Business Intelligence Dashboards for Lending as SaaS

॥वसुधैव कुटुम्बकम्॥

THESIS SUBMITTED TO
Symbiosis Institute of Geoinformatics

FOR PARTIAL FULFILLMENT OF THE M. Sc. DATA SCIENCE AND SPATIAL ANALYTICS DEGREE

By
PRANAV KATARIYA
PRN: 22070243026
(2022 -24)

Symbiosis Institute of Geoinformatics
Symbiosis International (Deemed University)
5th Floor, Atur Centre, Gokhale Cross Road
Model Colony, Pune – 411016

CERTIFICATE

Certified that this thesis titled '**Business Intelligence Dashboards for Lending as SaaS**' is a bonafide work done by **Mr. Pranav Katariya** at '**Symbiosis Institute of Geoinformatics**', under our supervision.

Supervisor, Internal	**Supervisor, External**
Dr.Vidya Patkar	Mr. Abhilash Rathore
Symbiosis Institute of Geoinformatics	Fintaar Technologies Pvt Ltd

UNDERTAKING

The thesis titled '**Business Intelligence Dashboards for Lending as SaaS**' is the Bonafide work of **Mr. Pranav Katariya**.

University can use and reuse the dissertation work in future.

Date: 08/06/2024

Name of the student: Mr. Pranav Katariya

Signature

Name of the External Supervisor: Mr. Abhilash Rathore

Organization: Fintaar Technologies Pvt Ltd

Signature

ACKNOWLEDGEMENT

I would like to express my sincere gratitude and appreciation to all those who have contributed to the successful completion of this project.

First and foremost, I would like to thank my project supervisor Dr.Vidya Patkar and Mr. Abhilash Rathore for their guidance, valuable insights, and continuous support throughout the duration of this project. Their expertise and encouragement have been invaluable in shaping the direction of this work.

I am also grateful to the faculty members of for providing a conducive learning environment and access to resources that have aided in the successful execution of this project.

I extend my heartfelt thanks to my friends and classmates who have provided assistance and support whenever needed. Their encouragement and constructive feedback have been instrumental in refining the ideas and methodologies employed in this project.

Lastly, I would like to acknowledge my family for their unwavering support, patience, and understanding throughout my academic journey. Their love and encouragement have been the driving force behind my accomplishments.

I am deeply grateful to all those mentioned above and many others who have played a role, however big or small, in the successful completion of this project. Their contributions have been invaluable, and I am truly thankful for their assistance.

PRANAV KATARIYA

(M.sc Data Science and Spatial Analytics)

PRN: 22070243026

INDEX

PREFACE……………………………………………………………………………………8

1. INTRODUCTION……………………….……………………………………………..10
2. LITERATURE REVIEW……………..………………………………………………..12
3. MATERIALS AND METHODS..……………..………………………………………16
   3.1 Software and Packages …..…………………………………………………….…...16
      3.1.1 Power BI………....………………………….....………………..……..16
      3.1.2 Flask………………………………………………………..……………..…..16
      3.1.3 Flutter…………………………………………...………………....16
      3.1.4 MySQL...………………………………………………………………………17
      3.1.5 Google Gemini……………………..…………………………………….17
   3.2 Methodology………..…..……………………………………………………….....18
      3.2.1 Data Collection………………………………………………...……18
      3.2.2 Data Flow Architecture……………………………………………………..19
      3.2.3 Dataset……………………….………………………………………20
      3.2.4 Deployment Architecture………………………….……………………..22
      3.2.5 Natural Language to SQL Query Architecture………..……………………23
4. RESULTS AND DISCUSSION…………………..……………………………………24
   4.1 Natural Language to SQL Query Converter………………………………..24
   4.2 Sales………………………………………………………………………………26
      4.2.1 Sales Summary…………………………………………………………...26
   4.3 Loan Origination System (LOS)……………………………………………...27
      4.3.1 Loan Processing Summary…………………………………………………27
      4.3.2 Loan Processing Trend……………………………………………………..28
   4.4 Loan Management System……………………………………………………29
      4.4.1 Portfolio Summary……………………………………………………29
      4.4.2 Portfolio Monthly Summary…………………………………………….30
   4.5 Debt Collections…………………………………………………………………..31
      4.5.1 Debt Collection Summary…………………………………………………31
      4.5.2 Payment Trends…………………………………………………………..32
      4.5.3 Collector Performance……………………………………………………...33
5. CONCLUSION……..……………………………………………………………...34

REFERENCES…………………………………………………………………………...35

APPENDIX………………………………………………………………………….....36

List of Figures

Figure 1: Data Flow Diagram …………………………………………………………….…..19

Figure 2: Deployment Architecture Diagram………………………………….…….…………..22

Figure 3: Text to SQL Query Converter Diagram ………………………….……………...24

Figure 4: UI for Querying the Database……………………………….……...................26

Figure 5: Sales Summary Dashboard……………………………….........................27

Figure 6: Loan Processing Summary Dashboard……………………….……...............28

Figure 7: Loan Processing Trend Dashboard………………….………………………..29

Figure 8: Portfolio Summary Dashboard…………………….……………………………… 30

Figure 9: Portfolio Monthly Performance Dashboard…………………….……………….31

Figure 10: Debt Collection Summary Dashboard…………………………………….32

Figure 11: Payment Trends Dashboard………………………………………………….33

Figure 12: Collector Performance Dashboard……………………………………………..34

Abbreviation List

SaaS: Software as a Service

CXO: Chief Executive Officer

BI: Business Intelligence

LMS: Loan Management System

LOS: Loan Origination System

ETL: Extract Transform Load

API: Application Programming Interface

UI: User Interface

PL: Personal Loan

HL: Home Loan

AU: Automobile Loan

BL: Business Loan

PREFACE

In the dynamic and competitive realm of the financial industry, particularly within the lending sector, the ability to swiftly adapt to market changes and make informed decisions is paramount. The " Business Intelligence Dashboards for Lending as SaaS" project emerges as a pivotal solution aimed at equipping lending institutions with state-of-the-art Business Intelligence (BI) tools, delivered through a flexible and scalable Software as a Service (SaaS) model.

Over the past few years, the lending landscape has undergone significant transformations, driven by technological advancements, changing customer expectations, and evolving regulatory requirements. The data accumulated during this period serves as a critical foundation for our project, offering invaluable insights into lending trends, borrower behavior, and operational efficiencies.

This project is meticulously designed to address the growing need for real-time, actionable intelligence among Chief Experience Officers (CXOs) in the lending sector. By leveraging comprehensive data analysis and sophisticated dashboard development, we aim to empower CXOs with the tools they need to drive strategic initiatives and enhance overall business performance.

The journey begins with the creation of a robust Data Warehouse Schema, ensuring a solid and scalable foundation for data storage and retrieval. This is followed by the implementation of an efficient Data Pipeline that guarantees seamless data flow and integration from multiple sources. The culmination of these efforts is the development of intuitive and interactive dashboards, specifically tailored to meet the unique needs of CXOs in the lending industry.

Key areas such as Sales, Loan Management System (LMS), Loan Origination System (LOS), and Debt Collections are meticulously covered in these dashboards, providing a comprehensive view of the business operations. Furthermore, the integration of these dashboards into a web portal and a mobile application ensures that CXOs have easy access to critical information, anytime and anywhere.

The real-time update feature is a cornerstone of this project, enabling CXOs to stay informed with immediate insights, thereby facilitating prompt and informed decision-making. Rigorous testing and stabilization processes are employed to ensure the reliability and robustness of the entire system, guaranteeing a seamless user experience.

In essence, the " Business Intelligence Dashboards for Lending as SaaS" project is not just a technological initiative but a strategic enabler designed to transform the way lending institutions operate and make decisions. By harnessing the power of data from the past three years and presenting it through sophisticated BI tools, we are setting the stage for a future where lending institutions can thrive in a rapidly evolving financial landscape.

1. INTRODUCTION

In today's fast-paced financial landscape, the ability to access real-time insights and make data-driven decisions is essential for success, particularly in the lending sector. However, many Chief Experience Officers (CXOs) of lending institutions have struggled with obtaining a comprehensive, up-to-the-minute view of their business operations. This challenge has often resulted in delayed decision-making and missed opportunities.

Recognizing this critical need, the " Business Intelligence Dashboards for Lending as SaaS" project was conceived. This initiative aims to bridge the gap by providing lending institutions with advanced Business Intelligence (BI) solutions delivered as Software as a Service (SaaS). The project is designed to offer CXOs immediate and actionable insights into various aspects of their business, thereby facilitating informed decision-making and strategic planning.

The core of this project lies in its ability to transform vast amounts of data into meaningful, real-time information. Covering the data from January 2021 to March 2024, the project leverages historical and current data to build a robust and insightful BI system. This system is intended to empower CXOs by presenting critical business metrics and trends through dynamic and interactive dashboards.

The project encompasses several key components:

1. **Dynamic Dashboard Development:**
   - Creation of intuitive and user-friendly dashboards tailored specifically for CXOs in the lending sector.
   - Comprehensive coverage of crucial business areas such as Sales, Loan Management System (LMS), Loan Origination System (LOS), and Debt Collections.
2. **Accessibility and Integration:**
   - Seamless embedding of dashboards into a web portal and a mobile application, ensuring easy access across various devices and platforms.
   - Enhanced user accessibility to critical business insights anytime, anywhere.
3. **Real-time Updates:**
   - Provision of real-time updates to CXOs, ensuring they are constantly informed about their business operations.
   - Facilitation of prompt and effective decision-making through timely and relevant information.

4. **Testing and Stabilization:**
    - Rigorous testing processes to identify and address potential issues, ensuring the system's stability and reliability.
    - Delivery of a seamless and consistent user experience through thorough stabilization efforts.
5. **Natural Language Query Interface:**
    - Input Form: The application provides an input form where CXOs can type their questions in natural language. This form is integrated into the home page or specific dashboard pages.
    - Google Gemini API Integration: The application sends the natural language question to the Google Gemini API, which generates the corresponding SQL query.
    - SQL Execution: The generated SQL query is executed against the MySQL database. The database contains various datasets relevant to the lending business, such as leads, loan processing, and collections data.
    - Results Display: The results of the SQL query are displayed in a user-friendly format, allowing CXOs to gain insights without needing to understand or write SQL.

The " Business Intelligence Dashboards for Lending as SaaS" project is not merely a technological advancement but a strategic necessity for lending institutions. By addressing the pressing needs of CXOs and providing them with real-time, actionable insights, this project sets the stage for enhanced operational efficiency and competitive advantage in the ever-evolving financial sector.

2. LITERATURE REVIEW

In the domain of big data analytics, Doko and Mishkovski undertook a comprehensive investigation into the utilization of Power BI for advanced analytics in the context of credit registry management. Their research, [1], delves into the methodology and application of Power BI tools in the banking sector, specifically focusing on credit registry databases. Through their study, the authors aimed to leverage the capabilities of Power BI to analyze and visualize large volumes of data without requiring extensive IT knowledge.The authors began by emphasizing the growing importance of effective data analysis and visualization in the face of increasing data volumes, particularly in financial services. They highlighted the challenges encountered in traditional systems and tools when dealing with large datasets and advocated for the adoption of modern analytics tools like Power BI. Doko and Mishkovski illustrated the significance of Credit Registry databases in contributing to credit risk management and maintaining the stability of the banking system, positioning it as a prominent dataset for big data analytics. The research paper systematically outlined the process of implementing Power BI tools, starting from data importation and preprocessing to model creation, visualization, and deployment on the Power BI Service [1]. Through a series of stages, including SQL server processing, Excel data integration, and Power BI Desktop usage, the authors demonstrated how they processed and analyzed the credit registry dataset effectively.

The paper [2] presents the development of a dashboard aimed at enhancing the sales performance of credit cards in a banking company. The study emphasizes the importance of data visualization in understanding complex datasets and highlights the challenges faced by the banking industry in manually processing sales reports using tools like Microsoft Excel [2]. The authors propose the creation of a dashboard using Microsoft Power BI to provide insights into credit card sales performance, focusing on key metrics such as sales by brand, product, channel, and branch [2]. The dashboard aims to help sales agents identify performing markets, analyze customer demographics, and formulate strategies to improve sales performance [2]. Through a systematic methodology involving data gathering, analysis, and validation, the authors demonstrate the effectiveness of the dashboard in facilitating data-driven decision-making and optimizing credit card sales operations [2]. The paper concludes by emphasizing the significance of data visualization in driving business performance and highlights the potential benefits of leveraging modern analytics tools in the banking industry [2].

The paper [3] delves into API monitoring, a critical process for ensuring the reliability, availability, and accuracy of Application Programming Interfaces (APIs) [3]. It underscores the importance of visualising datasets in analytics, particularly in today's context, where tools such as Microsoft Power BI play a pivotal role [3]. The focus lies on delineating the features of Power BI, its relevance in banking data analysis, and its integration with Shivalik Bank's API through interactive visualisation [3]. Additionally, it sheds light on the organisational environment of Shivalik Bank, the array of tools employed including Postman, Finacle, PowerBI, Putty, SQL, and the benefits of utilising such techniques in application testing and beyond [3].

Kristantoputra and Subramanian propose a Business Intelligence (BI) system to streamline Know Your Customer (KYC) assessments for credit card services [4]. Utilising tools like Microsoft SQL Server and Tableau, they develop a data warehouse and dashboard to efficiently process applicant data and evaluate loan repayment status. The study underscores the significance of BI in enhancing decision-making and operational productivity within the banking sector, offering a solution to the challenges posed by manual KYC processes.

Inamdar and Gursoy explore the implementation of a highly adaptable and customisable Business Intelligence (BI) platform hosted on the cloud [5]. They highlight the shift towards cloud-based storage and platforms to address challenges faced by traditional BI systems, such as handling large volumes of data and integration complexities. The paper discusses the architecture of the proposed BI platform, its application in the health insurance industry, and the benefits of cloud computing for BI, including scalability and cost reduction [5].

The research paper [6] discusses the utilization of data analytics techniques in financial institutions to enhance operational efficiency and market expansion. Through the implementation of Business Intelligence (BI) tools and data warehouse techniques, the study focuses on analyzing borrower behavior and response to call center interactions. The star schema is employed for data warehouse design, with dimensions such as age group and gender being evaluated for their impact on loan collection efficiency. Decision tree analysis is proposed for future works to further enhance loan repayment prediction. The paper also highlights related studies on fraud analysis and loan repayment factors, emphasizing the importance of data-driven approaches in financial decision-making. Additionally, it outlines the methodology involving Extract, Transform, Load (ETL) processes, data analysis, and reporting techniques, with a focus on non-functional requirements such as security.

Author(s) Name	**Title**	**Tools Used**	**Summary**
Fisnik Doko, Igor Mishkovski	Advanced analytics of Big Data using Power BI: Credit Registry Use Case	Microsoft Power BI	The research investigates the use of Power BI for advanced analytics in credit registry management within the banking sector, highlighting its effectiveness in processing and visualising large datasets.
Darrel John Beltran, Yves Kangleon, Ariel Kelly Balan and Joel de Goma	Credit Card Sales Performance Dashboard	Microsoft Power BI	The paper discusses the development of a Microsoft Power BI dashboard to enhance credit card sales performance in banking, underlining the role of data visualisation in facilitating informed decision-making and operational optimisation.
Sahu, Shaswat, Sidhu	API Monitoring Dashboard	Microsoft Power BI, Excel	The paper explores API monitoring's significance and the integration of Microsoft Power BI for visualising banking data, particularly within Shivalik Bank's operational framework, highlighting its role in ensuring API reliability and enhancing application testing efficiency.

Mario Caesar Kristantoputra and Preethi Subramanian	Implementation of Business intelligence in Know Your Customer (KYC) for credit card customers' loan repayment status	Tableau	The proposal outlines a Business Intelligence system utilising Microsoft SQL Server and Tableau to automate KYC assessments for credit card services, aiming to improve decision-making and operational efficiency within the banking sector.
Siddhesh Rajesh Inamdar , Kemal Gursoy	Cloud hosted business-data driven BI platforms	Microsoft Power BI, MySQL	The paper discusses the implementation of a highly adaptable cloud-based Business Intelligence (BI) platform, highlighting its architecture, application in the insurance industry, and advantages such as scalability and cost reduction, aiming to overcome challenges faced by traditional BI systems.
Maduka Ashan, Madhuranga Chathura, PPG Dinesh Asanka	Loan Data Analysis Using Data Warehouse Techniques	SQL Server Integration Services (SSIS), Excel	The research paper explores analytics' role in financial institutions, focusing on BI tools and data warehousing to analyse borrower behaviour proposing decision tree analysis for loan repayment prediction and data-driven approaches in decision-making.

3. MATERIALS AND METHODS

3.1 Software and Packages

In this study, Power BI, Flask, Flutter, and MySQL are used for generating reports, embedding dashboards, and establishing database connections.

3.1.1 Power BI

Power BI is a powerful business analytics tool developed by Microsoft. It enables users to visualize data and share insights across their organization or embed them in an app or website. In this project, Power BI is used for:

Generating Reports: Creating detailed and interactive reports that provide CXOs with comprehensive insights into key business metrics and trends.

Data Visualization: Transforming raw data into visually compelling dashboards that facilitate easy interpretation and informed decision-making.

3.1.2 Flask

Flask is a lightweight and flexible web framework for Python. It is designed to make it easy to build web applications and services. In this project, Flask is used for:

Web Portal Development: Creating a web portal that serves as the central access point for CXOs to interact with the BI dashboards.

Embedding Dashboards: Seamlessly integrating Power BI dashboards into the web portal to ensure easy access and interaction.

API Development: Building RESTful APIs to manage data flows between the front-end and the back-end services.

3.1.3 Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop natively compiled applications for mobile, web, and desktop from a single codebase. In this project, Flutter is used for:

Mobile Application Development: Building a mobile application that allows CXOs to access the BI dashboards on-the-go.

Embedding Dashboards: Ensuring that the interactive Power BI dashboards are accessible and user-friendly within the mobile application.

Cross-Platform Compatibility: Providing a seamless user experience across both iOS and Android devices.

3.1.4 MySQL

MySQL is a widely-used open-source relational database management system. It is known for its reliability, scalability, and ease of use. In this project, MySQL is used for:

Database Management: Storing and managing the vast amounts of data collected from January 2021 to January 2024.

Data Integration: Serving as the primary database for integrating data from various sources into the BI system.

Data Connection: Facilitating real-time data connectivity and retrieval for the Power BI dashboards to ensure that CXOs have access to the latest information.

3.1.5 Google Gemini

Google Gemini is an AI-powered natural language processing tool. In this project, it enables CXOs to query the MySQL database using everyday language, facilitating real-time access to business insights without needing SQL knowledge.

3.2 Methodology

3.2.1 Data Collection

The data originates from various sources such as web applications and mobile applications, where leads are initially collected from the company's clients who are engaged in lending. These leads are then funnelled into the Loan Origination System (LOS), which processes and evaluates their eligibility for loans. Within the LOS, the status of each loan application is meticulously tracked, with possible statuses including Accepted, Rejected, New Leads, and Disbursed. Only the loans with a status of Disbursed, meaning the funds have been released to the applicant, are forwarded to the Loan Management System (LMS).

The LMS manages the lifecycle of these disbursed loans, providing detailed information such as Outstanding Amount, Paid Amount, Overdues, and Loan Type. This system allows for comprehensive management of each loan's financial health and repayment status. If there are any overdue amounts, this data is transferred to the Debt Collection system, which handles the recovery of these debts. The Debt Collection system provides detailed information about the overdue accounts, ensuring that necessary recovery actions are taken promptly.

This comprehensive data flow ensures meticulous tracking and management of loan applications from their inception through to loan disbursement and repayment. It allows the company to maintain a detailed and up-to-date record of each client's financial interactions, facilitating efficient oversight and action on overdue amounts. By integrating these systems, the company can optimize its lending operations, improve customer service, and enhance its financial performance through better debt recovery processes. This seamless integration of data collection, processing, and management underscores the importance of robust information systems in the modern lending industry.

