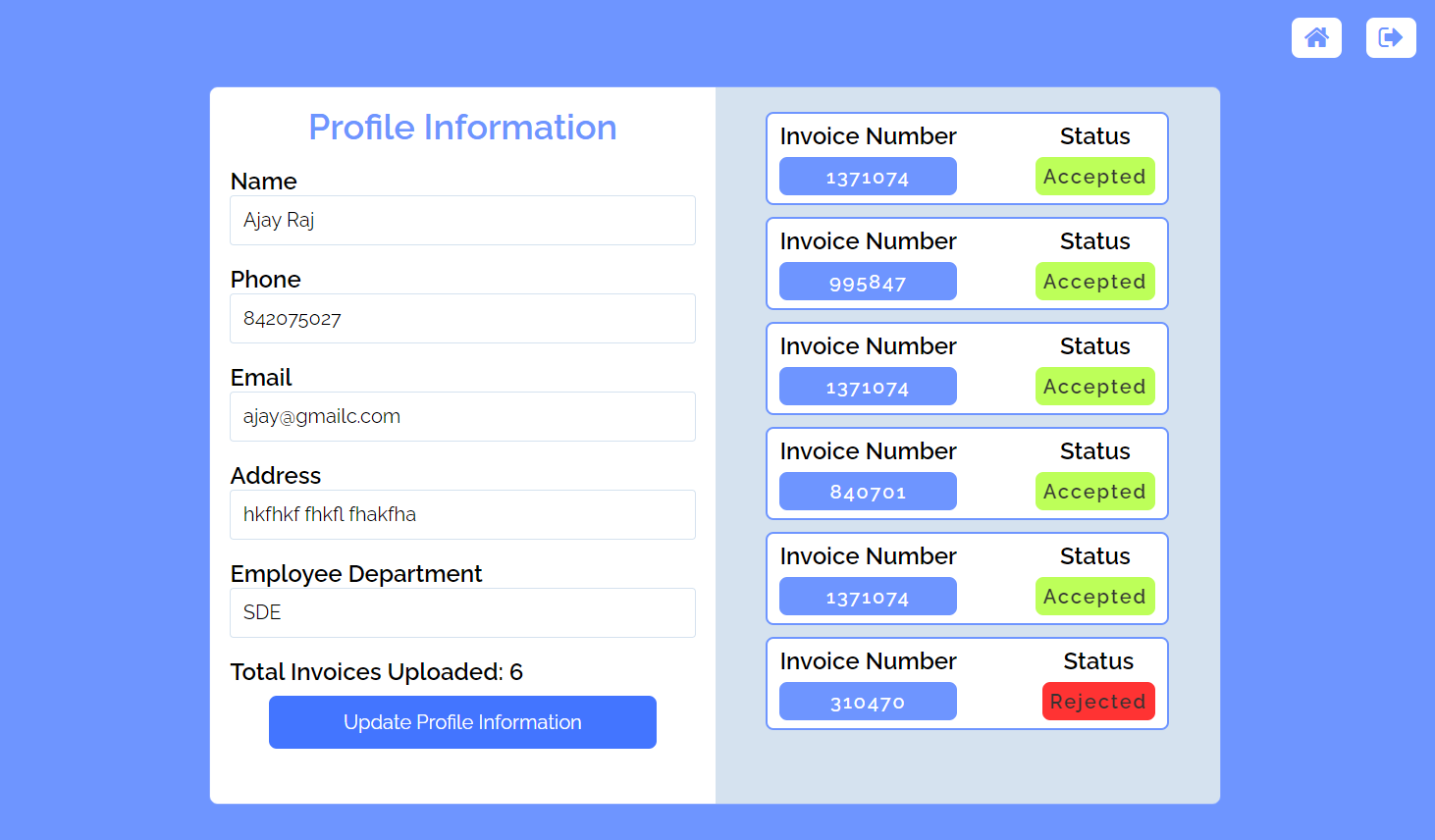
**Figure 3.1 Login page**

The login page serves as the gateway to the invoice detail extractor application, catering to both employees and the finance department within the company by the backend. Users are required to authenticate themselves by entering their user ID and password in order to access the application's features and functionalities.