3.2.2 Data Flow Architecture

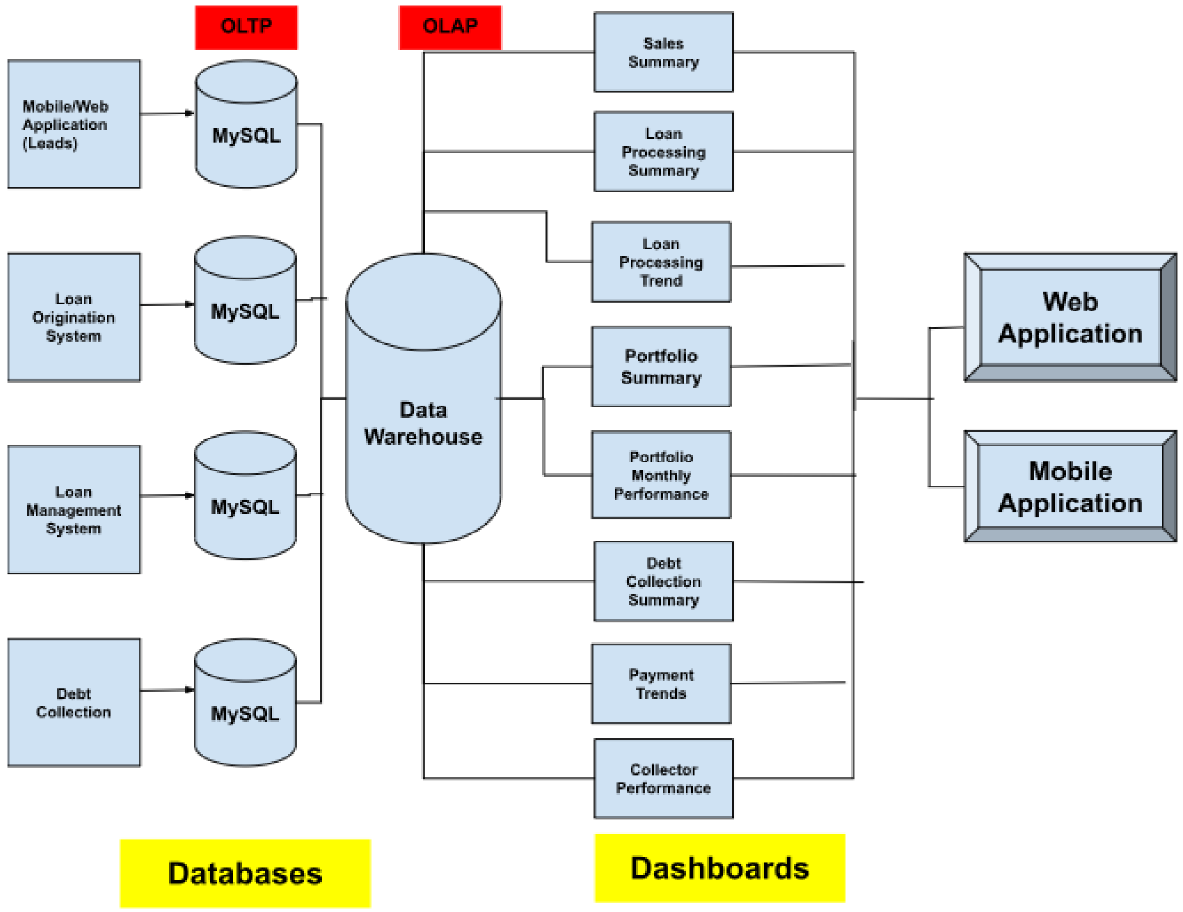


Figure 1: Data Flow Diagram

The provided diagram illustrates the comprehensive data flow for the "BI Dashboards for Lending as SaaS" project. Data is initially collected from various sources, including leads from mobile and web applications, the Loan Origination System (LOS), the Loan Management System (LMS), and Debt Collection processes. Each of these sources stores data in its respective MySQL database, functioning as OLTP (Online Transaction Processing) systems. This data is then extracted and transferred to a centralized Data Warehouse, an OLAP (Online Analytical Processing) system, where it is organized for efficient querying and analysis. It is noteworthy that none of the tables in the Data Warehouse are related to each other, ensuring a straightforward data structure.

From the Data Warehouse, data is accessed via APIs to generate a variety of dashboards, including Sales Summary, Loan Processing Summary, Loan Processing Trend, Portfolio

Summary, Portfolio Monthly Performance, Debt Collection Summary, Payment Trends, and Collector Performance. These dashboards provide CXOs with comprehensive insights into key business metrics. The dashboards are then embedded into both a web application, developed using Flask , and a mobile application, developed using Flutter, ensuring that CXOs can access real-time, actionable insights seamlessly across different platforms. This integrated approach empowers lending institutions with the tools they need to make informed decisions and enhance their operational efficiency.

3.2.3 Dataset

In this project, the following attributes are distributed across eight different datasets, each tailored to provide insights for specific dashboards:

No of Leads: The number of customers applying for loans. This metric helps in understanding the demand and the effectiveness of marketing strategies.

Applied Loan Amount: The total amount of loan applied for by the customers. It provides insights into the average loan size and helps in financial forecasting.

State: The state from which the loan application originates. This helps in regional analysis and identifying geographical trends.

City: The city from which the loan application originates. It further refines geographical analysis and helps in city-specific strategic planning.

Loan Type: The category of loan applied for, such as Home Loan (HL), Personal Loan (PL), and Auto Loan (AU). This helps in analyzing the demand for different loan products.

DSA (Direct Selling Agent): The agent responsible for selling the loan products. It helps in tracking the performance of different DSAs.

Month: The month during which the loan application was made. It is used for monthly trend analysis.

Year: The year during which the loan application was made. It helps in yearly performance comparison.

Loan Status: The current status of the loan application, such as New Leads, Approved, Rejected, Pending, and Disbursed. It provides insights into the loan processing pipeline.

Date: The specific date of the loan application or related activity. It helps in daily tracking and time-series analysis.

Bucket: Categorization based on the number of days the payment is overdue, such as 0-30 days, 30-60 days, 60-90 days, and 90+ days. It helps in aging analysis of overdue payments.

OS Loan Amount (Outstanding Loan Amount): The remaining amount of the loan that needs to be paid by the borrower. It is crucial for financial health assessment.

Overdues: The amount of money that is overdue for payment. This helps in assessing the risk and effectiveness of collections.

Paid: The amount of loan repaid by the customer. It is essential for tracking repayment performance.

Product: The specific loan product being offered. It helps in product-specific performance analysis.

Customer Segment: The classification of customers based on their profession or background, such as Housewife, Teacher, Scientist, Lawyer, etc. It helps in segment-specific targeting and analysis.

Income Group: The income classification of the customers, such as Low, Medium, and High. It aids in understanding the financial profile of the customer base.

Agency: The agency to which the DSA belongs. It helps in analyzing the performance of different agencies.

Collector: The individual assigned to recover overdue payments. This helps in assessing the efficiency of collection efforts.

No of Calls/Day: The number of calls made by the collector in a day. It measures the activity level of collectors.

PTP/Day (Promise to Pay): The number of promises to pay received by the collector in a day. It is an indicator of the effectiveness of collection calls.

Not Connected/Day: The number of calls made by the collector that did not connect. It helps in understanding the challenges faced in collections.

Not Ready to Pay/Day: The number of customers who are not ready to pay after being contacted. It indicates potential issues in the repayment process.

Disputes Reported/Day: The number of disputes reported by customers in a day. It helps in identifying and addressing customer grievancesPaid no of loans/day.

3.2.4 Deployment Architecture

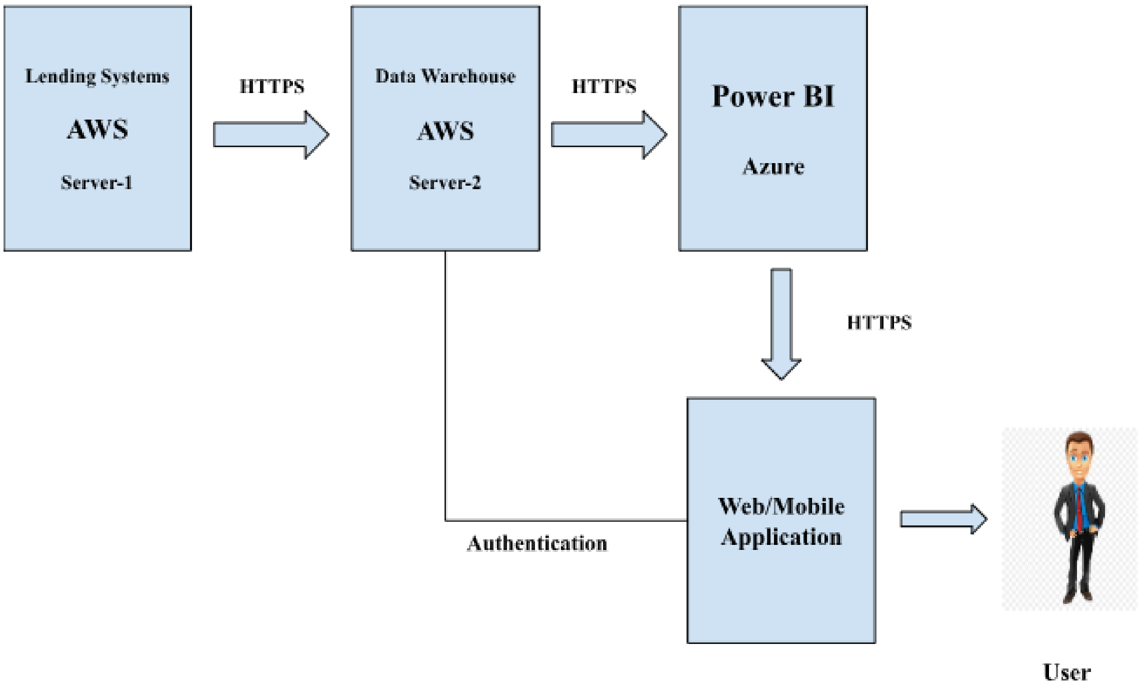


Figure 2: Deployment Architecture Diagram

The deployment architecture illustrates the flow of data within the system. Initially, data is received and stored in AWS Server-1. This serves as the entry point for incoming data. Subsequently, the data is transmitted securely over HTTPS to the Data Warehouse, which resides on AWS Server-2. Here, the data undergoes processing, analysis, and storage in a structured manner.

Following its processing within the Data Warehouse, the data is then forwarded to Power BI, a business analytics tool hosted on Azure. This transfer also occurs via HTTPS, ensuring encryption and security throughout the transmission process. Once received by Power BI, the data is visualised, allowing users to gain insights and make informed decisions based on the analysed data via Web/Mobile application.

3.2.5 Natural Language to SQL Query Architecture

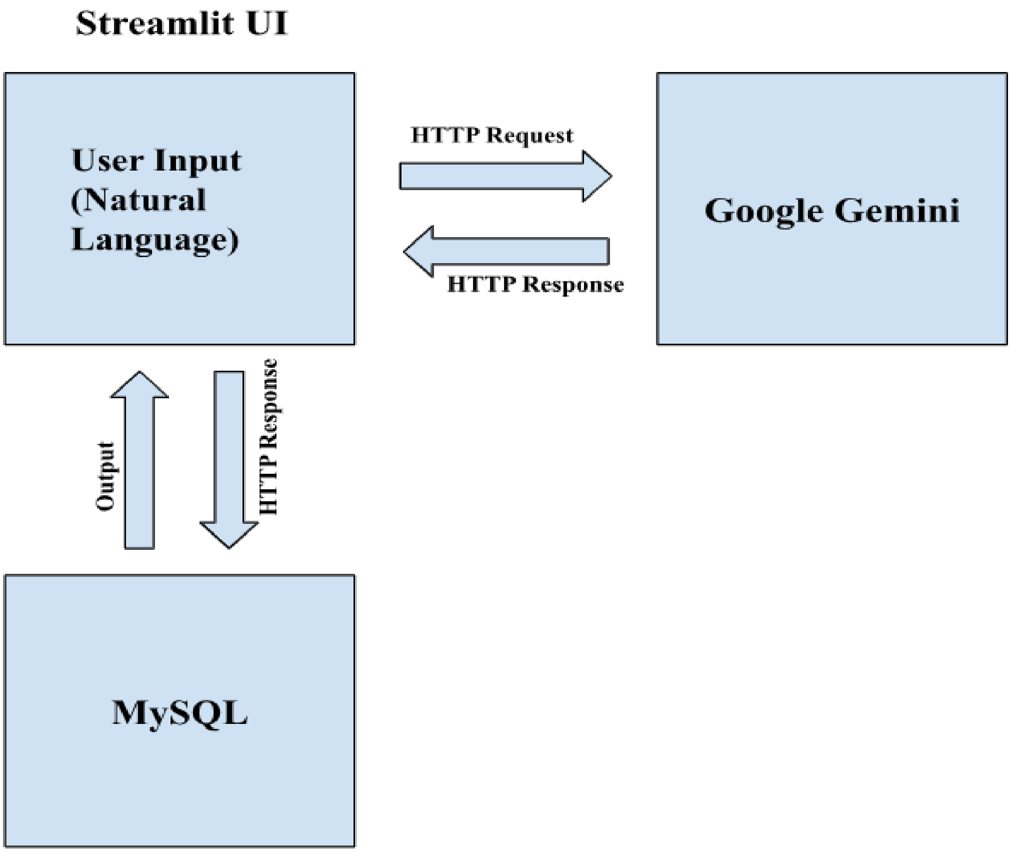
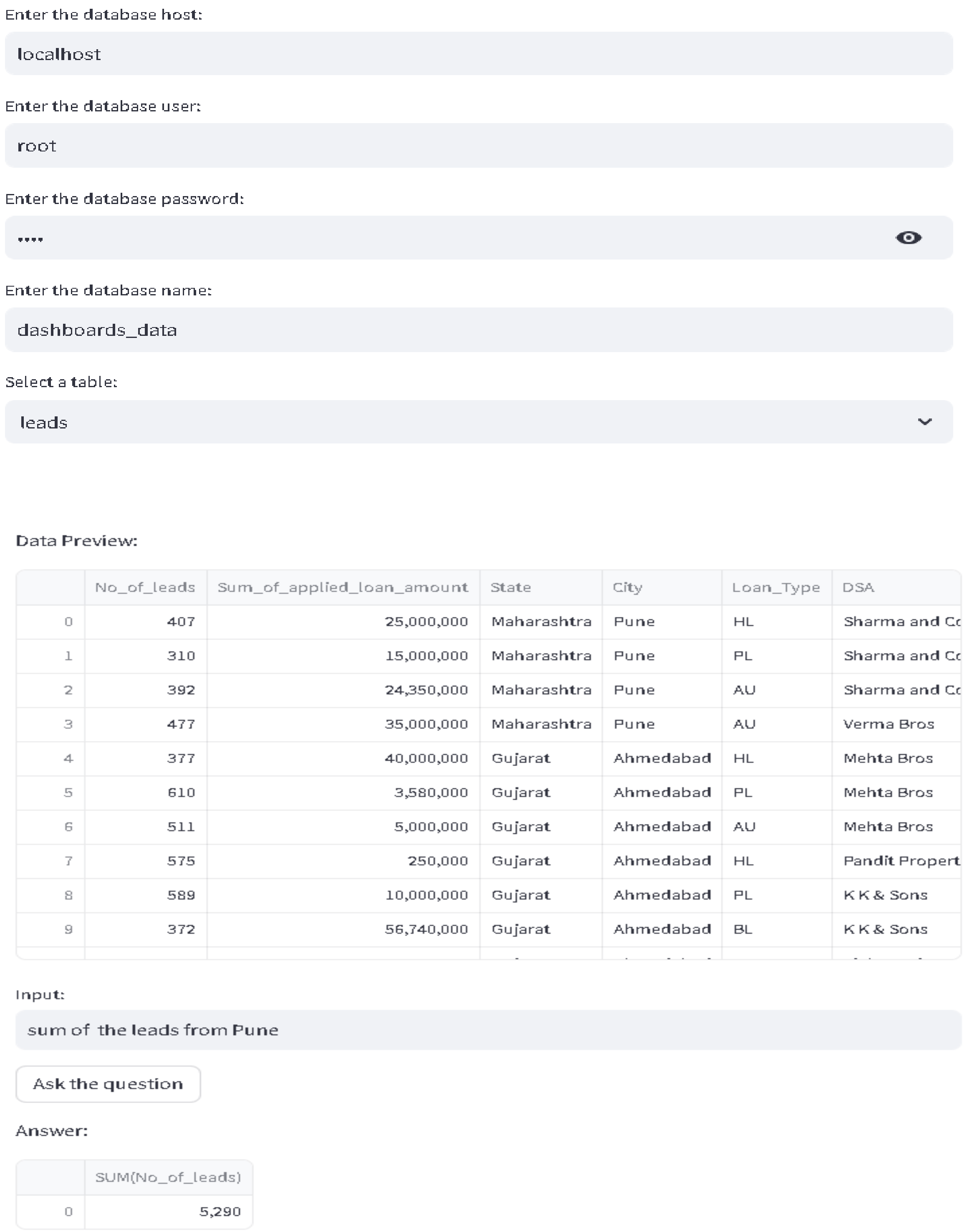


Figure 3: Text to SQL Query Converter Diagram

The system begins with the Streamlit UI, where users input queries or commands in natural language. This input is then processed by the Google Gemini API, which interprets the natural language input and converts it into structured data or commands. The processed information is then returned to the Streamlit UI, where users may be presented with relevant options or information. Once the user's intent is understood, the backend, implemented in Python or another suitable language, takes over. It interacts with the Streamlit UI, external APIs like the Google Gemini API, and the MySQL database. The backend executes SQL queries against the MySQL database based on the interpreted user input or other application logic. The MySQL database stores the application data and returns query results to the backend. Finally, the backend can process these results further or display them back to the user through the Streamlit UI. This architecture enables users to interact with the application using natural language while ensuring efficient data processing and retrieval from the MySQL database.

4. RESULTS AND DISCUSSION

4.1 Natural Language to SQL Query Converter



Enter the database host:

localhost

Enter the database user:

root

Enter the database password:

••••

Enter the database name:

dashboards_data

Select a table:

leads

Data Preview:

	No_of_leads	Sum_of_applied_loan_amount	State	City	Loan_Type	DSA
0	407	25,000,000	Maharashtra	Pune	HL	Sharma and C
1	310	15,000,000	Maharashtra	Pune	PL	Sharma and C
2	392	24,350,000	Maharashtra	Pune	AU	Sharma and C
3	477	35,000,000	Maharashtra	Pune	AU	Verma Bros
4	377	40,000,000	Gujarat	Ahmedabad	HL	Mehta Bros
5	610	3,580,000	Gujarat	Ahmedabad	PL	Mehta Bros
6	511	5,000,000	Gujarat	Ahmedabad	AU	Mehta Bros
7	575	250,000	Gujarat	Ahmedabad	HL	Pandit Propert
8	589	10,000,000	Gujarat	Ahmedabad	PL	K K & Sons
9	372	56,740,000	Gujarat	Ahmedabad	BL	K K & Sons

Input:

sum of the leads from Pune

Answer:

	SUM(No_of_leads)
0	5,290

Figure 4: UI for Querying the Database

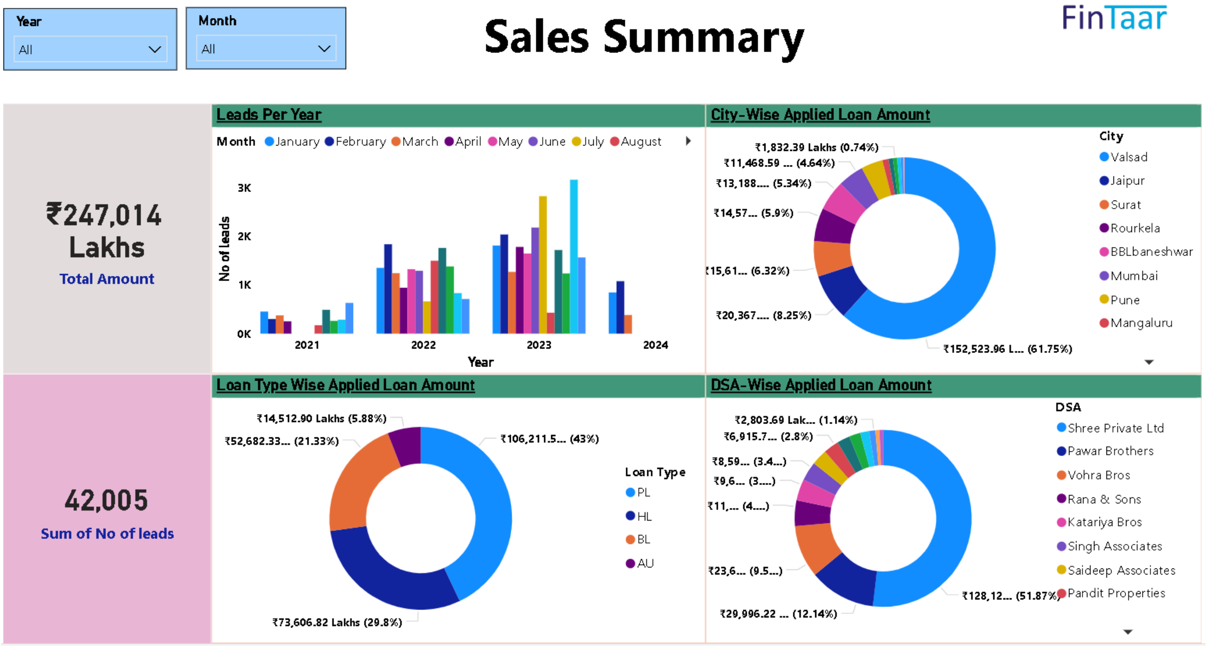
The image shows a web application interface created with Streamlit and integrated into a Flask app, designed with CXOs (Chief Experience Officers) in mind. In this interface, users are prompted to enter essential details for connecting to a MySQL database, including the host, user, password, and database name. Additionally, users can input business-related questions in natural language in a designated text box.

When a CXO submits their natural language query, the application processes this input using the Google Gemini API, which translates the natural language query into an SQL query. The generated SQL query is then sent to the MySQL database for execution.

Once the MySQL database executes the query, the results are returned and displayed within the interface. This setup allows CXOs to interact with the database using natural language, eliminating the need for technical knowledge of SQL. By leveraging the Google Gemini API, the application translates their input into precise SQL commands, retrieves the desired data from the database, and presents it in a user-friendly manner. This enables CXOs to make data-driven decisions efficiently, without relying on technical staff for data queries.

4.2 Sales

4.2.1 Sales Summary



Sales Summary

Year
All

Month
All

₹247,014 Lakhs
Total Amount

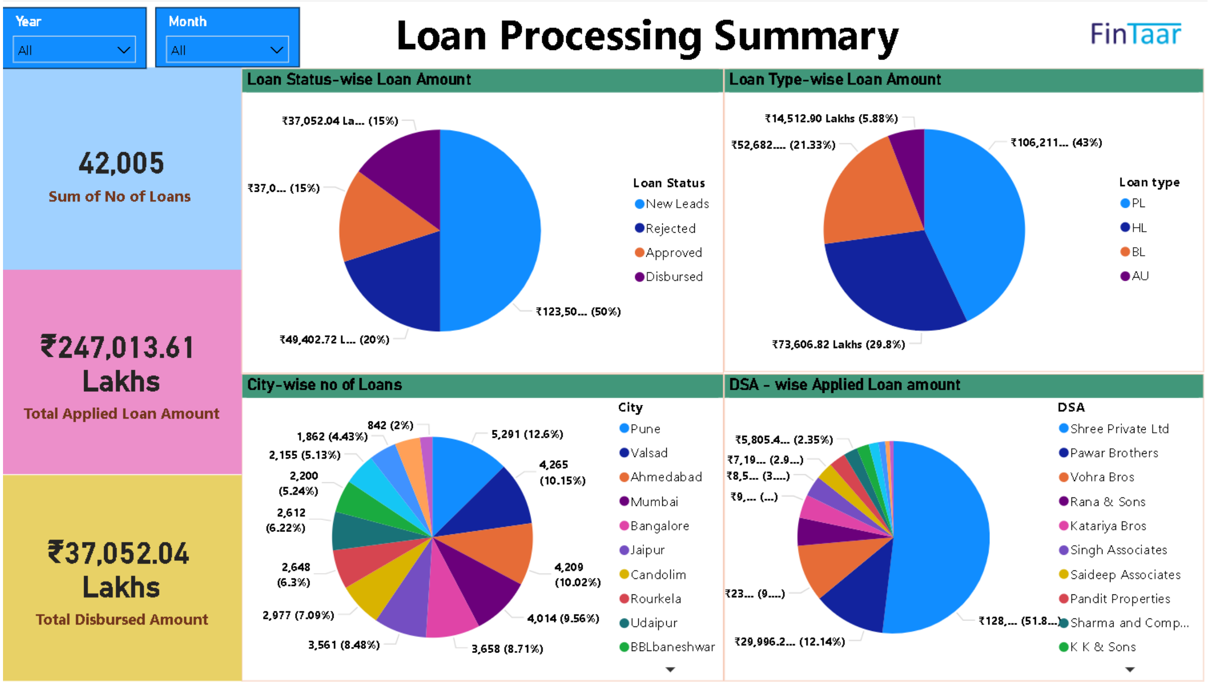
42,005
Sum of No of leads

Figure 5: Sales Summary Dashboard

The Sales Summary report provides a comprehensive overview of the loan applications and leads over the years. The report highlights a total applied loan amount of ₹247,014 Lakhs and a total of 42,005 leads. It includes a bar chart that displays the number of leads received each month from 2021 to 2023, illustrating trends and fluctuations in lead generation over these years. A pie chart shows the distribution of the applied loan amount across different cities, with notable contributions from cities like Mumbai, Pune, Surat, and Jaipur. Another pie chart breaks down the applied loan amount by loan type, indicating the proportions of Personal Loans (PL), Home Loans (HL), Business Loans (BL), etc. Additionally, the report features a pie chart that presents the distribution of the applied loan amount by Direct Selling Agents (DSAs), showcasing the contributions of various DSAs such as Shree Private Ltd, Pawar Brothers, and Vohra Bros. This detailed breakdown of loan applications provides valuable insights into regional distribution, loan types, and the performance of different DSAs, serving as an essential tool for understanding the dynamics of loan applications and effectively managing the lending process.

4.3 Loan Origination System (LOS)

4.3.1 Loan Processing Summary



Year: All

Month: All

42,005
Sum of No of Loans

₹247,013.61 Lakhs
Total Applied Loan Amount

₹37,052.04 Lakhs
Total Disbursed Amount

Figure 6: Loan Processing Summary Dashboard

The Loan Processing Summary for the financial institution, covering the years 2021 to 2024, reveals substantial data about the institution's loan activities. The total applied loan amount stands at ₹247,013.61 Lakhs, while the total disbursed amount is significantly lower at ₹37,052.04 Lakhs. In terms of loan status, the largest amount applied for was in New Leads, totaling ₹123,506.00 Lakhs, though the majority of applications were classified as new leads. Disbursed loans constituted the smallest portion, both in terms of amount (₹37,052.04 Lakhs) and number (2,648). The predominant loan type was PL loans, accounting for ₹106,211.56 Lakhs. Analyzing by city, the highest number of loan applications were from Pune (5291) and Valsad (4265). Regarding Direct Selling Agents (DSAs), DSA Pawar Brothers submitted the most applications, whereas DSA Shree Private Ltd accounted for the highest loan amount applied for, totaling ₹128,125.57 Lakhs. The considerable gap between the applied and disbursed amounts suggests a high rate of loan rejections or withdrawals, highlighting a critical aspect of the institution's loan processing dynamics.

4.3.2 Loan Processing Trend

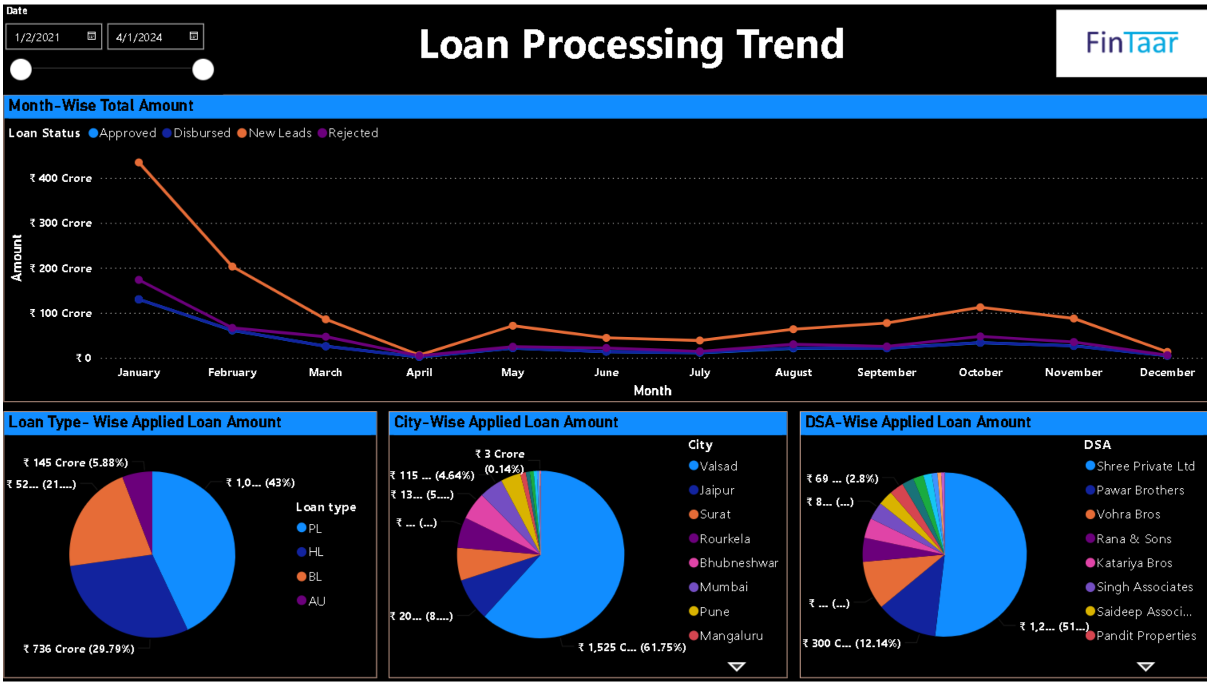


Figure 7: Loan Processing Trend Dashboard

The BI report on Loan Processing Trend from January 2021 to April 2024 provides a detailed analysis of various metrics related to loan applications. It covers month-wise loan status, showing categories such as approved, disbursed, new leads, and rejected loans. The data reveals that the amount disbursed significantly declined from January to April 2021, maintaining a lower trend throughout the rest of the year, with similar patterns observed in new leads, rejections, and approved loans. In terms of loan type, personal loans dominate the market, followed by home loans, business loans, and auto loans. City-wise analysis shows Valsad as the leading city for applied loan amounts, followed by other cities like Jaipur and Surat. DSA-wise data highlights Shree Private Ltd as the top player, along with significant contributions from Pawar Brothers, and others. Overall, the report indicates a peak in loan processing activities in early 2021, with personal and home loans being the most prominent, and Valsad along with major DSAs like Shree Private Ltd playing crucial roles.

4.4 Loan Management System

4.4.1 Portfolio Summary

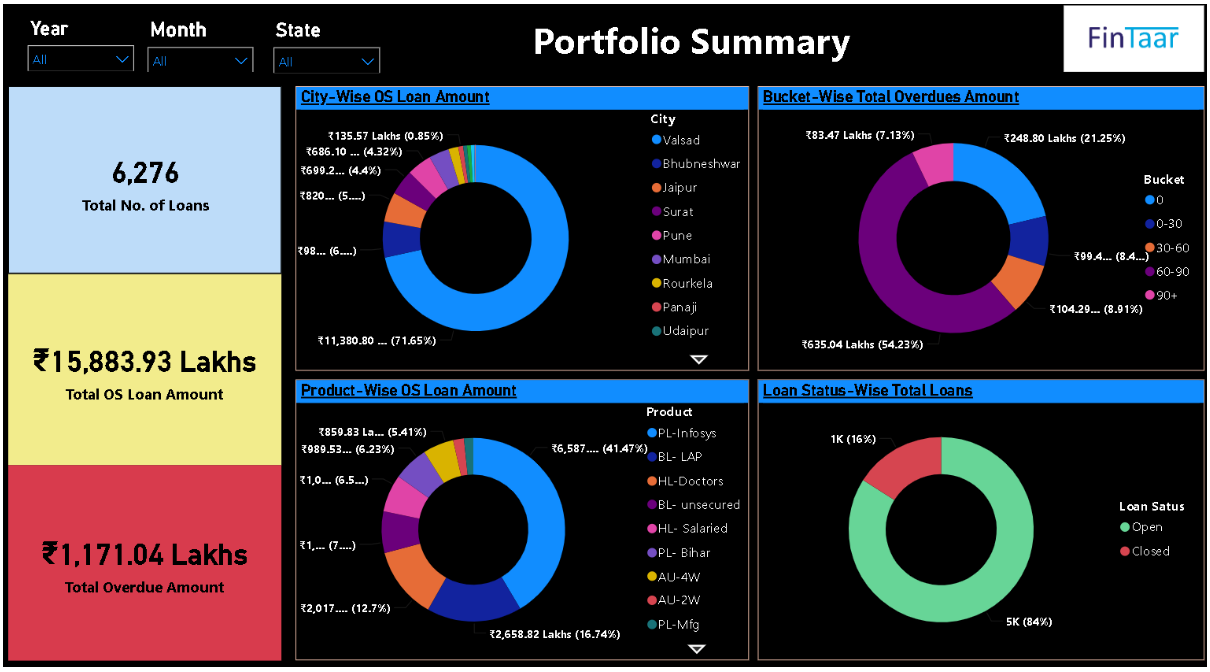


Figure 8: Portfolio Summary Dashboard

The BI report on the Portfolio Summary of a loan management system provides a comprehensive overview of disbursed loans. It indicates that a total of 6,276 loans have been disbursed, with an outstanding loan amount of ₹15,883.93 Lakhs and a total overdue amount of ₹1,171.04 Lakhs. The city-wise analysis shows that Valsad holds the highest outstanding loan amount, followed by other cities like Pune, Jaipur, and Bhubaneswar. Overdue amounts are categorized into buckets of 0-30 days, 30-60 days, 60-90 days, and 90+ days, with the 60-90 days bucket having the highest overdue amounts. Product-wise, personal loans, business loans and home loans have the largest outstanding amounts, while auto loans also make significant contribution. Additionally, the report highlights that the majority of loans are closed, with a smaller portion remaining open. This detailed breakdown helps in understanding the distribution and status of the loan portfolio, aiding in effective loan management and decision-making.

4.4.2 Portfolio Monthly Summary

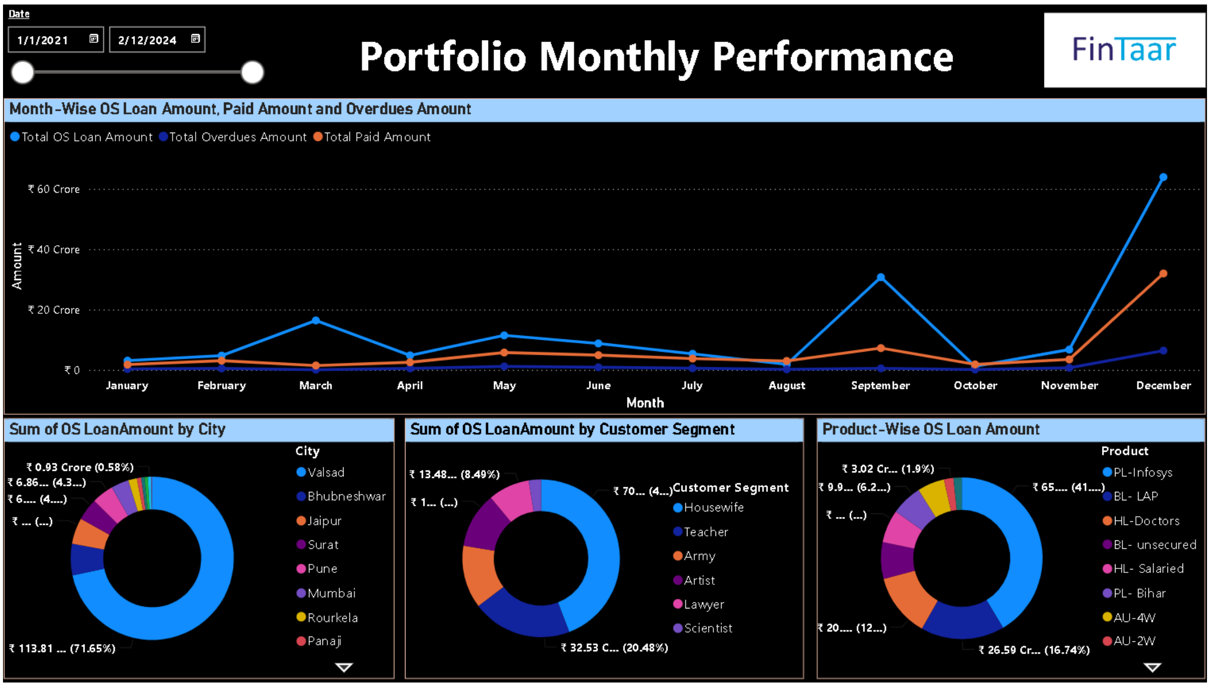
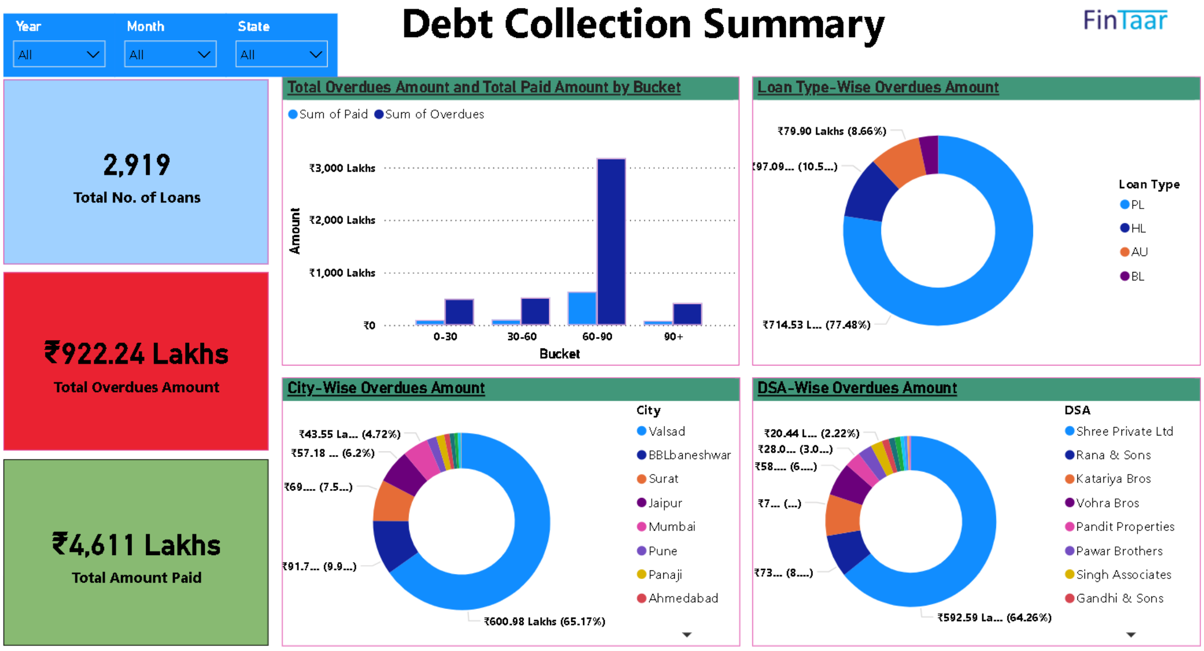


Figure 9: Portfolio Monthly Performance Dashboard

The BI report on Portfolio Monthly Performance provides a comprehensive analysis of disbursed loans, focusing on various metrics over a specified period from January 2021 to December 2024. It highlights the month-wise outstanding loan amount, paid amount, and overdue amount, showing notable peaks in loan activities during March and December. The report also breaks down the sum of outstanding loan amounts by city, customer segment, and product type. Valsad leads with the highest outstanding loan amount, followed by Bhubneshwar and other cities. In terms of customer segments, housewives have the highest outstanding loan amounts, followed by teachers and army personnel. Product-wise, personal loans for Infosys employees and salaried home loans have the highest outstanding amounts, with other loan categories like business loans and auto loans also contributing significantly. This detailed analysis helps in understanding loan distribution and performance, aiding in strategic planning and effective loan management.