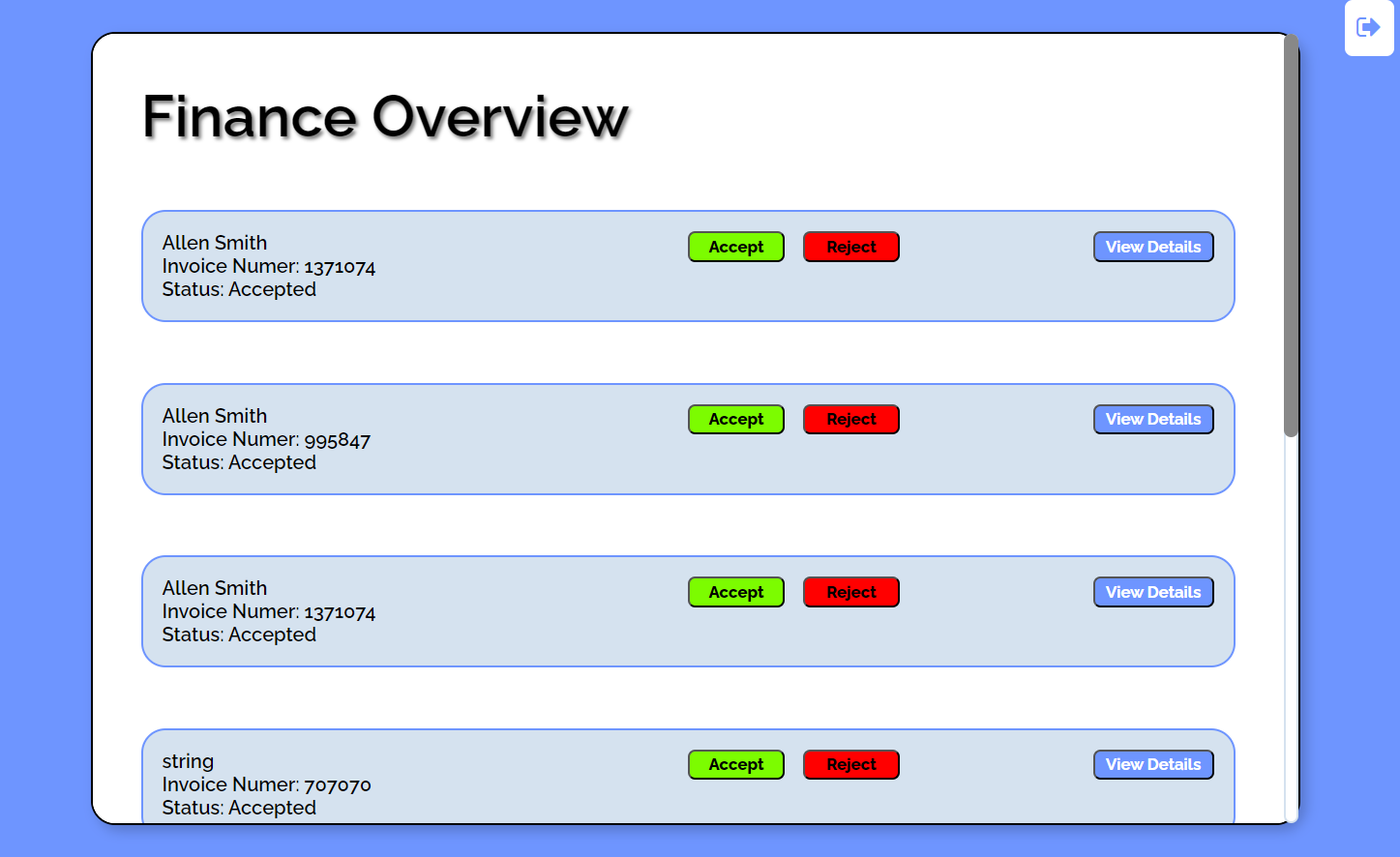
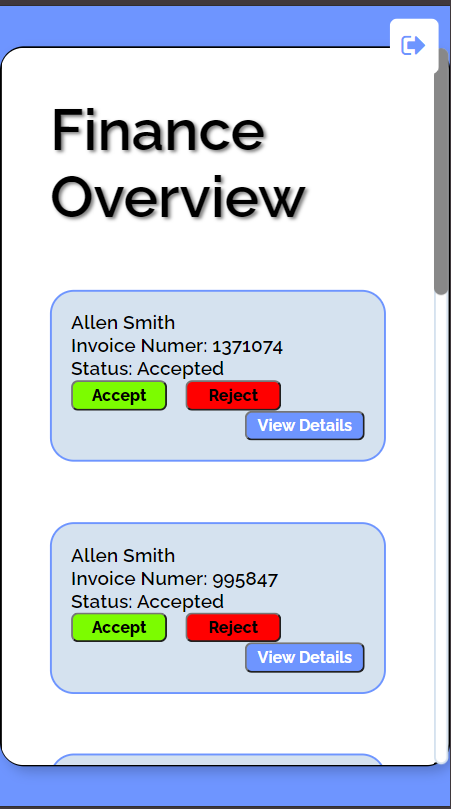
**Figure 3.2 Dashboard for User**