4.5 Debt Collections

4.5.1 Debt Collection Summary



Debt Collection Summary

Year: All
Month: All
State: All

2,919
Total No. of Loans

₹922.24 Lakhs
Total Overdues Amount

₹4,611 Lakhs
Total Amount Paid

Figure 10: Debt Collection Summary Dashboard

The provided BI report is a comprehensive Debt Collection Summary from the Loan Management System, focusing on the overall debt collection metrics. It displays several key performance indicators, including the total number of loans, the total amount of overdues, and the total amount paid. The report breaks down the total overdues and paid amounts by different buckets, which represent various time periods. Additionally, it categorizes the overdue amounts by loan type, city, and DSA (Direct Selling Agent). This categorization helps in understanding the distribution and performance of the loans in different segments, providing insights into which areas, loan types, and DSAs have the highest overdues. The visual representation through bar charts and pie charts aids in quickly grasping the data distribution and highlights areas that may need more focused collection efforts.

4.5.2 Payment Trends

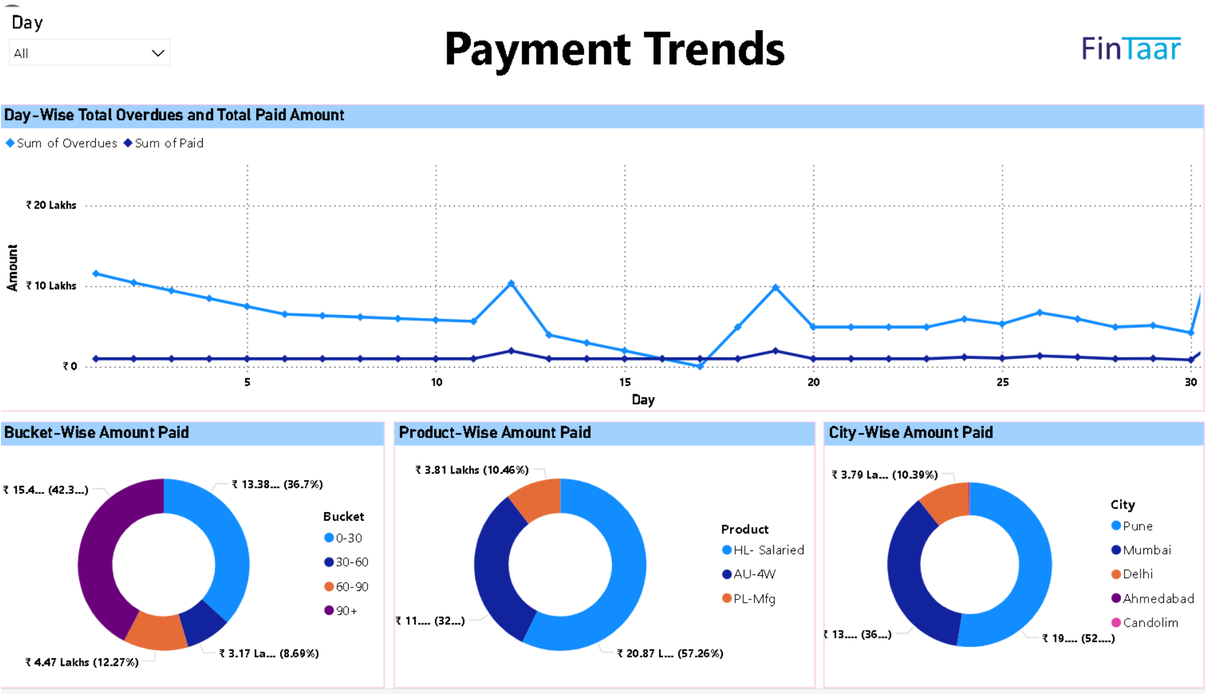


Figure 11: Payment Trends Dashboard

The provided BI report offers a detailed analysis of Payment Trends in Debt Collection. It highlights the day-wise total overdue amounts and total paid amounts, allowing for the observation of trends and patterns in payments over the course of a month. The report also categorizes the amounts paid by different buckets, which represent distinct overdue periods, showing the distribution of payments across these time frames. Additionally, it breaks down the paid amounts by product type, providing insights into how different loan products are performing in terms of collections. Lastly, the report includes a city-wise distribution of the paid amounts, giving a geographical perspective on the payment trends and indicating which cities have higher payment activities. This comprehensive overview aids in identifying areas of strength and concern in the debt collection process.

4.5.3 Collector Performance

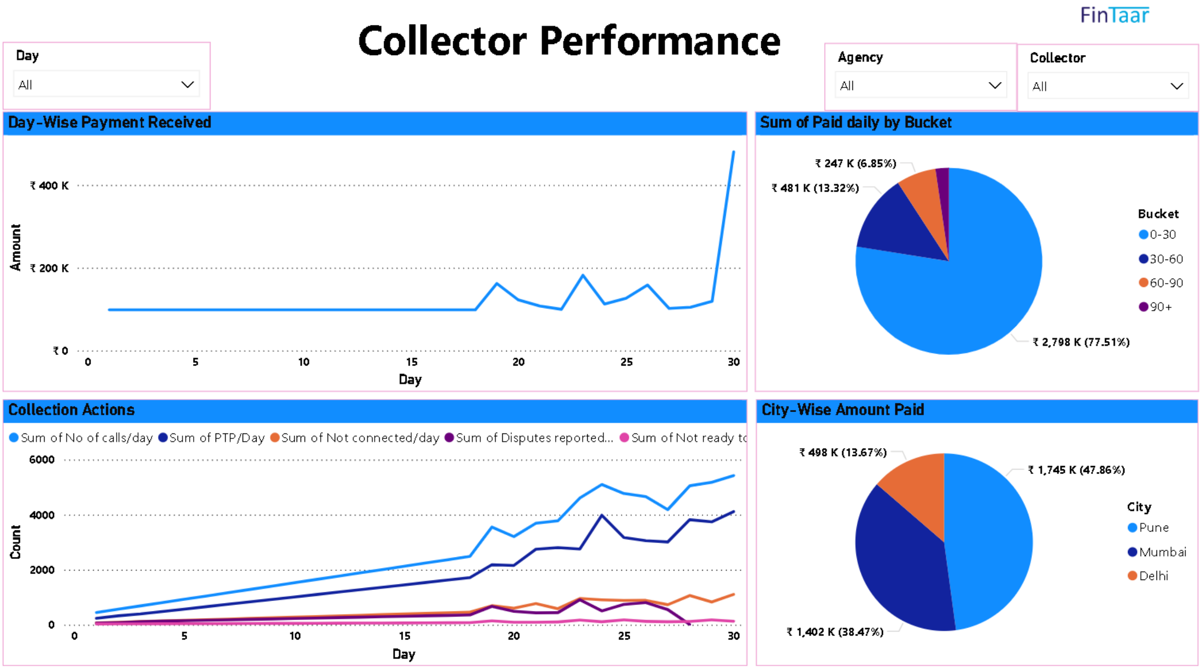


Figure 12: Collector Performance Dashboard

The BI report on Collector Performance provides an in-depth analysis of various metrics related to debt collection activities. It features the day-wise payment received, illustrating how payments have trended over a specified period. The report also breaks down the sum of the amount paid daily by different buckets, which categorize overdue periods, offering insights into the effectiveness of collections across these time frames. Additionally, the report details collection actions, including the total number of calls made per day, the sum of promises to pay (PTP) per day, the number of not connected calls per day, the disputes reported daily, and the count of those not ready to pay per day. This helps in evaluating the productivity and challenges faced by collectors. Furthermore, the city-wise amount paid is displayed, providing a geographical perspective on the collections and identifying which cities are contributing most to the payments. This comprehensive view aids in assessing and improving the overall performance of debt collectors.

5. CONCLUSION

The "Business Intelligence Dashboards for Lending as SaaS" project represents a significant advancement in empowering CXOs of lending institutions with real-time, actionable insights into their business operations. By leveraging a comprehensive technology stack comprising Power BI for dynamic and interactive dashboards, Flask for embedding these dashboards into web and mobile applications, and MySQL for robust data management, this project addresses the critical need for real-time business intelligence in the lending sector.

The integration of natural language to SQL query conversion using Google Gemini API, facilitated through a Streamlit interface, further enhances the user experience by enabling CXOs to effortlessly retrieve data insights without requiring SQL knowledge. This innovative approach ensures that decision-makers can quickly access the information they need to make informed decisions, thereby driving strategic planning and operational efficiency.

Overall, the project underscores the transformative potential of combining cutting-edge BI tools, seamless integration frameworks, and advanced natural language processing capabilities to deliver a powerful, user-friendly, and accessible business intelligence solution for the lending industry. Through rigorous testing and stabilization efforts, the system is designed to provide a reliable and stable user experience, ensuring that lending institutions can rely on this solution for their critical business needs.

REFERENCES

[1] F. Doko and I. Mishkovski, "(PDF) Advanced analytics of Big Data using Power BI: Credit Registry Use Case," *ResearchGate*, May 2020. https://www.researchgate.net/publication/350523900_Advanced_analytics_of_Big_Data_using_Power_BI_Credit_Registry_Use_Case

[2] D. Beltran, Y. Kangleon, A. Balan, and J. De Goma, "Credit Card Sales Performance Dashboard," Mar. 2021. Available: http://www.ieomsociety.org/singapore2021/papers/1186.pdf

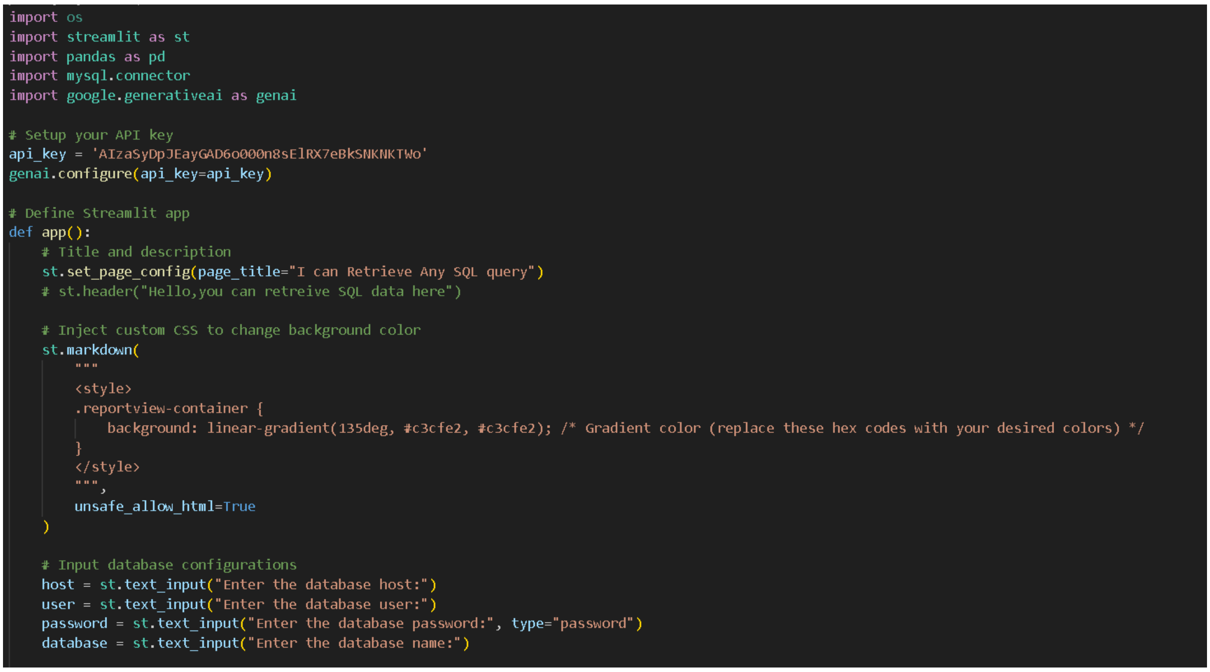
[3] S. Sahu and Sidhu, "API Monitoring Dashboard," *Juit.ac.in*, 2023, doi: http://ir.juit.ac.in:8080/jspui/jspui/handle/123456789/9819.

[4] Mario Caesar Kristantoputra and P. Subramanian, "Implementation of Business intelligence in Know Your Customer (KYC) for credit card customers' loan repayment status," *E3S web of conferences*, vol. 389, pp. 09030–09030, Jan. 2023, doi: https://doi.org/10.1051/e3sconf/202338909030.

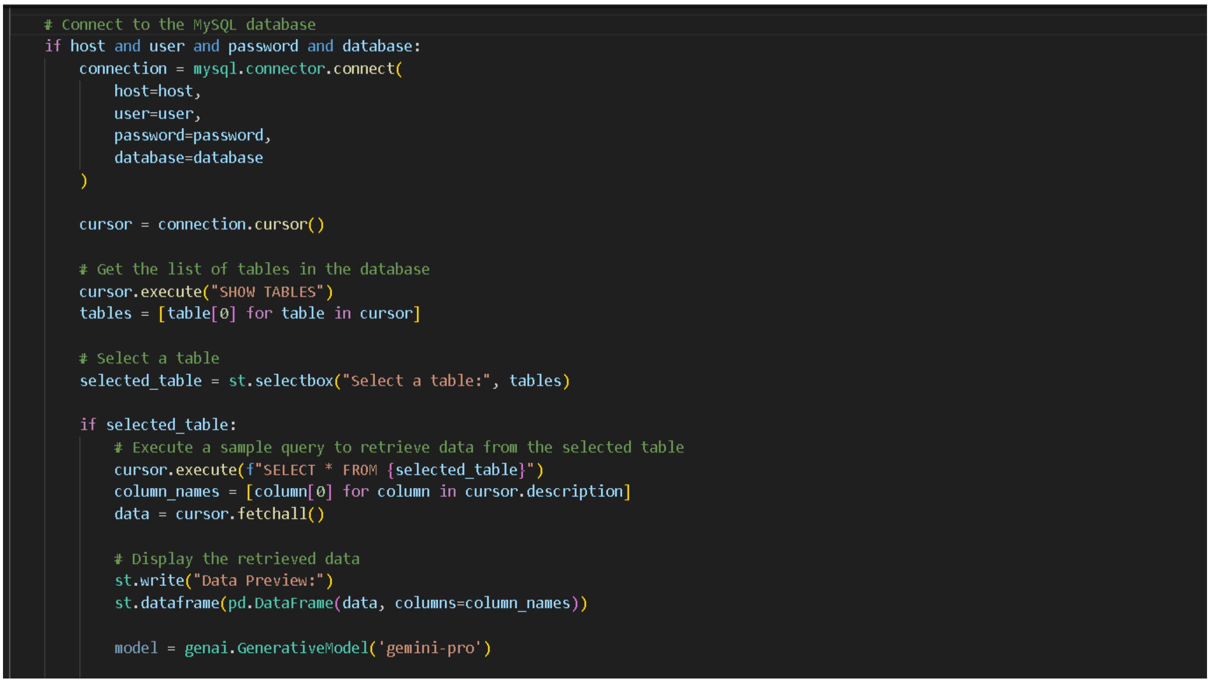
[5] S. R. Inamdar and K. Gursoy, "Cloud hosted business-data driven BI platforms," *scholarship.libraries.rutgers.edu*, 2019. https://scholarship.libraries.rutgers.edu/esploro/outputs/journalArticle/991031549933404646.

[6] M. Ashan, M. Chathura, and P. D. Asanka, "Loan Data Analysis Using Data Warehouse Techniques," *International Conference on Advanced Research in Computing (ICARC-2021),* 2021. https://www.researchgate.net/publication/353923642_Loan_Data_Analysis_Using_Data_Warehouse_Techniques.

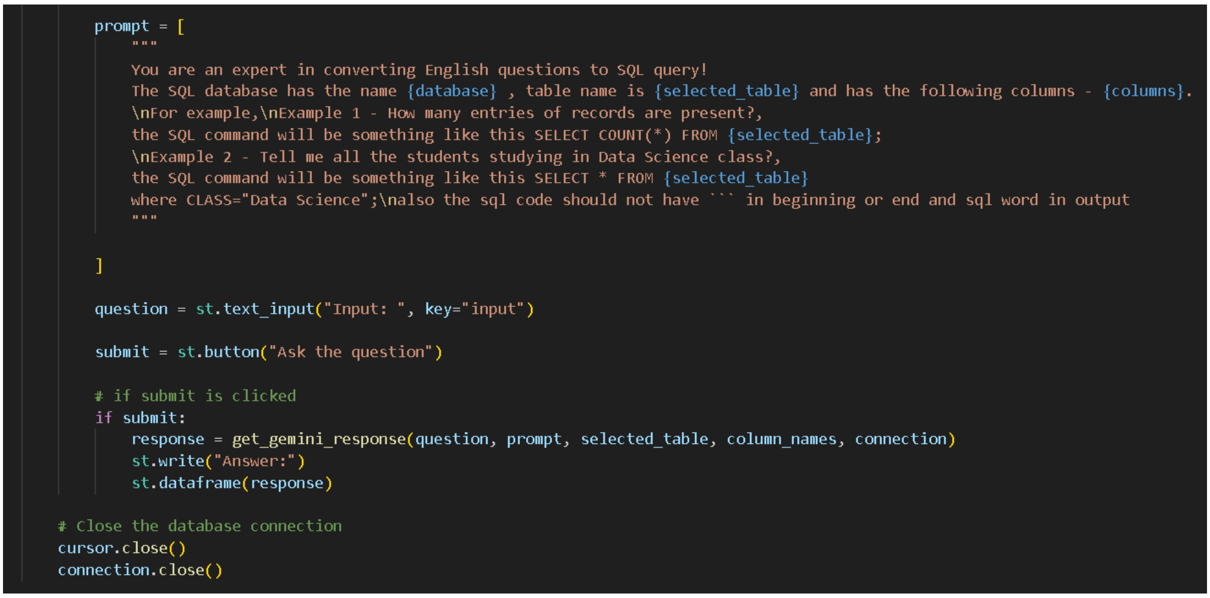
APPENDIX



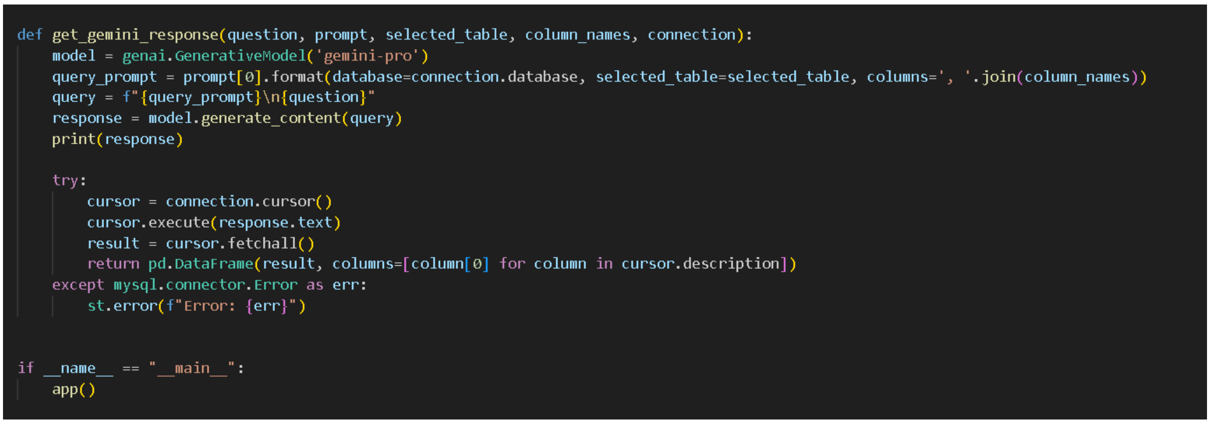
```
import os
import streamlit as st
import pandas as pd
import mysql.connector
import google.generativeai as genai

# Setup your API key
api_key = 'AIzaSyDpJEayGAD6o000n8sElRX7eBkSNKNKTWo'
genai.configure(api_key=api_key)

# Define Streamlit app
def app():
    # Title and description
    st.set_page_config(page_title="I can Retrieve Any SQL query")
    # st.header("Hello,you can retreive SQL data here")

    # Inject custom CSS to change background color
    st.markdown(
        """
        <style>
        .reportview-container {
            background: linear-gradient(135deg, #c3cfe2, #c3cfe2); /* Gradient color (replace these hex codes with your desired colors) */
        }
        </style>
        """,
        unsafe_allow_html=True
    )

    # Input database configurations
    host = st.text_input("Enter the database host:")
    user = st.text_input("Enter the database user:")
    password = st.text_input("Enter the database password:", type="password")
    database = st.text_input("Enter the database name:")
```



```
    # Connect to the MySQL database
    if host and user and password and database:
        connection = mysql.connector.connect(
            host=host,
            user=user,
            password=password,
            database=database
        )

        cursor = connection.cursor()

        # Get the list of tables in the database
        cursor.execute("SHOW TABLES")
        tables = [table[0] for table in cursor]

        # Select a table
        selected_table = st.selectbox("Select a table:", tables)

        if selected_table:
            # Execute a sample query to retrieve data from the selected table
            cursor.execute(f"SELECT * FROM {selected_table}")
            column_names = [column[0] for column in cursor.description]
            data = cursor.fetchall()

            # Display the retrieved data
            st.write("Data Preview:")
            st.dataframe(pd.DataFrame(data, columns=column_names))

            model = genai.GenerativeModel('gemini-pro')
```



```
        prompt = [
            """
            You are an expert in converting English questions to SQL query!
            The SQL database has the name {database} , table name is {selected_table} and has the following columns - {columns}.
            \nFor example,\nExample 1 - How many entries of records are present?,
            the SQL command will be something like this SELECT COUNT(*) FROM {selected_table};
            \nExample 2 - Tell me all the students studying in Data Science class?,
            the SQL command will be something like this SELECT * FROM {selected_table}
            where CLASS="Data Science";\nalso the sql code should not have ``` in beginning or end and sql word in output
            """

        ]

        question = st.text_input("Input: ", key="input")

        submit = st.button("Ask the question")

        # if submit is clicked
        if submit:
            response = get_gemini_response(question, prompt, selected_table, column_names, connection)
            st.write("Answer:")
            st.dataframe(response)

    # Close the database connection
    cursor.close()
    connection.close()
```