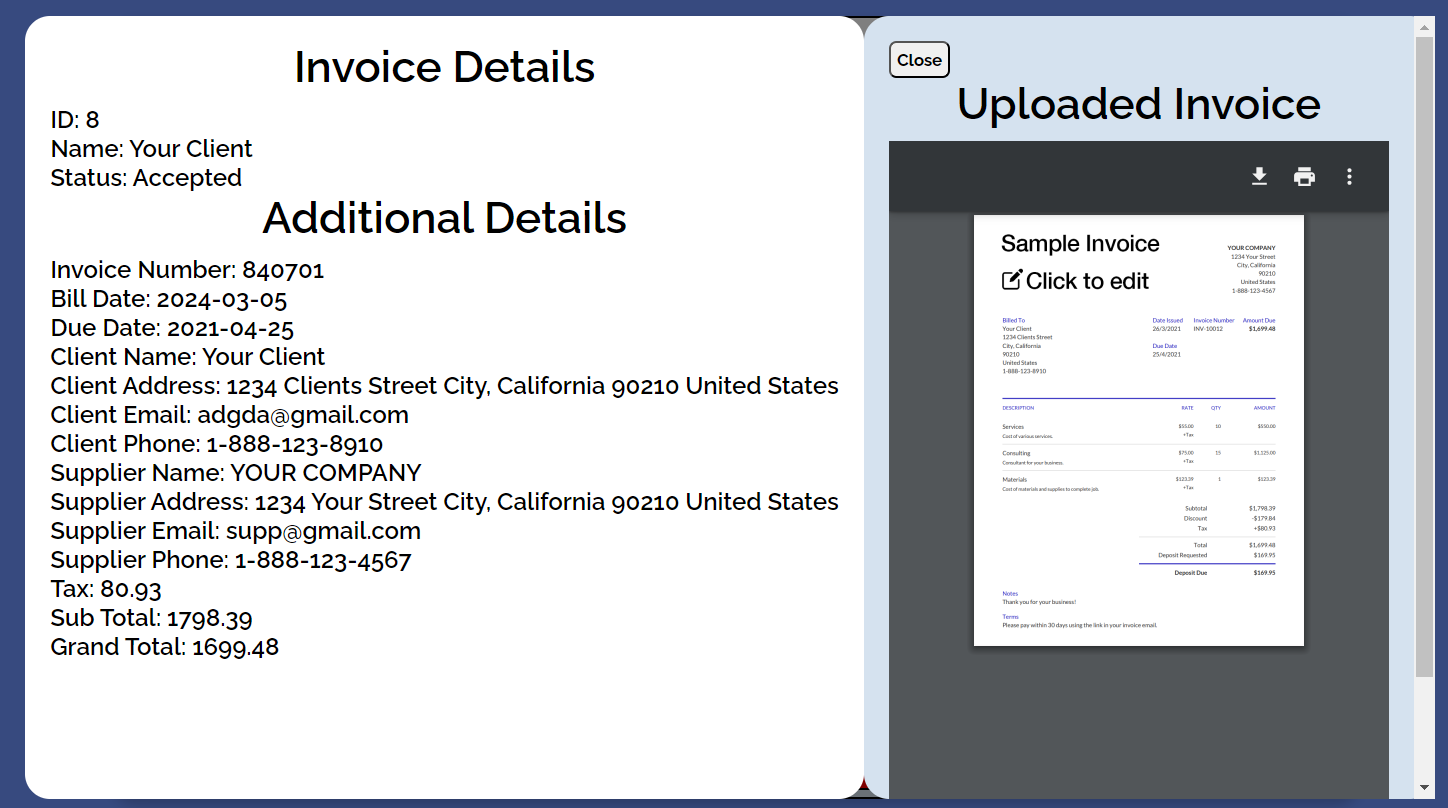
The dashboard simplifies the process for employees to submit invoices by enabling them to upload PDF or image files directly. Upon upload, the dashboard automatically extracts relevant details from the invoices and populates them into the corresponding form fields. Once the user confirms the information and submits the form, the request is forwarded to the backend system for further processing and approval. This streamlined approach enhances efficiency and reduces manual data entry efforts for both employees and the finance department.



**Figure 3.3 User Profile**

The user profile page provides users with the ability to view and edit their personal information, such as name, email, and address. Additionally, users can track the status of their submitted invoices, including whether they have been approved, pending, or rejected by the finance department.



**Figure 3.4 Finance dashboard**

The finance dashboard is a tool specifically tailored for members of the finance department to oversee and process invoice requests submitted by employees. Through this dashboard, finance department members can efficiently review, approve, or reject requests. Additionally, they can access comprehensive details for each request, aiding in informed decision-making and streamlined workflow management.

## ANALYSIS OF RESULTS

Accuracy Assessment:

* The LLM model proved high accuracy in extracting key fields such as invoice number, date, and total amount.
* The model produced consistent and valid responses for the sample input, showing robust performance in both finding relevant information and minimizing false positives.
* Some variability was seen in the performance across diverse types of invoices, with invoices involving tables presenting greater challenges but still achieving acceptable accuracy levels. The sample input didn’t include handwritten invoice, but their inclusion could have posed a threat to the model's accuracy and speed

Performance Evaluation:

* The extraction speed of the LLM model was satisfactory, with minimal latency in processing invoices.
* Resource consumption was within acceptable limits. To increase the throughput and decrease the latency of the overall project, the model was effectively hosted on a T4 GPU provided by Google Collab

**CHAPTER 4**

**4.1 LEARNING OUTCOMES**

* Automated Data Extraction Utilizing LLM Model: The design leverages the capabilities of the LLM model for automated extraction of key fields such as invoice number, date, and total amount from uploaded invoices. The project will provide an understanding of how the LLM model can be utilized for automated invoice data extraction.
* Technical Implementation of LLM Model: Helps in developing technical proficiency in implementing the LLM model for automated invoice data extraction. This includes understanding how to fine-tune and optimize the model for the specific task of extracting invoice details efficiently.
* The design seamlessly integrates the LLM-based extraction feature into the existing software application's workflow. This facilitates the development of integration skills and understanding of how to incorporate the LLM model's outputs into the application's data processing pipeline.
* Quality Assurance and Validation: The design incorporates validation checks and quality assurance measures to ensure the accuracy and reliability of the LLM-based extraction system. The project facilitates the acquisition of skills in evaluating and validating the model's outputs to minimize errors and discrepancies.
* User Experience Design: The design focuses on designing user-friendly interfaces for uploading invoices and presenting the extracted details to users. The project will provide insights into designing interfaces that facilitate user interaction with the LLM-based extraction feature, enhancing usability and user satisfaction.
* Project Management and Collaboration: The project enhances project management skills through collaborative planning, execution, and monitoring, with a focus on effective teamwork for seamless integration into the existing application.
* Compliance and Security Awareness: The project emphasizes compliance with data privacy and security regulations in implementing the extraction feature, fostering participants' awareness of regulatory requirements and their ability to implement necessary security measures for user data protection.
* Continuous Improvement and Feedback Incorporation: The design incorporates mechanisms for gathering user feedback on the performance of the extraction feature. It enables learning how to analyze feedback and iteratively enhance the model's accuracy and effectiveness over time.

**4.2 CONCLUSIONS AND FUTURE WORK**

In conclusion, the automated invoice data extraction project represents a significant step towards streamlining expense management processes. By reducing the burden of manual data entry, the system improves efficiency, minimizes errors, and enhances user experience. The successful implementation of this project demonstrates its feasibility and potential benefits for businesses and individuals alike.

In the future, there is a huge scope for new work and enhancements. One key area is the integration of machine learning algorithms to further improve the accuracy and reliability of the data extraction process. Additionally, expanding the system's capabilities to handle a broader range of invoice formats and languages would increase its usability and impact. Furthermore, ongoing user feedback and iterative improvements are essential to ensure the system meets evolving user needs and remains a valuable tool for expense management. Overall, the project lays a solid foundation for continued innovation in the field of automated invoice processing and holds promise for further advancements in this area