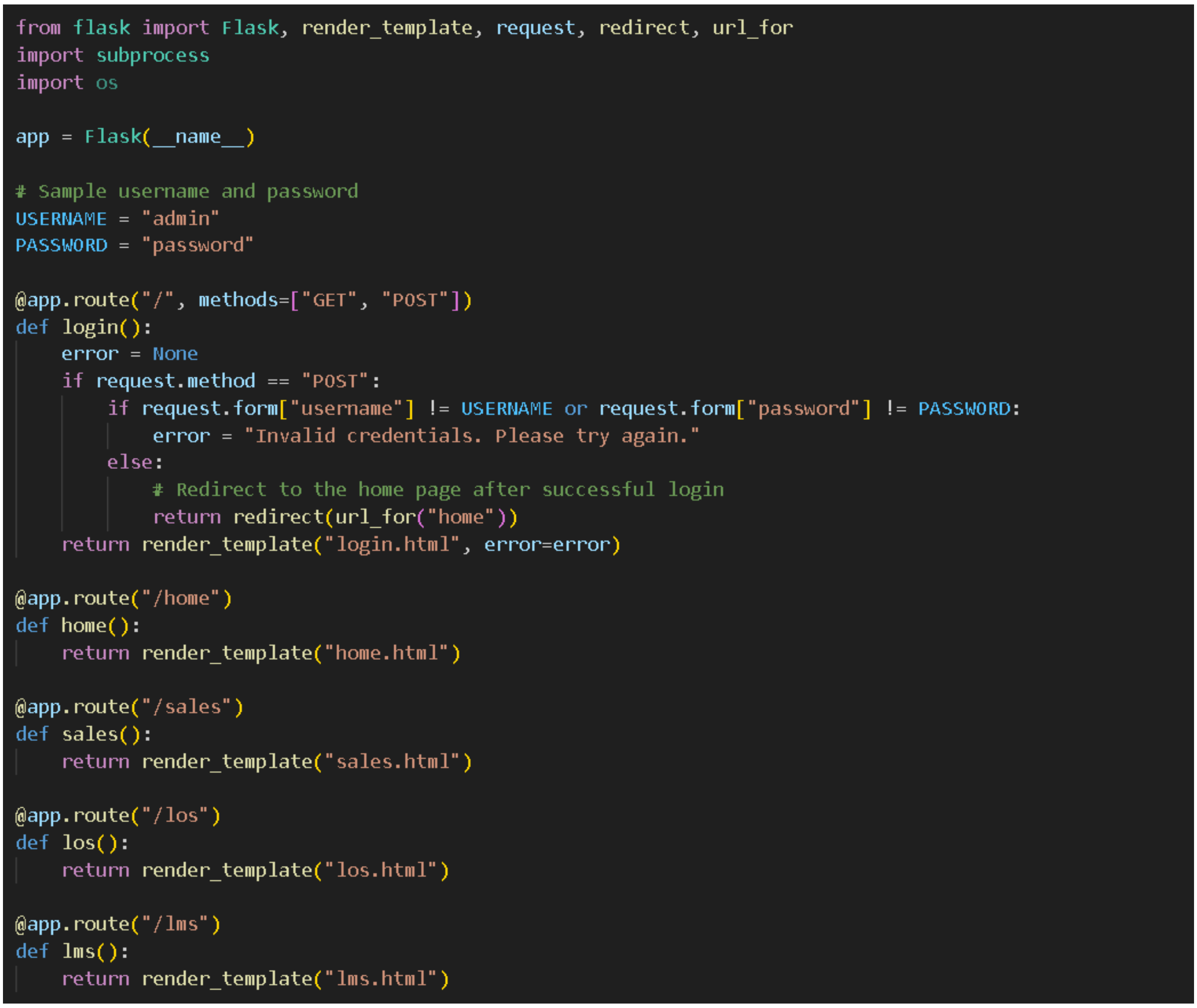


```
def get_gemini_response(question, prompt, selected_table, column_names, connection):
    model = genai.GenerativeModel('gemini-pro')
    query_prompt = prompt[0].format(database=connection.database, selected_table=selected_table, columns=', '.join(column_names))
    query = f"{query_prompt}\n{question}"
    response = model.generate_content(query)
    print(response)

    try:
        cursor = connection.cursor()
        cursor.execute(response.text)
        result = cursor.fetchall()
        return pd.DataFrame(result, columns=[column[0] for column in cursor.description])
    except mysql.connector.Error as err:
        st.error(f"Error: {err}")


if __name__ == "__main__":
    app()
```

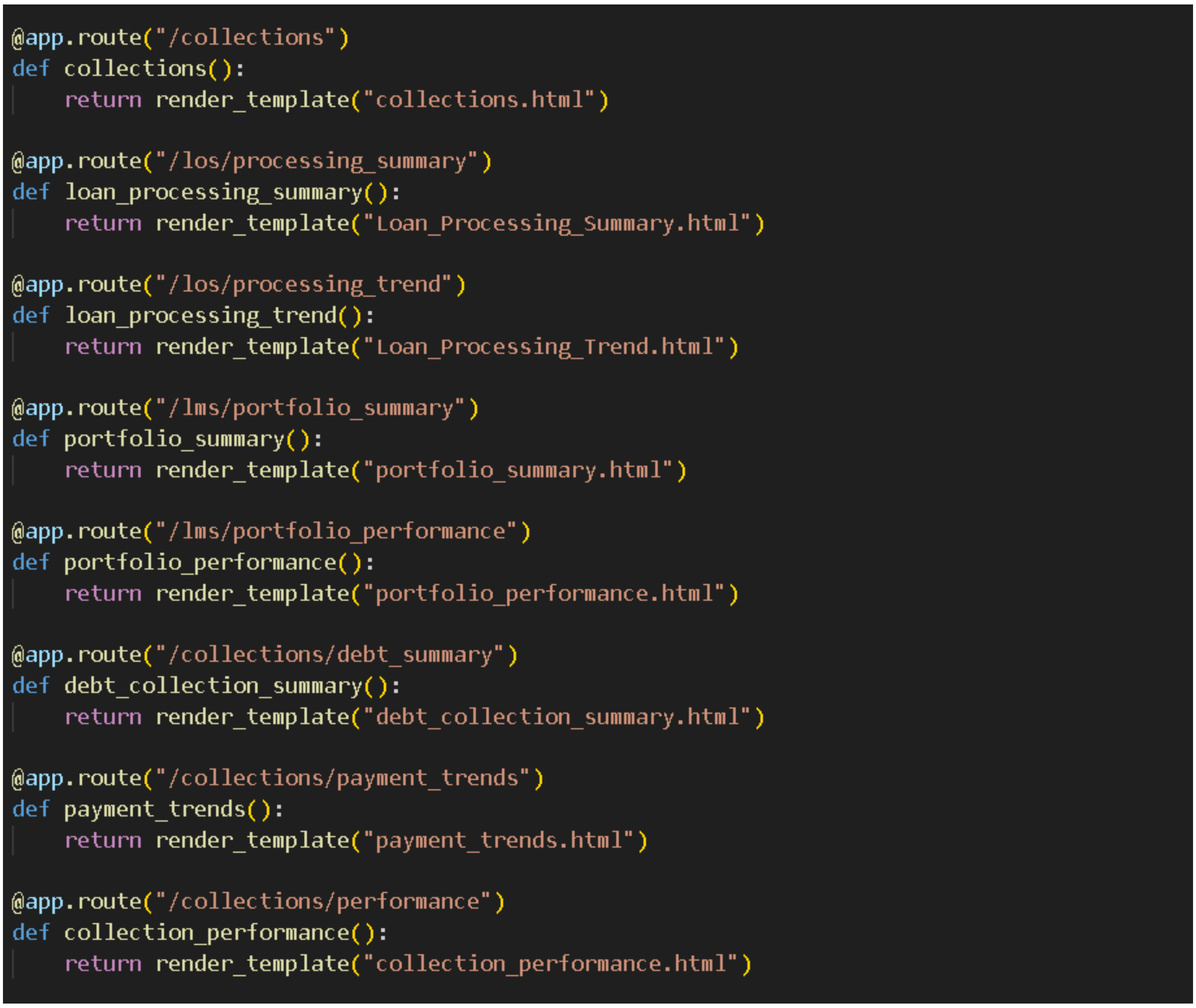
This Streamlit application connects to a MySQL database, retrieves data from selected tables, and uses Google's Generative AI model (gemini-pro) to convert English questions into SQL queries. It begins by importing necessary libraries, configuring the Google API key, and defining the main app function. The app's interface allows users to input database connection details (host, user, password, database) and establish a connection using mysql.connector. Once connected, it retrieves and displays the list of tables, enabling the user to select a table and view its data.

The app includes a custom CSS injection to modify the background color for a better visual experience. After selecting a table, the data is fetched and displayed using pandas and Streamlit's dataframe function. Users can then input English questions about the data, which are processed by the AI model to generate corresponding SQL queries. These queries are executed against the database, and the results are displayed in the app.

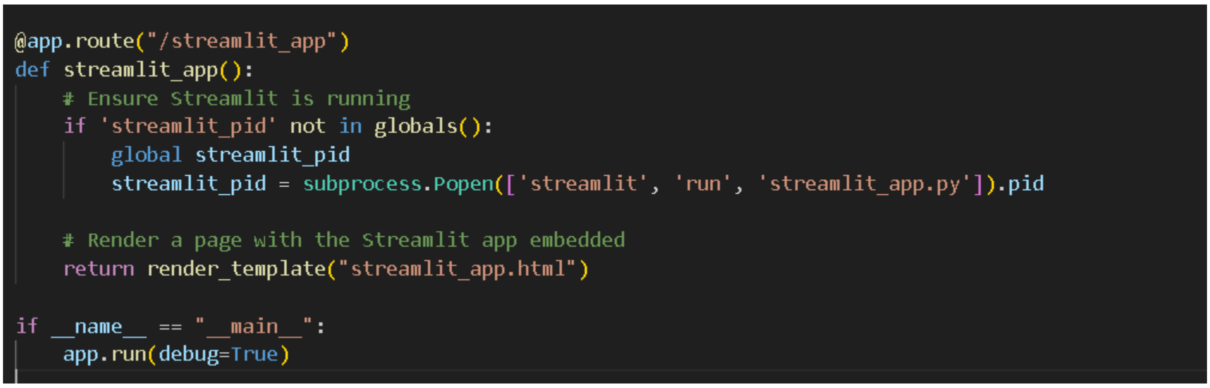
A separate function, get_gemini_response, handles the interaction with the AI model, generating the SQL query based on a formatted prompt that includes the database and table details. The generated query is executed, and the results are returned as a dataframe. The app ensures proper resource management by closing database connections after use. Overall, the application provides an intuitive interface for querying a MySQL database using natural language, leveraging the power of Google's AI.



```
from flask import Flask, render_template, request, redirect, url_for
import subprocess
import os

app = Flask(__name__)

# Sample username and password
USERNAME = "admin"
PASSWORD = "password"

@app.route("/", methods=["GET", "POST"])
def login():
    error = None
    if request.method == "POST":
        if request.form["username"] != USERNAME or request.form["password"] != PASSWORD:
            error = "Invalid credentials. Please try again."
        else:
            # Redirect to the home page after successful login
            return redirect(url_for("home"))
    return render_template("login.html", error=error)

@app.route("/home")
def home():
    return render_template("home.html")

@app.route("/sales")
def sales():
    return render_template("sales.html")

@app.route("/los")
def los():
    return render_template("los.html")

@app.route("/lms")
def lms():
    return render_template("lms.html")
```



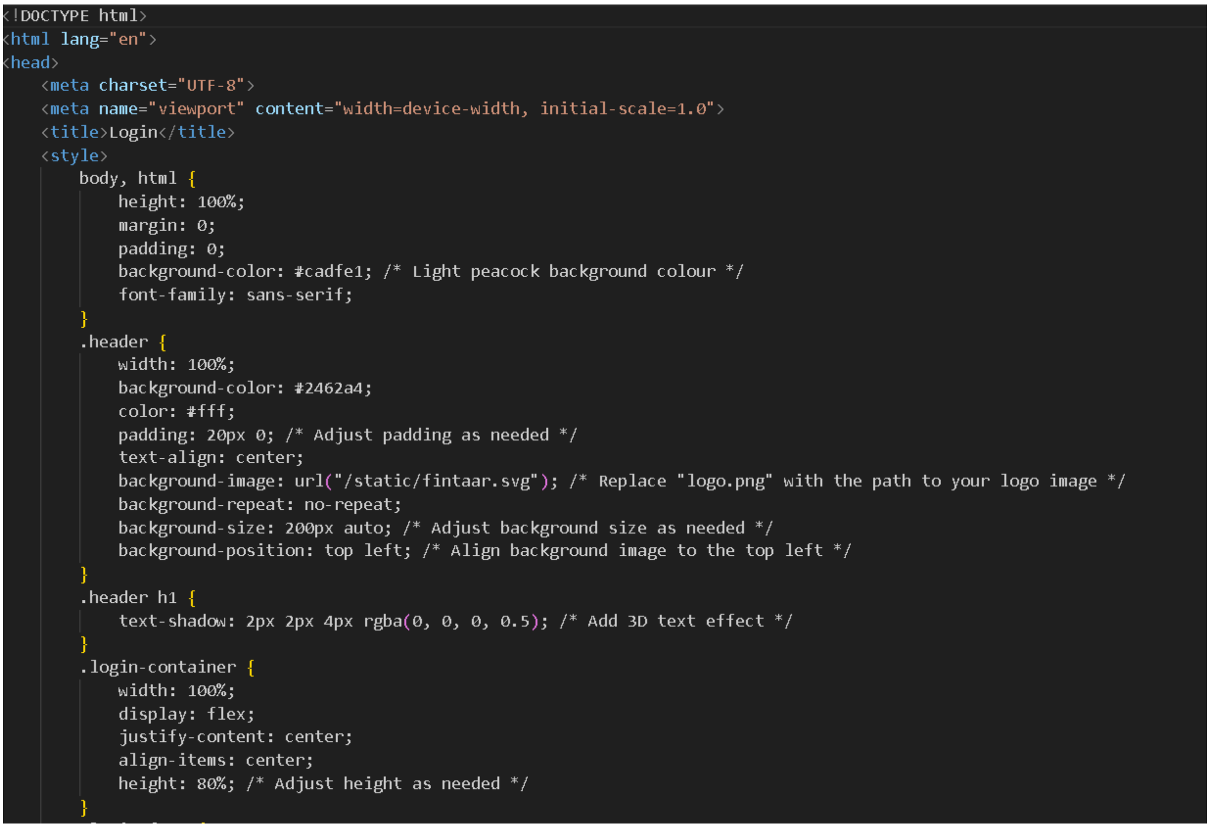
```
@app.route("/collections")
def collections():
    return render_template("collections.html")

@app.route("/los/processing_summary")
def loan_processing_summary():
    return render_template("Loan_Processing_Summary.html")

@app.route("/los/processing_trend")
def loan_processing_trend():
    return render_template("Loan_Processing_Trend.html")

@app.route("/lms/portfolio_summary")
def portfolio_summary():
    return render_template("portfolio_summary.html")

@app.route("/lms/portfolio_performance")
def portfolio_performance():
    return render_template("portfolio_performance.html")

@app.route("/collections/debt_summary")
def debt_collection_summary():
    return render_template("debt_collection_summary.html")

@app.route("/collections/payment_trends")
def payment_trends():
    return render_template("payment_trends.html")

@app.route("/collections/performance")
def collection_performance():
    return render_template("collection_performance.html")
```



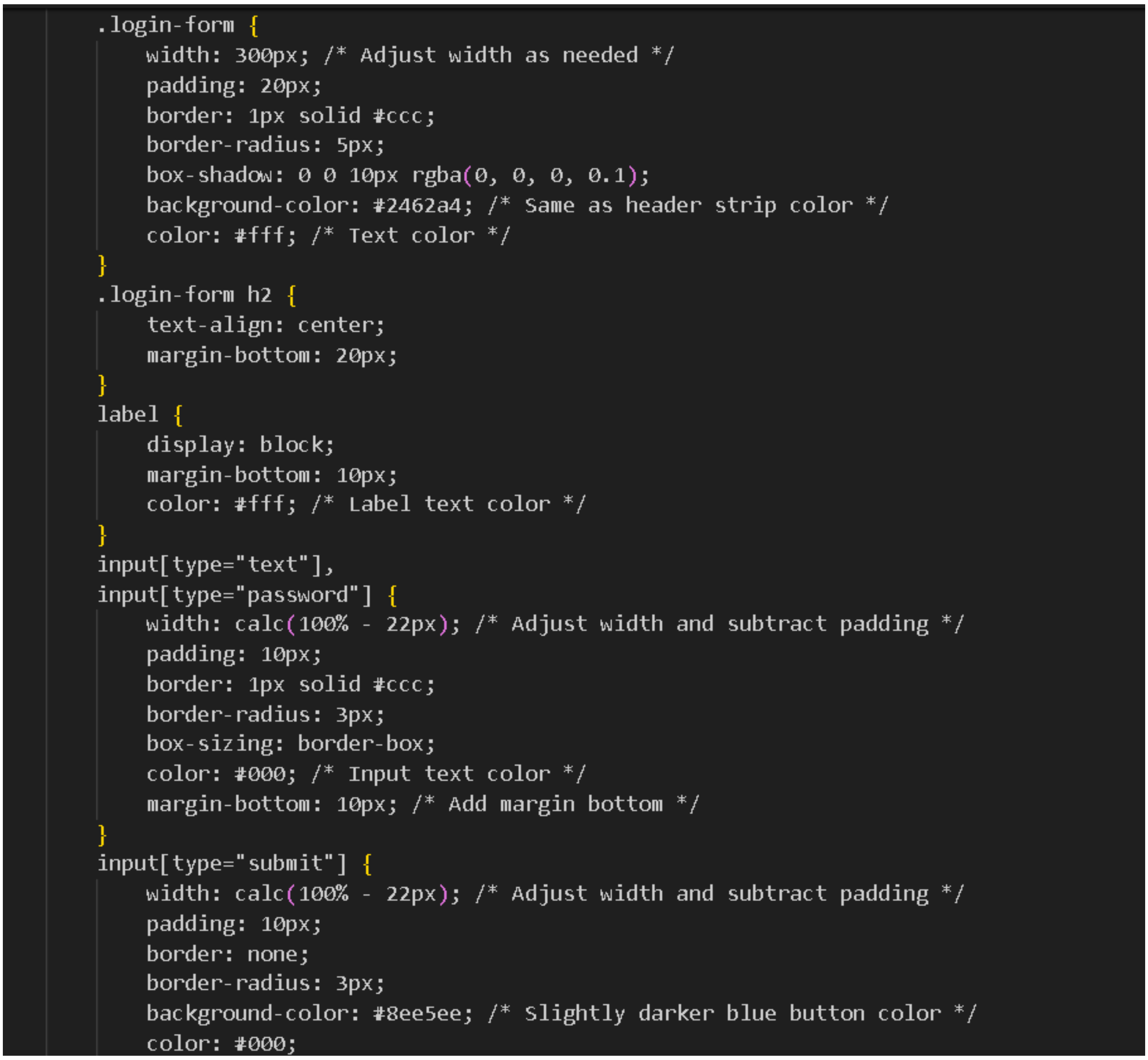
```
@app.route("/streamlit_app")
def streamlit_app():
    # Ensure Streamlit is running
    if 'streamlit_pid' not in globals():
        global streamlit_pid
        streamlit_pid = subprocess.Popen(['streamlit', 'run', 'streamlit_app.py']).pid

    # Render a page with the Streamlit app embedded
    return render_template("streamlit_app.html")

if __name__ == "__main__":
    app.run(debug=True)
```

The provided Flask main.py file serves as the backend framework for a web application designed to provide various business intelligence dashboards for the lending sector. It includes a login functionality to ensure secure access, where users must enter a valid username and password to proceed. Upon successful login, users are redirected to a home page from where they can navigate to specific dashboards tailored to different aspects of the lending business, including sales, loan origination system (LOS), loan management system (LMS), and collections. Each dashboard is linked to a specific URL route, allowing users to easily access detailed views on loan processing summaries, loan processing trends, portfolio summaries, portfolio performance, debt collection summaries, payment trends, and collection performance.

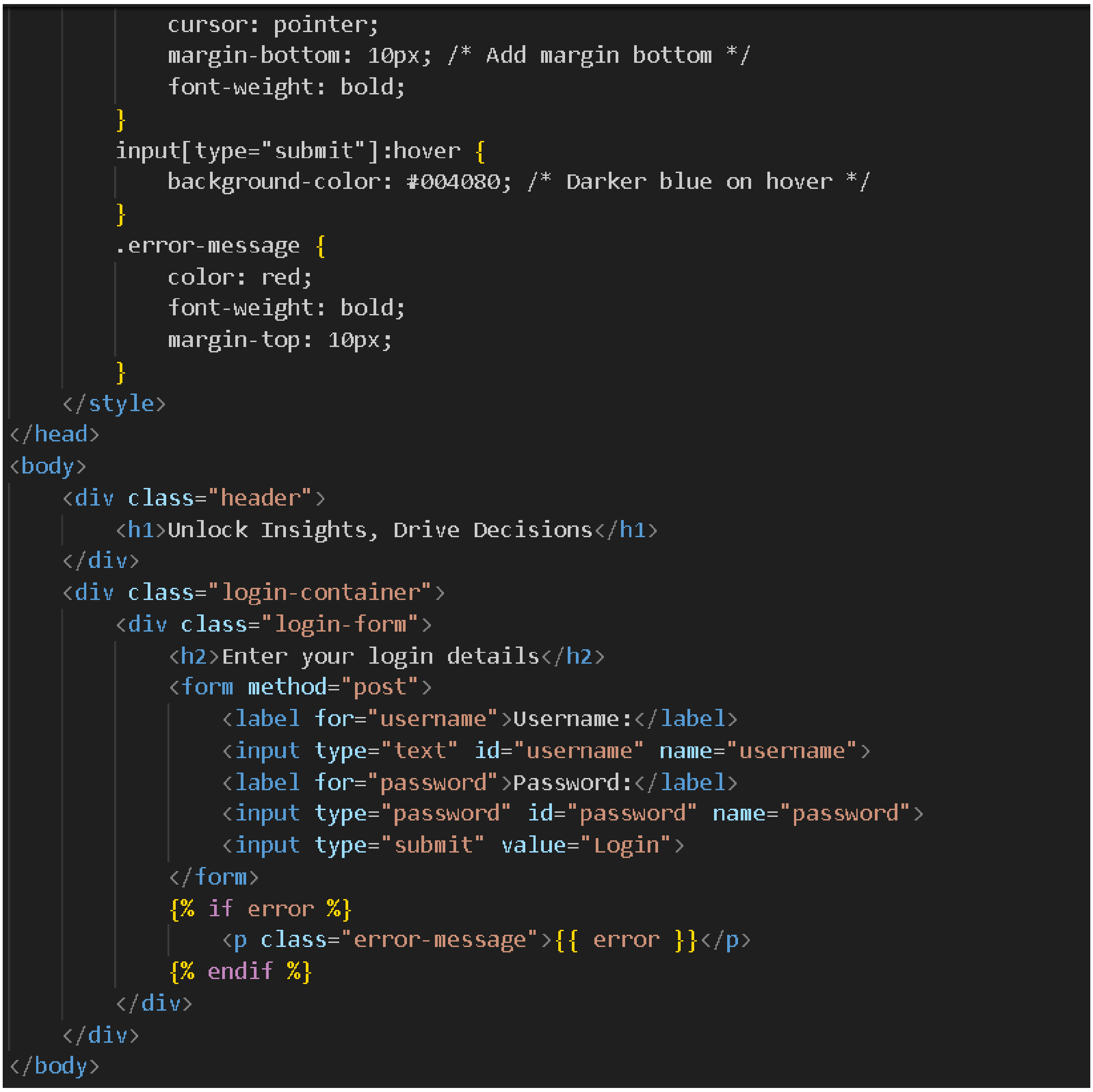
Additionally, the application integrates a Streamlit-based component for natural language to SQL query conversion, enabled by the Google Gemini API. This functionality is embedded within the Flask app and is initiated by ensuring Streamlit is running in the background. This integration allows users to enter natural language queries through the Streamlit UI, which are then converted into SQL queries and executed against a MySQL database to retrieve the required data. This setup enhances the user experience by simplifying data retrieval and enabling real-time insights without the need for SQL knowledge. The Flask routes facilitate seamless navigation and interaction within the application, providing a comprehensive BI solution for lending institutions.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <style>
        body, html {
            height: 100%;
            margin: 0;
            padding: 0;
            background-color: #cadfe1; /* Light peacock background colour */
            font-family: sans-serif;
        }
        .header {
            width: 100%;
            background-color: #2462a4;
            color: #fff;
            padding: 20px 0; /* Adjust padding as needed */
            text-align: center;
            background-image: url("/static/fintaar.svg"); /* Replace "logo.png" with the path to your logo image */
            background-repeat: no-repeat;
            background-size: 200px auto; /* Adjust background size as needed */
            background-position: top left; /* Align background image to the top left */
        }
        .header h1 {
            text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); /* Add 3D text effect */
        }
        .login-container {
            width: 100%;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 80%; /* Adjust height as needed */
        }
```



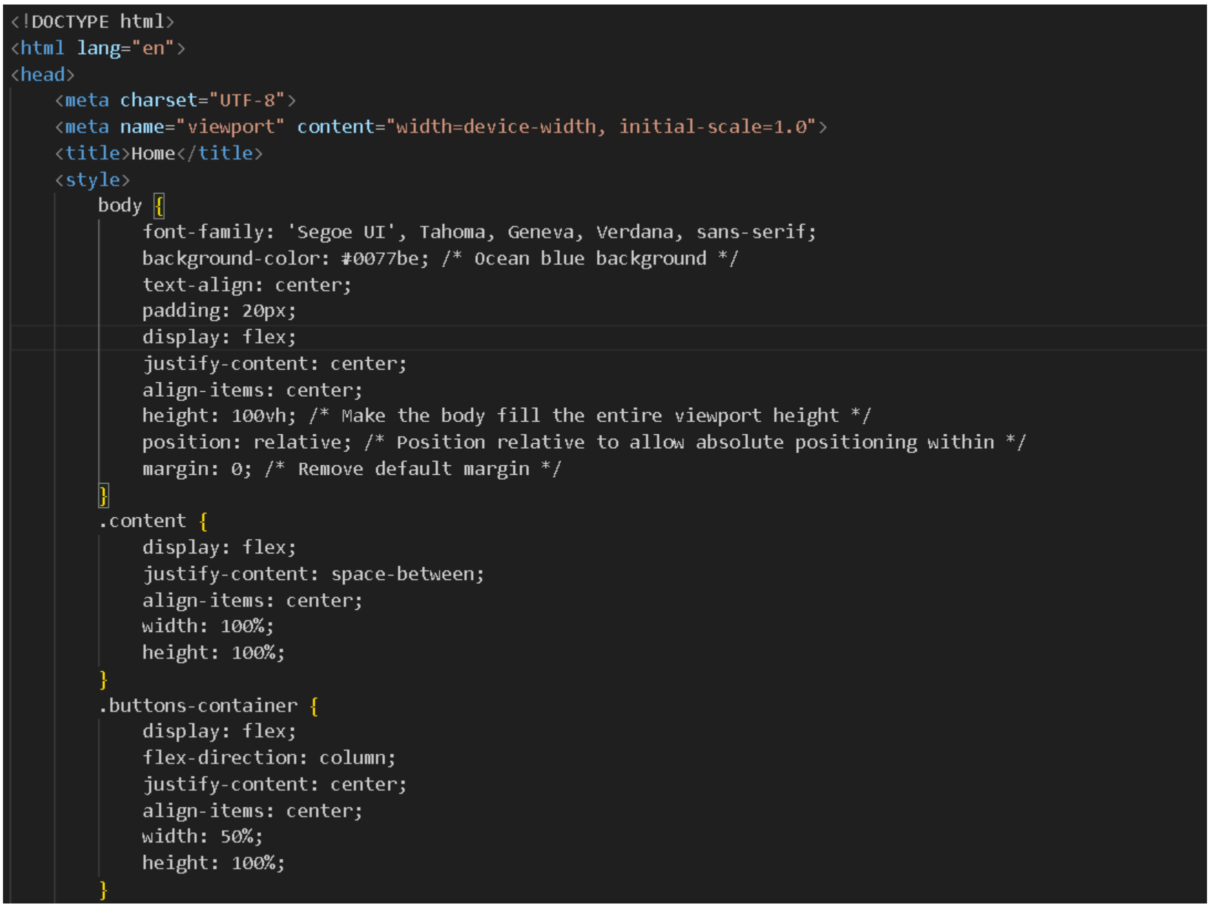
```
.login-form {
    width: 300px; /* Adjust width as needed */
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    background-color: #2462a4; /* Same as header strip color */
    color: #fff; /* Text color */
}
.login-form h2 {
    text-align: center;
    margin-bottom: 20px;
}
label {
    display: block;
    margin-bottom: 10px;
    color: #fff; /* Label text color */
}
input[type="text"],
input[type="password"] {
    width: calc(100% - 22px); /* Adjust width and subtract padding */
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
    box-sizing: border-box;
    color: #000; /* Input text color */
    margin-bottom: 10px; /* Add margin bottom */
}
input[type="submit"] {
    width: calc(100% - 22px); /* Adjust width and subtract padding */
    padding: 10px;
    border: none;
    border-radius: 3px;
    background-color: #8ee5ee; /* Slightly darker blue button color */
    color: #000;
```



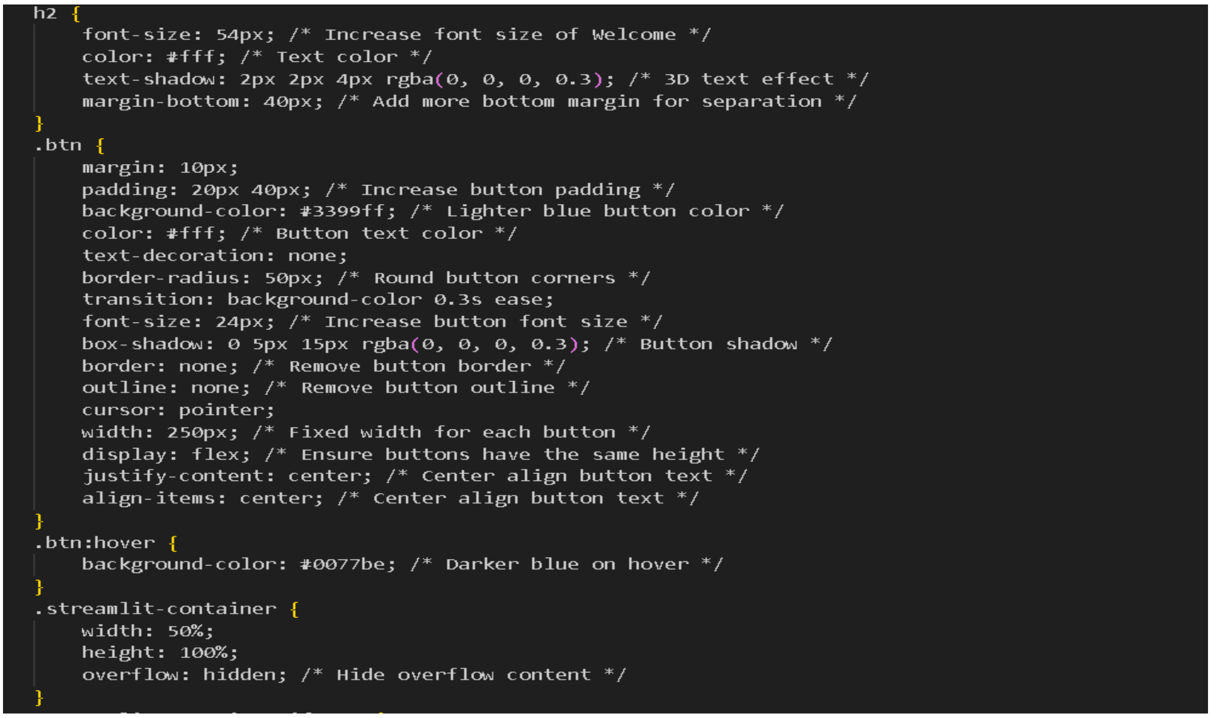
```
            cursor: pointer;
            margin-bottom: 10px; /* Add margin bottom */
            font-weight: bold;
        }
        input[type="submit"]:hover {
            background-color: #004080; /* Darker blue on hover */
        }
        .error-message {
            color: red;
            font-weight: bold;
            margin-top: 10px;
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>Unlock Insights, Drive Decisions</h1>
    </div>
    <div class="login-container">
        <div class="login-form">
            <h2>Enter your login details</h2>
            <form method="post">
                <label for="username">Username:</label>
                <input type="text" id="username" name="username">
                <label for="password">Password:</label>
                <input type="password" id="password" name="password">
                <input type="submit" value="Login">
            </form>
            {% if error %}
                <p class="error-message">{{ error }}</p>
            {% endif %}
        </div>
    </div>
</body>
```

This HTML code represents a well-designed login page with a cohesive and visually appealing style. The page is structured with a header and a login form. The header features a blue background with a logo positioned at the top left and a slogan in white text that has a 3D effect. The background color of the page is a light peacock blue, creating a calm and professional appearance. The login form, centered both vertically and horizontally within the page, is styled with a matching blue background, white text, and a border with rounded corners and a subtle shadow for a modern look. The form includes fields for entering a username and password, and a submit button that changes color when hovered over, enhancing user experience. Additionally, the form is designed to display error messages in red if there is an issue with the login credentials, ensuring clear communication with the user.

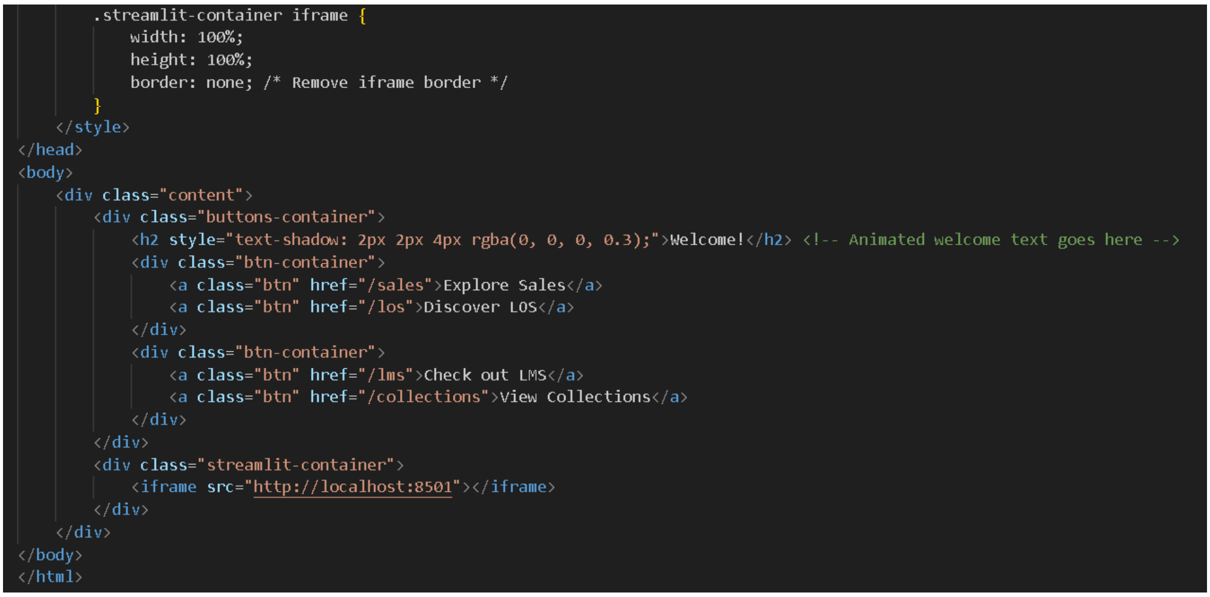
This HTML template effectively combines aesthetics and functionality, making it suitable for a user authentication interface in a web application.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            background-color: #0077be; /* Ocean blue background */
            text-align: center;
            padding: 20px;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh; /* Make the body fill the entire viewport height */
            position: relative; /* Position relative to allow absolute positioning within */
            margin: 0; /* Remove default margin */
        }
        .content {
            display: flex;
            justify-content: space-between;
            align-items: center;
            width: 100%;
            height: 100%;
        }
        .buttons-container {
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
            width: 50%;
            height: 100%;
        }
```



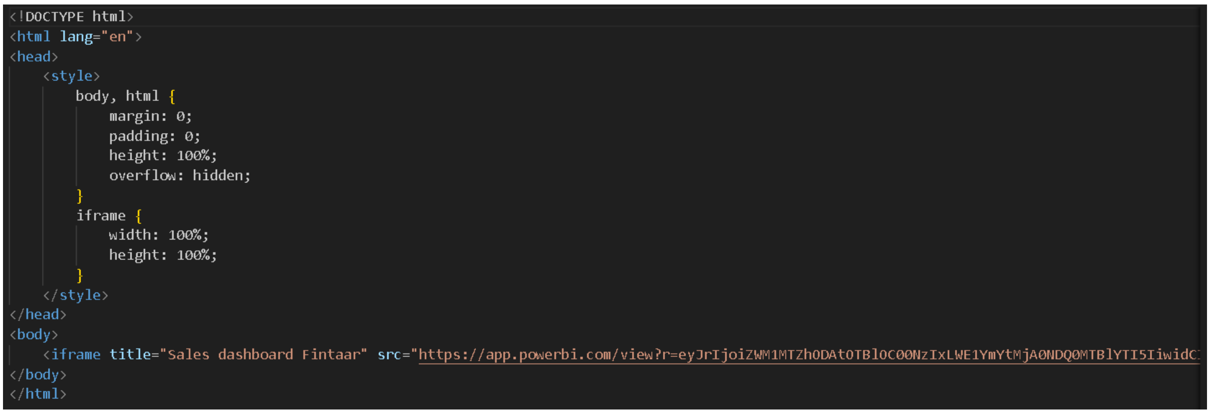
```
h2 {
    font-size: 54px; /* Increase font size of Welcome */
    color: #fff; /* Text color */
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3); /* 3D text effect */
    margin-bottom: 40px; /* Add more bottom margin for separation */
}
.btn {
    margin: 10px;
    padding: 20px 40px; /* Increase button padding */
    background-color: #3399ff; /* Lighter blue button color */
    color: #fff; /* Button text color */
    text-decoration: none;
    border-radius: 50px; /* Round button corners */
    transition: background-color 0.3s ease;
    font-size: 24px; /* Increase button font size */
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3); /* Button shadow */
    border: none; /* Remove button border */
    outline: none; /* Remove button outline */
    cursor: pointer;
    width: 250px; /* Fixed width for each button */
    display: flex; /* Ensure buttons have the same height */
    justify-content: center; /* Center align button text */
    align-items: center; /* Center align button text */
}
.btn:hover {
    background-color: #0077be; /* Darker blue on hover */
}
.streamlit-container {
    width: 50%;
    height: 100%;
    overflow: hidden; /* Hide overflow content */
}
```



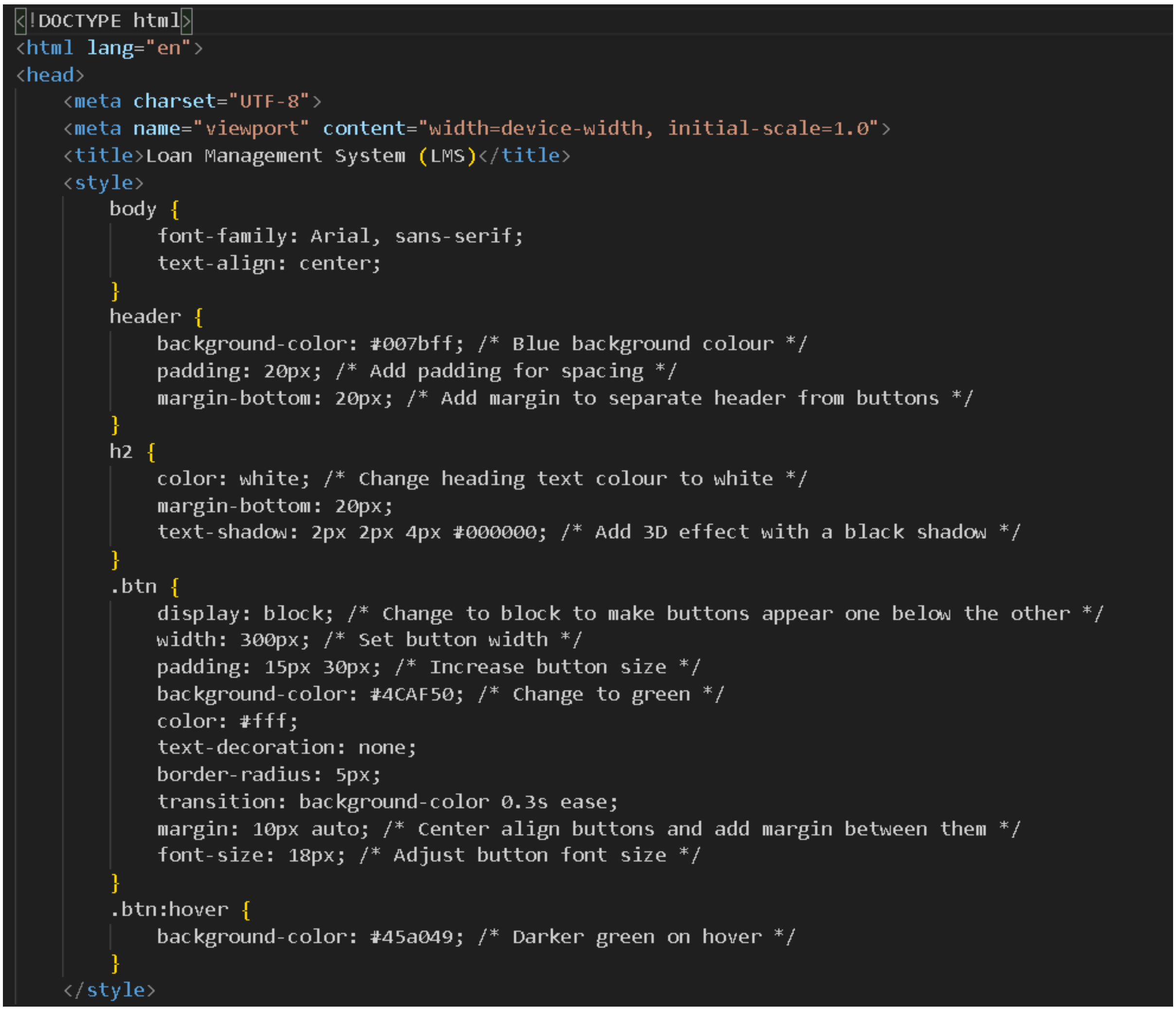
```
        .streamlit-container iframe {
            width: 100%;
            height: 100%;
            border: none; /* Remove iframe border */
        }
    </style>
</head>
<body>
    <div class="content">
        <div class="buttons-container">
            <h2 style="text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);">Welcome!</h2> <!-- Animated welcome text goes here -->
            <div class="btn-container">
                <a class="btn" href="/sales">Explore Sales</a>
                <a class="btn" href="/los">Discover LOS</a>
            </div>
            <div class="btn-container">
                <a class="btn" href="/lms">Check out LMS</a>
                <a class="btn" href="/collections">View Collections</a>
            </div>
        </div>
        <div class="streamlit-container">
            <iframe src="http://localhost:8501"></iframe>
        </div>
    </div>
</body>
</html>
```

The provided HTML code defines the structure and styling for a web page that serves as the home page of a web application. The page is designed with a clean, user-friendly interface featuring a central layout split into two main sections. On the left side, there is a vertical arrangement of buttons within a container that allows users to navigate to different parts of the application, specifically to sections related to sales, loan origination systems (LOS), loan management systems (LMS), and collections. Each button is styled with a modern look, including rounded corners, a light blue background that changes to a darker blue on hover, and a shadow effect to create a 3D appearance.

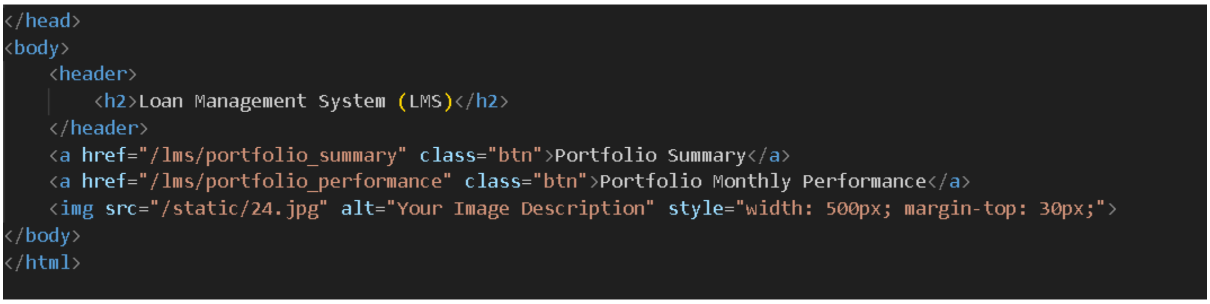
On the right side, an iframe is embedded to display the Streamlit application, which is accessible via http://localhost:8501. This Streamlit component provides an interactive interface where users can input natural language queries that are converted into SQL queries using the Google Gemini API and executed against a MySQL database, returning the results directly within the iframe. The entire page is styled to have an ocean blue background and utilizes a responsive design to ensure elements are centered and appropriately sized for the viewport. The layout effectively integrates the traditional web navigation with a dynamic Streamlit application, providing a seamless user experience for data interaction and visualization.



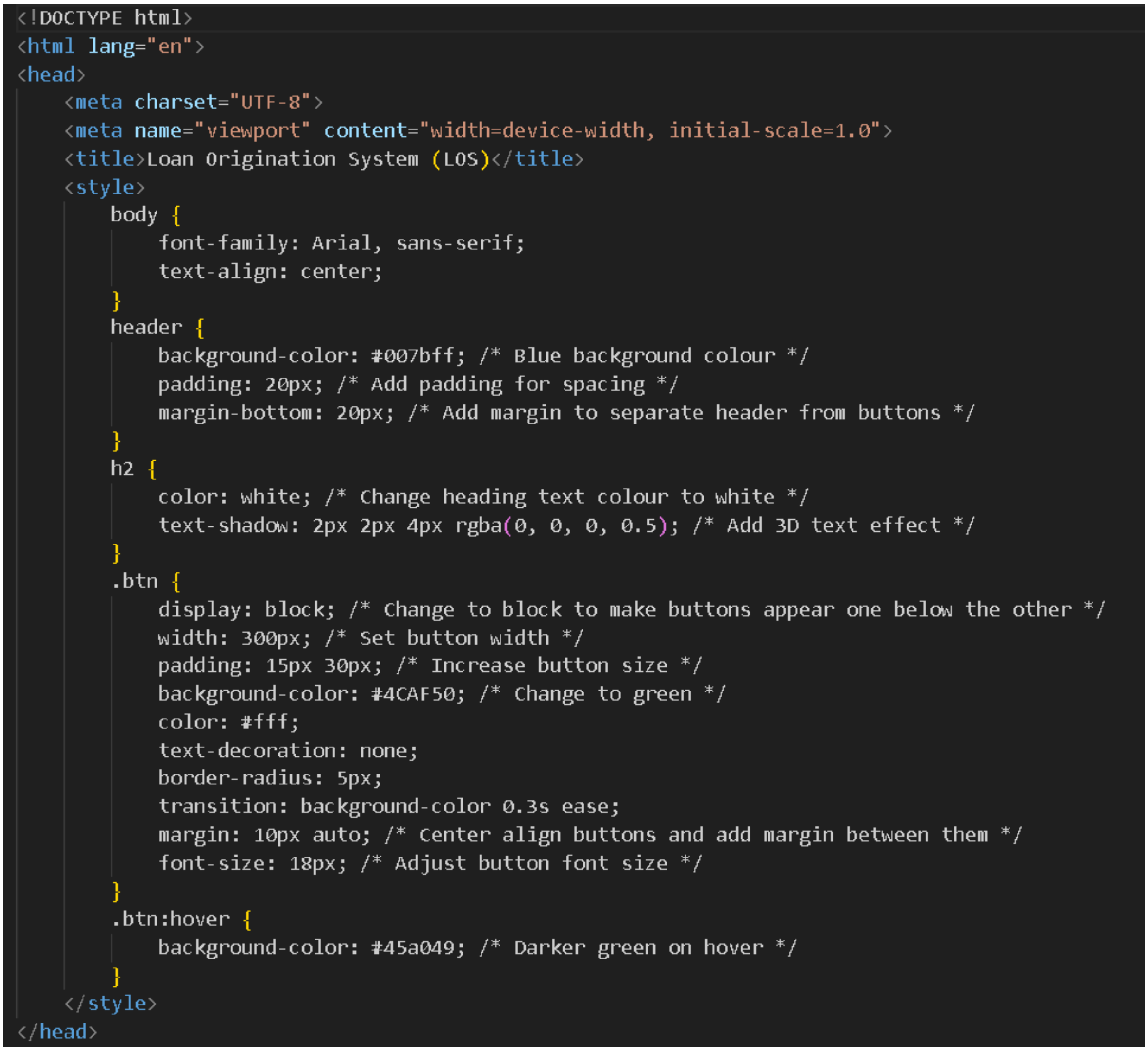
```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="Sales dashboard Fintaar" src="https://app.powerbi.com/view?r=eyJrIjoiZWM1MTZhODAtOTBlOC00NzIxLWE1YmYtMjA0NDQ0MTBlYTI5IiwidC
</body>
</html>
```



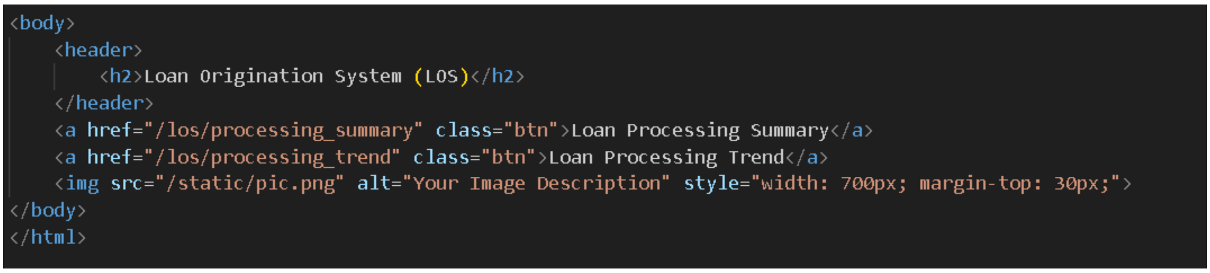
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Loan Management System (LMS)</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
        }
        header {
            background-color: #007bff; /* Blue background colour */
            padding: 20px; /* Add padding for spacing */
            margin-bottom: 20px; /* Add margin to separate header from buttons */
        }
        h2 {
            color: white; /* Change heading text colour to white */
            margin-bottom: 20px;
            text-shadow: 2px 2px 4px #000000; /* Add 3D effect with a black shadow */
        }
        .btn {
            display: block; /* Change to block to make buttons appear one below the other */
            width: 300px; /* Set button width */
            padding: 15px 30px; /* Increase button size */
            background-color: #4CAF50; /* Change to green */
            color: #fff;
            text-decoration: none;
            border-radius: 5px;
            transition: background-color 0.3s ease;
            margin: 10px auto; /* Center align buttons and add margin between them */
            font-size: 18px; /* Adjust button font size */
        }
        .btn:hover {
            background-color: #45a049; /* Darker green on hover */
        }
    </style>
```



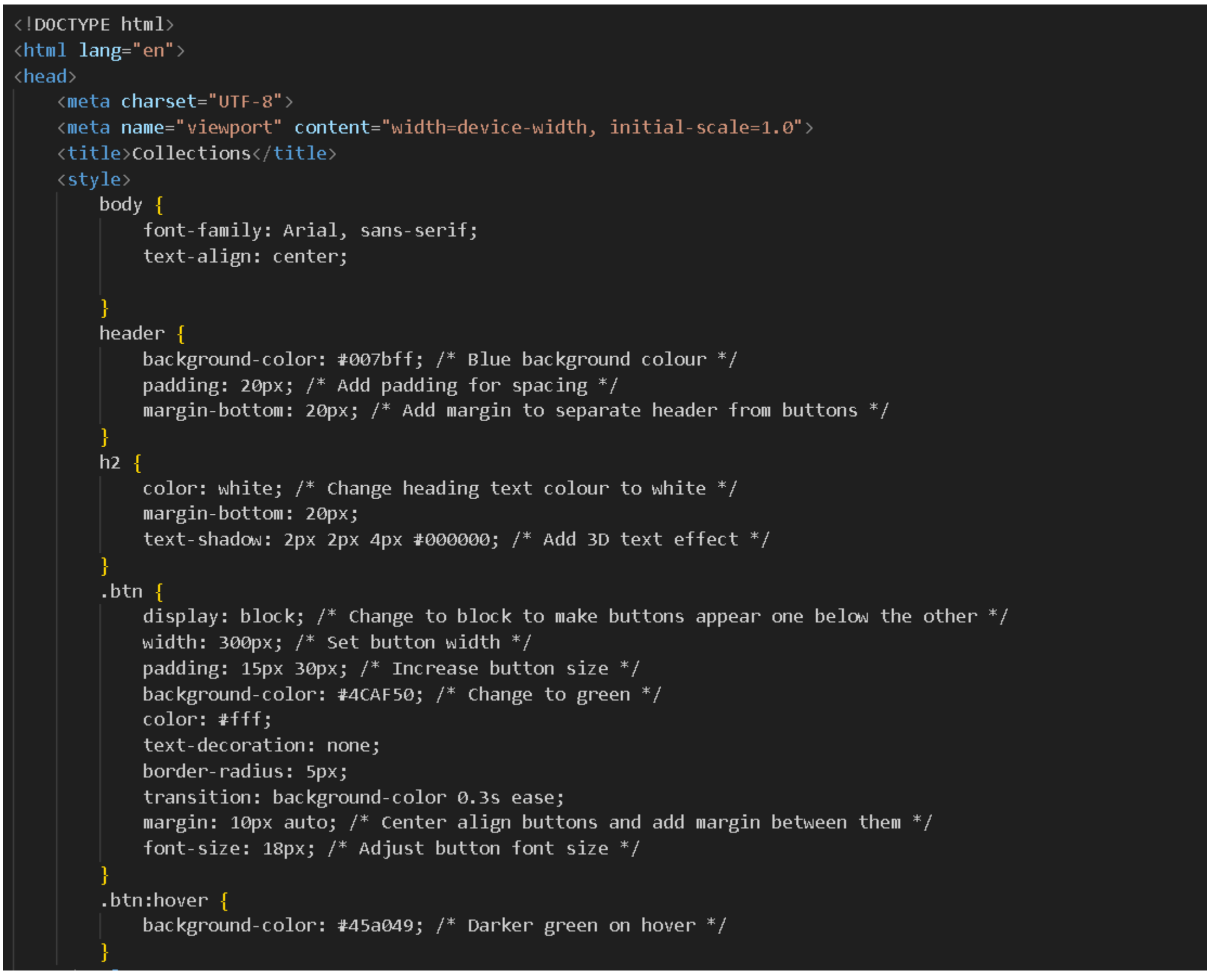
```
</head>
<body>
    <header>
        <h2>Loan Management System (LMS)</h2>
    </header>
    <a href="/lms/portfolio_summary" class="btn">Portfolio Summary</a>
    <a href="/lms/portfolio_performance" class="btn">Portfolio Monthly Performance</a>
    <img src="/static/24.jpg" alt="Your Image Description" style="width: 500px; margin-top: 30px;">
</body>
</html>
```



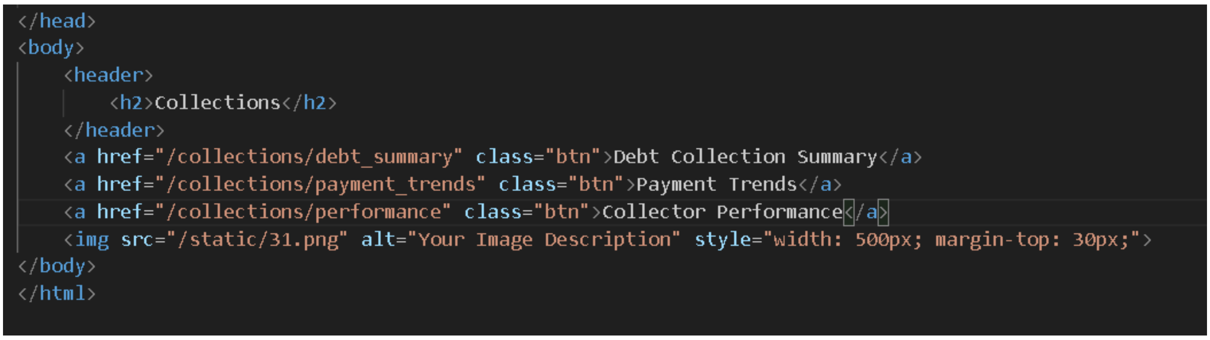
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Loan Origination System (LOS)</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
        }
        header {
            background-color: #007bff; /* Blue background colour */
            padding: 20px; /* Add padding for spacing */
            margin-bottom: 20px; /* Add margin to separate header from buttons */
        }
        h2 {
            color: white; /* Change heading text colour to white */
            text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5); /* Add 3D text effect */
        }
        .btn {
            display: block; /* Change to block to make buttons appear one below the other */
            width: 300px; /* Set button width */
            padding: 15px 30px; /* Increase button size */
            background-color: #4CAF50; /* Change to green */
            color: #fff;
            text-decoration: none;
            border-radius: 5px;
            transition: background-color 0.3s ease;
            margin: 10px auto; /* Center align buttons and add margin between them */
            font-size: 18px; /* Adjust button font size */
        }
        .btn:hover {
            background-color: #45a049; /* Darker green on hover */
        }
    </style>
</head>
```



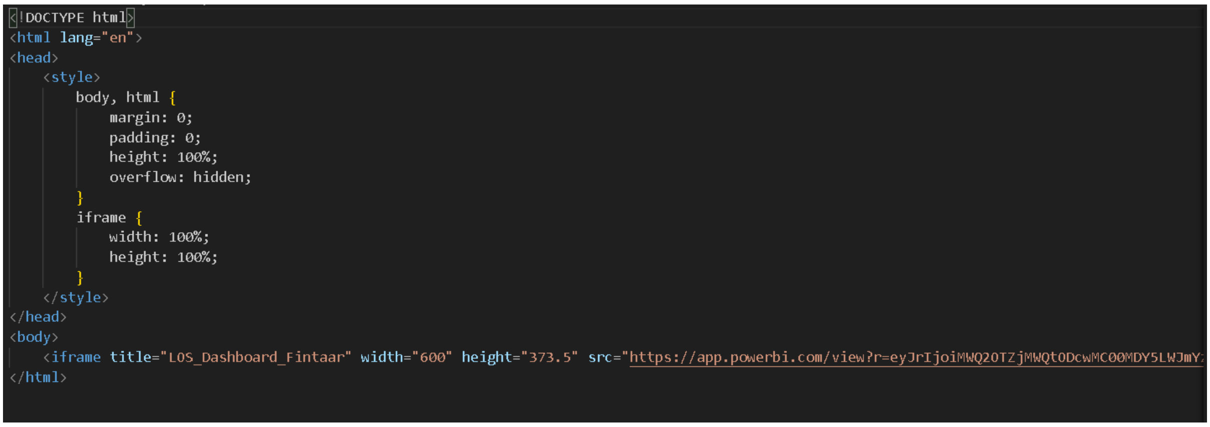
```
<body>
    <header>
        <h2>Loan Origination System (LOS)</h2>
    </header>
    <a href="/los/processing_summary" class="btn">Loan Processing Summary</a>
    <a href="/los/processing_trend" class="btn">Loan Processing Trend</a>
    <img src="/static/pic.png" alt="Your Image Description" style="width: 700px; margin-top: 30px;">
</body>
</html>
```



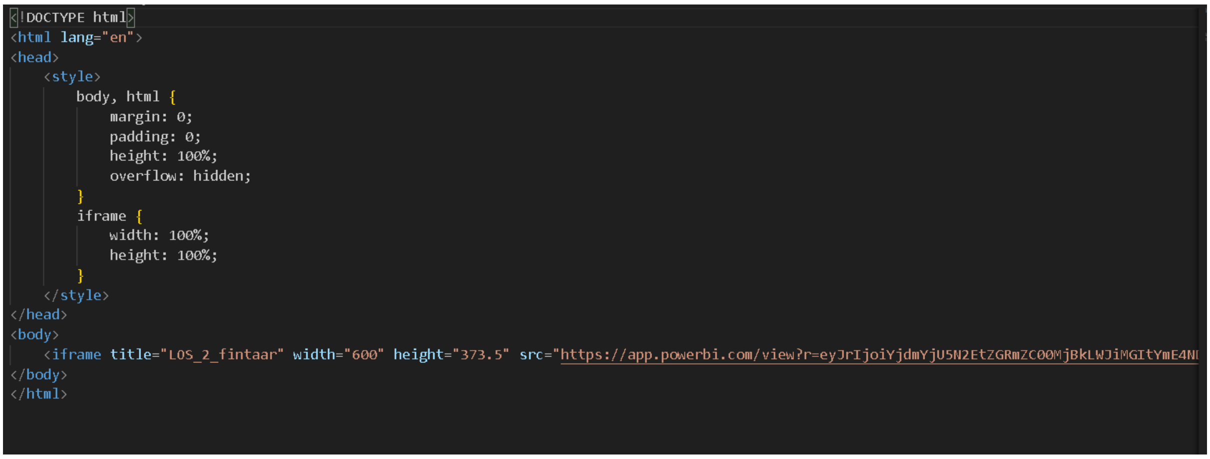
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Collections</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;

        }
        header {
            background-color: #007bff; /* Blue background colour */
            padding: 20px; /* Add padding for spacing */
            margin-bottom: 20px; /* Add margin to separate header from buttons */
        }
        h2 {
            color: white; /* Change heading text colour to white */
            margin-bottom: 20px;
            text-shadow: 2px 2px 4px #000000; /* Add 3D text effect */
        }
        .btn {
            display: block; /* Change to block to make buttons appear one below the other */
            width: 300px; /* Set button width */
            padding: 15px 30px; /* Increase button size */
            background-color: #4CAF50; /* Change to green */
            color: #fff;
            text-decoration: none;
            border-radius: 5px;
            transition: background-color 0.3s ease;
            margin: 10px auto; /* Center align buttons and add margin between them */
            font-size: 18px; /* Adjust button font size */
        }
        .btn:hover {
            background-color: #45a049; /* Darker green on hover */
        }
```



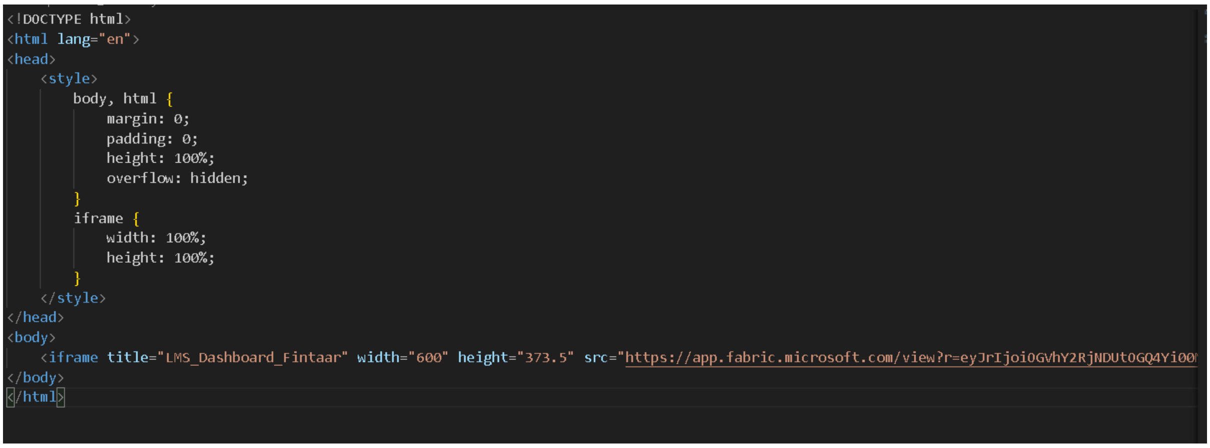
```
</head>
<body>
    <header>
        <h2>Collections</h2>
    </header>
    <a href="/collections/debt_summary" class="btn">Debt Collection Summary</a>
    <a href="/collections/payment_trends" class="btn">Payment Trends</a>
    <a href="/collections/performance" class="btn">Collector Performance</a>
    <img src="/static/31.png" alt="Your Image Description" style="width: 500px; margin-top: 30px;">
</body>
</html>
```



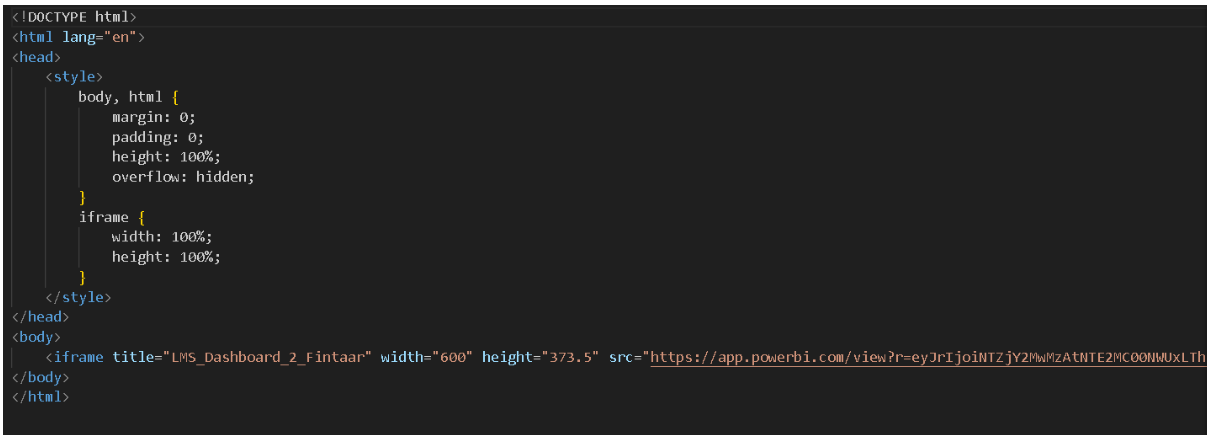
```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="LOS_Dashboard_Fintaar" width="600" height="373.5" src="https://app.powerbi.com/view?r=eyJrIjoiMWQ2OTZjMWQtODcwMC00MDY5LWJmY
</html>
```



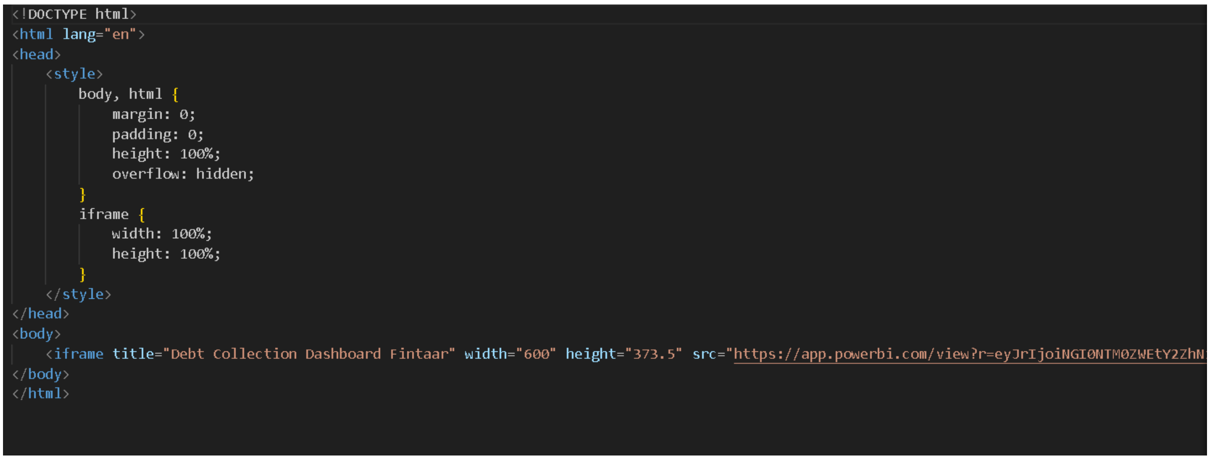
```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="LOS_2_fintaar" width="600" height="373.5" src="https://app.powerbi.com/view?r=eyJrIjoiYjdmYjU5N2EtZGRmZC00MjBkLWJiMGItYmE4N
</body>
</html>
```



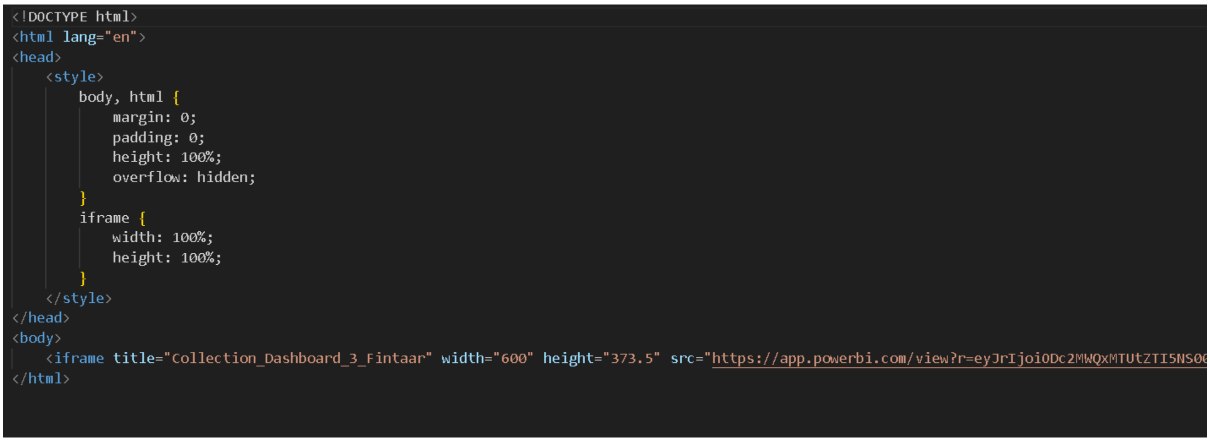
```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="LMS_Dashboard_Fintaar" width="600" height="373.5" src="https://app.fabric.microsoft.com/view?r=eyJrIjoiOGVhY2RjNDUtOGQ4Yi00
</body>
</html>
```



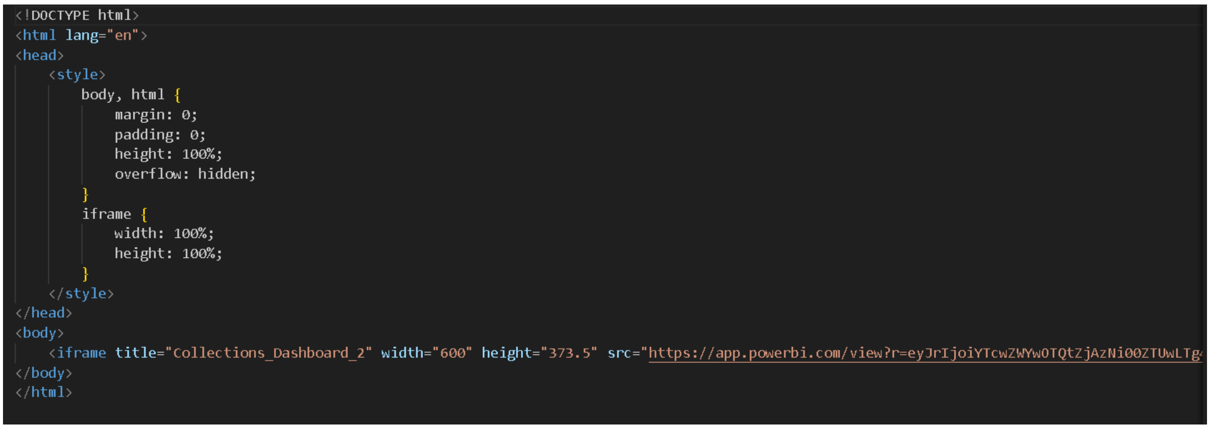
```
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="LMS_Dashboard_2_Fintaar" width="600" height="373.5" src="https://app.powerbi.com/view?r=eyJrIjoiNTZjY2MwMzAtNTE2MC00NWUxLTh
</body>
</html>
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="Debt Collection Dashboard Fintaar" width="600" height="373.5" src="https://app.powerbi.com/view?r=eyJrIjoiNGI0NTM0ZWEtY2ZhN
</body>
</html>
```

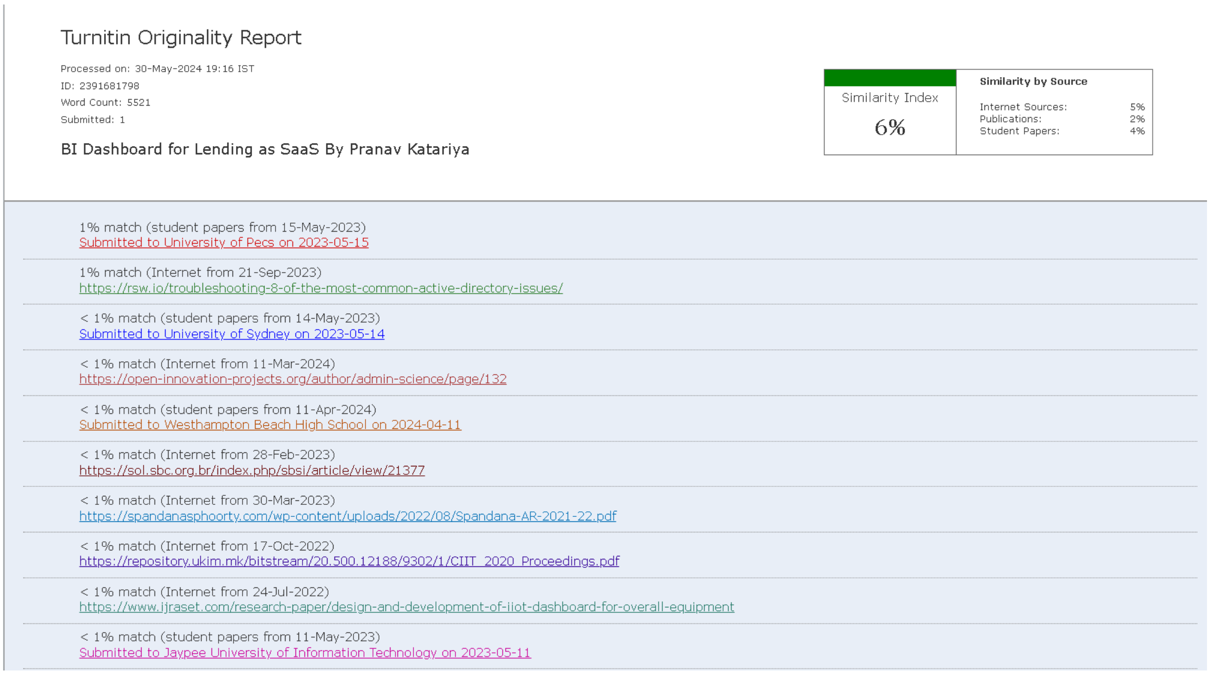


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="Collection_Dashboard_3_Fintaar" width="600" height="373.5" src="https://app.powerbi.com/view?r=eyJrIjoiODc2MWQxMTUtZTI5NS0
</html>
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            height: 100%;
            overflow: hidden;
        }
        iframe {
            width: 100%;
            height: 100%;
        }
    </style>
</head>
<body>
    <iframe title="Collections_Dashboard_2" width="600" height="373.5" src="https://app.powerbi.com/view?r=eyJrIjoiYTcwZWYwOTQtZjAzNi00ZTUwLTg
</body>
</html>
```

In this project, Power BI dashboards have been integrated into a web application through the creation of distinct templates for each dashboard. This approach ensures that each dashboard is presented in a tailored format, optimizing the user experience and facilitating easy navigation and interaction with the data. By embedding the Power BI dashboards into specific templates, the visual appeal and functionality of the application are enhanced, providing users with seamless access to various data insights and reports. This method also allows for better customization and maintenance of the dashboards, ensuring that each one can be updated or modified independently to meet the evolving needs of the users.



Turnitin Originality Report

Processed on: 30-May-2024 19:16 IST
ID: 2391681798
Word Count: 5521
Submitted: 1

BI Dashboard for Lending as SaaS By Pranav Katariya

Similarity Index

6%

Similarity by Source

Internet Sources:	5%
Publications:	2%
Student Papers:	4%

1% match (student papers from 15-May-2023)
Submitted to University of Pecs on 2023-05-15

1% match (Internet from 21-Sep-2023)
https://rsw.io/troubleshooting-8-of-the-most-common-active-directory-issues/

< 1% match (student papers from 14-May-2023)
Submitted to University of Sydney on 2023-05-14

< 1% match (Internet from 11-Mar-2024)
https://open-innovation-projects.org/author/admin-science/page/132

< 1% match (student papers from 11-Apr-2024)
Submitted to Westhampton Beach High School on 2024-04-11

< 1% match (Internet from 28-Feb-2023)
https://sol.sbc.org.br/index.php/sbsi/article/view/21377

< 1% match (Internet from 30-Mar-2023)
https://spandanasphoorty.com/wp-content/uploads/2022/08/Spandana-AR-2021-22.pdf

< 1% match (Internet from 17-Oct-2022)
https://repository.ukim.mk/bitstream/20.500.12188/9302/1/CIIT_2020_Proceedings.pdf

< 1% match (Internet from 24-Jul-2022)
https://www.ijraset.com/research-paper/design-and-development-of-iiot-dashboard-for-overall-equipment

< 1% match (student papers from 11-May-2023)
Submitted to Jaypee University of Information Technology on 2023-05-11

Proforma 4

Undertaking from the PG student while submitting his/her final dissertation to his respective institute

Ref. No. _________

I , the following student

Sr. No.	**Sequence of students names on a dissertation**	**Students name**	**Name of the Institute & Place**	**Email & Mobile**
1.	First Author	Pranav Katariya	SIG	Email:22070243026@sig.ac.in Mobile:8308366992

Note: Put additional rows in case of more number of students

hereby give an undertaking that the dissertation < TITLE>_been checked for its Similarity Index/Plagiarism through Turnitin software tool; and that the document has been prepared by me and it is my original work and free of any plagiarism. It was found that:

1.	The Similarity Index (SI) was: *(Note: SI range: 0 to 10%; if SI is >10%, then authors cannot communicate ms; **attachment of SI report is mandatory**)*	6%
2.	The ethical clearance for research work conducted obtained from: *(Note: Name the consent obtaining body; if 'not appliable' then write so)*	NA
3.	The source of funding for research was: *(Note: Name the funding agency; or write 'self' if no funding source is involved)*	Self
4.	Conflict of interest: *(Note: Tick √ whichever is applicable)*	No
5.	The material (adopted text, tables, figures, graphs, etc.) as has been obtained from other sources, has been duly acknowledged in the manuscript: *(Note: Tick √ whichever is applicable)*	Yes

In case if any of the above-furnished information is found false at any point in time, then the University authorities can take action as deemed fit against all of us.

Full Name &
Signature of the student

Name &
Signature of SIU Guide/Mentor

Date:

Endorsement by
Academic Integrity Committee (AIC)

Place: Pune

Note: It is mandatory that the Similarity Index report of plagiarism (only first page) should be appended to the UG/PG dissertation